Московский государственный университет имени М. В. Ломоносова

На правах рукописи
УДК 519.24

Шаповалов Роман Викторович

Методы структурного обучения в задачах совместной разметки

Специальность 01.01.09 —
«Дискретная математика и математическая кибернетика»

Диссертация на соискание учёной степени
кандидата физико-математических наук

Научный руководитель:
к. ф.-м. н.
Ветров Дмитрий Петрович

Москва – 2014

Содержание

Введение

Задачей машинного обучения с учителем является восстановление функциональной зависимости между случайными величинами X и Y по обучающей выборке $\{(\mathbf{x}^j, y^j)\}_{j=1}^J$. В классической постановке задачи Y является скалярной случайной величиной, а пары $(\mathbf{x}^j, y^j)$ получаются *независимой* выборкой из генеральной совокупности. Это позволяет прогнозировать значение y лишь по соответствующему значению $\mathbf{x}$. Однако во многих практических задачах это предположение о независимости не выполняется. Тогда моделирование зависимости между переменными Y^j позволяет повысить качество предсказания. Для этого необходимо принимать решение о значениях y^j совместно. Приведём несколько примеров таких задач из разных областей.

Компьютерное зрение. Одной из центральных задач компьютерного зрения является семантическая сегментация — одновременное распознавание категорий объектов сцены и их сегментация [1, §14.4.3]. В семантической сегментации изображений каждому пикселю изображения назначается одна из семантических категорий [2–4]. В семантической сегментации облаков точек, полученных лазерным сканированием или сшиванием карт глубины, каждой точке поверхности ставится в соответствие метка категории [5, 6]. При этом категории представляют собой сущности реального мира, такие как ‘земля’, ‘небо’, ‘велосипед’, ‘стол’, ‘книга’, и т.д. Соседние пиксели или точки могут быть предварительно сгруппированы в *суперпиксели*. Получение качественной семантической сегментации — значительный шаг к решению задачи понимания сцены. В данной работе эксперименты проводятся в основном с семантической сегментацией.

Родственной является задача оценки геометрии сцены по одному изображению [6, 7]. Предполагается, что оно представляет собой фотографию городской сцены, где могут присутствовать земля и небо, а между ними находятся в основном вертикальные поверхности, такие как стены домов. Каждому пикселю изображения необходимо сопоставить метку одной из категорий ‘земля’, ‘небо’, ‘вертикаль’. Подобная информация позволяет делать выводы о трёхмерной геометрии сцены и помогает решать более высокоуровневые задачи, такие как распознавание пешеходов или построение трёхмерной модели сцены.

Другая задача — определение диспаритетов пикселей через поиск соответствий в стереопаре — паре изображений, снятых с соседних ракурсов [8]. При определённых условиях найденные диспаритеты можно использовать для однозначного определения глубины точек сцены.

В задачах низкоуровневой **обработки изображений**, в частности, в обратных задачах восстановления изображений, также необходимо учитывать зависимость между исходными яркостями пикселей, для чего часто моделируют априорное распределение над изображениями. В задаче шумоподавления [9, 10] восстановленное значение цвета пикселя должно соответствовать цвету окружения. В задаче устранения размытости [11] также можно стремиться получить характерные именно для реальных фотографий локальные участки изображения.

Вычислительная лингвистика. В задаче определения частей речи необходимо учитывать семантический контекст, то есть предсказанные части речи для соседних слов [12,13]. Например, английское слово ‘run’ может быть глаголом, существительным, или прилагательным, а ‘to’ — частицей, предлогом или наречием: без контекста часть речи нельзя определить точно.

На стыке вычислительной лингвистики и компьютерного зрения находится задача распознавания символов (англ. *optical character recognition, OCR*) [14]. В случае, если качество сканированного текста невысокое, или при распознавании рукописного текста, использование контекста повышает надёжность распознавания. Точно так же учёт контекста необходим при распознавании речи [15].

Биоинформатика. При поиске генов, кодирующих данный белок, также необходимо учитывать контекст [16]. Участки экзонов и интронов в ДНК имеют некоторые инвариантные характеристики, которые невозможно моделировать на локальном уровне.

В задаче определения структуры белка́ требуется определить конформации боковых цепей. Конформация одной цепи состоит из 4 или менее переменных, описывающих пространственные углы, которые можно дискретизовать. Конформации соседних цепей зависят друг от друга из-за образования нековалентных связей между ядрами участвующих в них атомов. Поэтому конформации соседних цепей нужно определять совместно [17].

Приведённые выше задачи с математической точки зрения являются задачами *совместной разметки*. По *признаковому описанию* объекта $\mathbf{x} \in \mathcal{X}$ необходимо получить *разметку* — вектор $\mathbf{y} \in \mathcal{Y}$ из V меток $[y_v]_{v=1}^V$, элементы которого $y_v \in \mathcal{K} = \{1, \dots, K\}$. Например, в задаче семантической сегментации изображений описание объекта $\mathbf{x}$ может включать в себя признаки суперпикселей (такие как признаки цвета, текстуры, формы), а также признаки подмножеств суперпикселей, описывающие специфичные для всей группы взаимодействия (например, расстояние между парой суперпикселей). Ответом являются значения меток суперпикселей y_v, которым назначается один из элементов множества меток категорий $\mathcal{K}$. В более общем случае, когда $\mathbf{y}$ представляет собой произвольный комбинаторный объект, применение функции $f : \mathcal{X} \to \mathcal{Y}$ называется *структурным предсказанием* (англ. *structural prediction*), а задача восстановления такой функции по выборке $\{(\mathbf{x}^j, \mathbf{y}^j)\}_{j=1}^J$ — *структурным обучением* (англ. *structural learning*, или *structured-output learning*).

Для решения задачи структурного обучения можно искать функцию совместного распределения в параметрическом виде, максимизируя правдоподобие $\prod_{j=1}^J \mathsf{P}(\mathbf{x}^j, \mathbf{y}^j \mid \mathbf{w})$ по параметрам $\mathbf{w}$. Такой подход называется *порождающим* (англ. *generative*) [18, §1.5.4]. Его недостат-

ком является необходимость моделировать распределение на признаки объектов $\mathbf{x}$, которые могут быть непрерывными и многомерными. Зная совместное распределение, можно порождать новые пары $(\mathbf{x}, \mathbf{y})$, однако это само по себе не требуется для структурного предсказания. Поэтому на практике чаще используется *разделяющий* (англ. *discriminative*) подход, в рамках которого максимизируется условное правдоподобие $\prod_{j=1}^{J} \mathsf{P}(\mathbf{y}^j \mid \mathbf{x}^j, \mathbf{w})$. Получив оценку максимального правдоподобия на параметры $\mathbf{w}_{\mathrm{ML}}$, структурное предсказание можно выполнять, получая моду апостериорного распределения на $\mathbf{y}$ для нового объекта $\mathbf{x}$:

$$f(\mathbf{x}) \equiv \operatorname*{argmax}_{\bar{\mathbf{y}} \in \mathcal{Y}} \mathsf{P}(\bar{\mathbf{y}} \mid \mathbf{x}, \mathbf{w}_{\mathrm{ML}}). \tag{1}$$

Если восстановление распределения необходимо только для поиска моды, можно ещё более упростить модель. Значение функции распределения для неправильной метки $\bar{\mathbf{y}}$ может быть любым, лишь бы оно было достаточно малым. Значит, для каждого объекта $\mathbf{x}$ можно стремиться максимизировать отступ между значением плотности для верной метки $\mathbf{y}$ и второй после неё:

$$\log \mathsf{P}(\mathbf{y} \mid \mathbf{x}, \mathbf{w}) - \max_{\bar{\mathbf{y}} \neq \mathbf{y}} \log \mathsf{P}(\bar{\mathbf{y}} \mid \mathbf{x}, \mathbf{w}) \to \max_{\mathbf{w}}. \tag{2}$$

На практике такая точечная оценка для вероятности «негативных» примеров может упростить процесс обучения. Кроме этого, важным преимуществом такого подхода является возможность учитывать функции потерь, специфичные для задачи. В то время как максимизация правдоподобия считает все неправильные метки одинаково плохими, небольшие отклонения часто допустимы на практике. Например, в задаче семантической сегментации неправильная разметка небольшого числа пикселей является нежелательной, но не критичной. Поэтому предлагается делать допустимый отступ зависимым от отклонения разметки. Если пользователь задаёт функцию отклонения $\Delta(\bar{\mathbf{y}}, \mathbf{y})$, то целевая функция для объекта обучающей выборки $(\mathbf{x}, \mathbf{y})$ выглядит следующим образом:

$$\log \mathsf{P}(\mathbf{y} \mid \mathbf{x}, \mathbf{w}) - \max_{\bar{\mathbf{y}} \neq \mathbf{y}} \{\log \mathsf{P}(\bar{\mathbf{y}} \mid \mathbf{x}, \mathbf{w}) + \Delta(\bar{\mathbf{y}}, \mathbf{y})\} \to \max_{\mathbf{w}}. \tag{3}$$

Такой метод называется *структурным обучением на основе максимизации отступа* (англ. *max-margin learning*) [19], или *структурным методом опорных векторов* (англ. *structural support vector machine, SSVM*) [20]. Классическим способом задания функции Δ для задач разметки является расстояние Хэмминга: $\Delta(\bar{\mathbf{y}}, \mathbf{y}) = \sum_{v=1}^{V} [\![\bar{y}_v \neq y_v]\!]$, однако в последнее время появилось много исследований по заданию нетрадиционных функций потерь [21, 22]. Более формальное обоснование метода, а также целевая функция для выборки, состоящей из более чем одного объекта, приведены в разделе 1.3.2.

Следует заметить, что хотя задачи разметки и представляют собой широкий класс задач структурного предсказания, они их не исчерпывают. Например, задача синтаксического разбора предложений [23] имеет на выходе дерево разбора, которое естественным образом не описывается дискретным вектором. В данной работе мы концентрируемся именно на задачах разметки, потому что предсказание может моделироваться с помощью хорошо разработанно-

го математического аппарата *ненаправленных графических моделей*. Более подробный обзор графических моделей и структурного обучения приведён в главе 1.

Недостатком описанного выше подхода является большая вычислительная сложность как на этапе обучения, так и на этапе предсказания. Были предприняты попытки создать каскадную систему структурного предсказания, в которой точностью можно жертвовать ради скорости предсказания [24]. Альтернативный подход заключается в использовании последовательной классификации. Алгоритм «автоконтекст» [25] применяет простые классификаторы, чтобы оценивать метки на основе меток других переменных. На практике он имеет небольшую временну́ю сложность и позволяет учитывать контекст, однако в отличие от предыдущих методов он не обоснован теоретически (то есть, алгоритм обучения нельзя представить в виде минимизации некоторой целевой функции). Росс и др. [26] интерпретировали последовательную классификацию как обобщение алгоритма передачи сообщений в фактор-графе, однако это не добавило теоретических гарантий. Тем не менее, сравнительно небольшая вычислительная сложность обучения и предсказания, а также высокая гибкость модели, позволяют рассматривать алгоритмы на основе последовательной классификации как один из мощных подходов к задаче совместной разметки. Более подробный обзор связанных методов дан в разделе 4.4.

Целью данной работы является сокращение усилий по аннотации обучающей выборки, повышение точности и скорости работы методов структурного обучения для решения задач разметки. Описываемые методы могут быть применены к любым задачам структурной разметки. В иллюстративных целях и при проведении экспериментов в данной работе в основном исследуется применимость методов к задачам семантической сегментации.

Для достижения поставленной цели были решены следующие задачи:

1. Исследована формулировка задачи структурного обучения, при которой часть обучающей выборки размечена не полностью, а известны лишь некоторые статистики разметки (*слабая аннотация*), такие как множество присутствующих меток. Предложена общая схема построения функций потерь структурного SVM для объектов, полная разметка которых недоступна, описаны несколько специальных функций потерь для конкретных видов слабой аннотации, а также методика их комбинирования в рамках одной оптимизационной задачи. Для этих специализаций предложены алгоритмы оптимизации, необходимые для структурного обучения. Экспериментально показано, что использование слабой аннотации позволяет повысить точность распознавания в задачах семантической сегментации изображений и определения тэгов (ключевых слов из фиксированного списка) текстовых документов [27].

2. Исследованы модификации структурного метода опорных векторов, позволяющие обучать более гибкую графическую модель. В частности, использован аппарат *неассоциативных марковских сетей*, который ранее редко использовался из-за трудностей при оптимизации функционала. Принципиальная возможность их применимости была показана с помощью эвристического способа обучения потенциалов [28, 29], затем для

обучения был применён структурный метод опорных векторов [30]. В этой работе также исследована применимость ядерного перехода в структурном SVM и применение аналога гауссова ядра, а также предложена модификация функции потерь, позволяющая обучаться на данных с выраженным дисбалансом категорий. Результаты экспериментов на задаче семантической сегментации трёхмерных облаков точек, полученных лазерным сканированием, показывают, что эти модификации позволяют настраивать более точную модель.

3. Исследованы методы последовательной классификации для задач разметки. Предложен метод обучения последовательной классификации, позволяющий учитывать априорные знания о структуре пространственных зависимостей между метками в задаче семантической сегментации. Экспериментальная проверка показала, что этот приём позволяет учитывать пространственный контекст, что ведёт к повышению качества сегментации трёхмерных облаков точек, полученных сшиванием карт глубины [31].

Актуальность и новизна. Подходы на основе структурного метода опорных векторов и последовательной классификации активно используются научным сообществом для решения задач совместной разметки (см. например [5, 32–35]). При этом используемые модели являются довольно грубыми. С увеличением размера обучающей выборки растёт актуальность использования более гибких моделей, многие из которых были до сих пор не исследованы. В данной диссертации исследуются модификации существующих моделей, в частности, обобщается формулировка структурного метода опорных векторов для учёта различных типов слабоаннотированных и полностью размеченных объектов обучающей выборки, обобщается традиционно используемый аппарат ассоциативных марковских сетей, предлагаются новые эмпирические функции потерь и гауссова ядровая функция для структурного метода опорных векторов. Предлагается новый аппарат д-факторов для учёта контекстуальных зависимостей в моделях последовательной классификации. Экспериментальная валидация показывает, что рассмотренные модификации позволяют достичь цели диссертационной работы — повышения точности и скорости работы соответствующих методов, а также снижения требований к обучающей выборке.

Апробация результатов. Основные результаты работы докладывались и обсуждались на конференции по фотограмметрическому компьютерному зрению и анализу изображений «PCV 2010» (г. Париж, Франция), на конференции по трёхмерному моделированию и обработке, визуализации и передаче трёхмерных изображений «IEEE 3DIMPVT 2011» (г. Ханчжоу, Китай), на конференциях по интеллектуализации обработки информации «ИОИ 2012» (г. Будва, Черногория) и математическим методам распознавания образов «ММРО 2013» (г. Казань), на конференции по компьютерному зрению и распознаванию образов «IEEE CVPR 2013» (г. Портлэнд, Орегон, США). Основные результаты по теме диссертации изложены в 7 научных публикациях.

Основные положения, выносимые на защиту:

- методы, обобщающие структурный SVM для обучения нелинейной неассоциативной марковской сети по слабоаннотированным данным, а также метод, позволяющий учитывать дальнодействующие пространственные зависимости при последовательной классификации;
- методика назначения функций потерь структурного SVM, учитывающих особенности обучающей выборки;
- экспериментальная апробация предложенных методов, сравнение точности и скорости работы с существующими методами.

Объём и структура работы. Диссертация состоит из введения, четырёх глав и заключения. Полный объём диссертации составляет 119 страниц с 19 рисунками, 8 таблицами и 5 листингами алгоритмов. Список литературы содержит 92 наименования. В следующей главе изложены основные факты теории ненаправленных графических моделей и структурного обучения. В последующих главах изложен новый материал: в главе 2 описана методика обучения структурного классификатора по выборке с различными типами аннотации; в главе 3 описан метод решения задач разметки на основе неассоциативных марковских сетей, обучаемых нелинейным структурным SVM с функцией потерь, учитывающей дисбаланс категорий; в главе 4 описана адаптация последовательной классификации для учёта пространственного контекста в задаче семантической сегментации.

Нотация. Переменные обозначаются буквами латинского или греческого алфавитов. Скалярные переменные набраны курсивом, векторы — прямым полужирным начертанием, множества — каллиграфическим. Элементы вектора обозначаются с помощью нижнего или верхнего индекса или круглых скобок, например, i-й элемент вектора $\mathbf{a}$ может быть обозначен a_i, a^i или $a(i)$. Для преобразования логических выражений используются скобки Айверсона:

$$[\![b]\!] = \begin{cases} 1, & \text{если } b \text{ верно,} \\ 0, & \text{иначе.} \end{cases} \tag{4}$$

Знак «$\propto$» означает равенство с точностью до постоянного мультипликативного коэффициента (пропорциональность). Основные используемые в тексте диссертации обозначения собраны в таблице 1.

Благодарности. Автор выражает благодарность своему научному руководителю Дмитрию Петровичу Ветрову, коллегам по факультету ВМК МГУ, в частности, Ольге Бариновой, Александру Велижеву, Антону Конушину, Антону Осокину, а также Пушмиту Коли.

Таблица 1: Символы, используемые в тексте диссертации

символ	значение
α_t^n	Коэффициент, соответствующий вкладу типа факторов t на итерации n
$\alpha_{\bar{\mathbf{Y}}}$	Целевая переменная в двойственной формулировке SSVM
β	Параметр, контролирующий вклад за неплотность рамок
γ	Ширина гауссова ядра
γ_n	Размер шага (суб)градиентного метода на n-й итерации
$\Delta(\bar{\mathbf{y}}, \mathbf{y})$	Функция потерь для разметки $\bar{\mathbf{y}}$ относительно корректной разметки $\mathbf{y}$
$\mathrm{K}(\bar{\mathbf{y}}; \mathbf{z})$	Функция потерь для разметки $\bar{\mathbf{y}}$ относительно корректной аннотации $\mathbf{z}$
$\boldsymbol{\lambda}$	Множители Лагранжа в формулировке двойственного разложения
$\boldsymbol{\mu}_{f\to v}$	Сообщение из фактора в вершину
$\boldsymbol{\mu}_{v\to f}$	Сообщение из вершины в фактор
$\nu_p^{\bar{z}}$	Штраф за пустоту строки p рамки $\bar{z}$
σ_k	Оценка числа пикселей категории k в функции потерь для плотных рамок
τ_k	Оценка числа пикселей категории k в функции потерь для зёрен
$\boldsymbol{\tau}$	Релаксация переопределённого представления конфигурации $\mathbf{y}$
$\boldsymbol{\Upsilon}$	Переопределённое представление конфигурации $\mathbf{y}$
$\Phi_f(\mathbf{y}_{\mathcal{C}_f})$	Фактор распределения Гиббса над элементами $\mathcal{C}_f$
$\phi_f(\mathbf{y}_{\mathcal{C}_f})$	Потенциал клики $\mathcal{C}_f$ в марковской сети
$\boldsymbol{\psi}^t(\mathbf{y}_{\mathcal{C}_f}; \mathbf{x}_f)$	Вектор обобщённых признаков фактора типа t над $\mathcal{C}_f$
$\boldsymbol{\psi}(\mathbf{y}; \mathbf{x})$	Суммарный вектор обобщённых признаков объекта $\mathbf{x}$
$\omega_q^{\bar{z}}$	Штраф за пустоту столбца q рамки $\bar{z}$
$\mathbf{b}_v$	Вектор убеждений о значении y_v
C	Гиперпараметр SSVM, контролирующий силу регуляризации
$\mathcal{C}_f$	Подмножество индексов вершин марковской сети (суперпикселей)
c_v	Площадь v-го суперпикселя
d_f	Приёмник д-фактора f
$E(\mathbf{y})$	Энергия марковской сети на конфигурации $\mathbf{y}$
$E^i(\boldsymbol{\mu})$	Энергия в i-й подзадаче при двойственном разложении
$\mathcal{E}$	Множество индексов рёбер парно-сепарабельной марковской сети
F	Количество факторов в распределении Гиббса
$\mathcal{F}$	Набор частей множества д-факторов обучающей выборки $\mathbf{f}$
$\mathbf{f}$	Часть множества д-факторов обучающей выборки
$\mathbf{g}_n$	Функция-предиктор на n-й итерации последовательной классификации
$\mathbf{g}(\mathbf{w})$	Градиент целевой функции SSVM по её параметрам
$H(\bar{\mathbf{Y}}, \tilde{\mathbf{Y}})$	Скалярное произведение обобщённых признаков
I	Количество слабоаннотированных объектов
J	Число объектов в обучающей выборке
K	Количество меток категорий — компонентов разметки, $\lvert\mathcal{K}\rvert$
$\mathcal{K}$	Множество индексов меток категорий — компонентов разметки

Таблица 1: Символы, используемые в тексте диссертации

символ	значение
$\mathcal{K}_{\mathrm{b}}$, $\mathcal{K}_{\mathrm{p}}$, $\mathcal{K}_{\mathrm{a}}$	Разбиение множества категорий в определении рамочной функции потерь
$\dot{k}$	Метка категории при аннотации зёрнами
$\mathcal{L}(\bar{\mathbf{y}}, \mathbf{y})$	Штраф за неправильную разметку
$L(\mathbf{z})$	Подмножество разметок $\mathbf{y}$, совместных со слабой аннотацией $\mathbf{z}$
$L(\boldsymbol{\lambda}, \{\bar{\boldsymbol{\mu}}^i\})$	Функция Лагранжа в формулировке двойственного разложения
N	Число итераций в машинах вывода и градиентных методах
$\mathbf{p} = (p, q)$	Координаты пикселя изображения
$\mathbf{p}_t$	Трёхмерные координаты t-й точки облака
$\dot{\mathbf{p}}$	Координаты зерна при аннотации зёрнами
Q	Ядровая функция в нелинейном SSVM
r_k	Штраф за неправильную классификацию суперпикселя категории k
$\mathcal{S}_f$	Передатчик д-фактора f
s_k	Оценка числа пикселей категории k в слабой функции потерь
$\mathcal{T}$	Множество типов факторов
T	Число точек в облаке
$t(f)$	Тип фактора f
$\mathcal{V}$	Множество индексов вершин марковской сети
V	Число вершин марковской сети (суперпикселей), $\lvert\mathcal{V}\rvert$
v, u	Индексы вершин марковской сети
$v(\mathbf{p})$	Функция, возвращающая номер суперпикселя, включающего $\mathbf{p}$
$\mathbf{w}$	Вектор параметров модели (весов)
$\mathcal{X}$	Множество возможных признаковых описаний
$\mathbf{x}^j$	Признаковое описание j-го объекта выборки
$\mathbf{X}$	Конкатенация признаков всех объектов выборки
$\mathbf{x}_v^{\mathrm{v}}$	Признаковое описание v-й вершины объекта $\mathbf{x}$
$\mathbf{x}_{vu}^{\mathrm{e}}$	Признаковое описание ребра (v, u) объекта $\mathbf{x}$
$\mathcal{Y}$	Множество возможных целевых переменных (разметок)
$\mathbf{y}^j$	Значение целевой переменной (разметка) j-го объекта выборки
$\mathbf{Y}$	Конкатенация разметок всех объектов выборки
y_v	Значение v-го компонента разметки (метка v-го суперпикселя)
Z	Нормировочная константа в распределении Гиббса
$\mathbf{z}$	Слабая аннотация объекта
$\mathbf{z}^i$	Слабая аннотация i-го объекта выборки
$\bar{z}$	Элемент рамочной аннотации изображения $\mathbf{z}^{\mathrm{bb}}$
$\dot{z}$	Элемент зерновой аннотации изображения $\mathbf{z}^{\mathrm{os}}$

Глава 1

Ненаправленные графические модели и структурное обучение

В этой главе приведены теоретические основы выполненной работы. В рассматриваемой задаче разметку удобно моделировать с помощью вероятностного распределения над возможными конфигурациями. Тогда процесс предсказания сводится к выводу конфигурации, на которой это распределение достигает максимума, либо к выводу маргинальных распределений на отдельные переменные. В реальных задачах приходится рассматривать факторизации совместного распределения, делая предположения о независимостях между случайными величинами. С этими факторизациями удобно работать с помощью механизма *графических вероятностных моделей*. Мы рассмотрим только ненаправленные графические модели (марковские сети) как наиболее полезные для решения задач разметки.

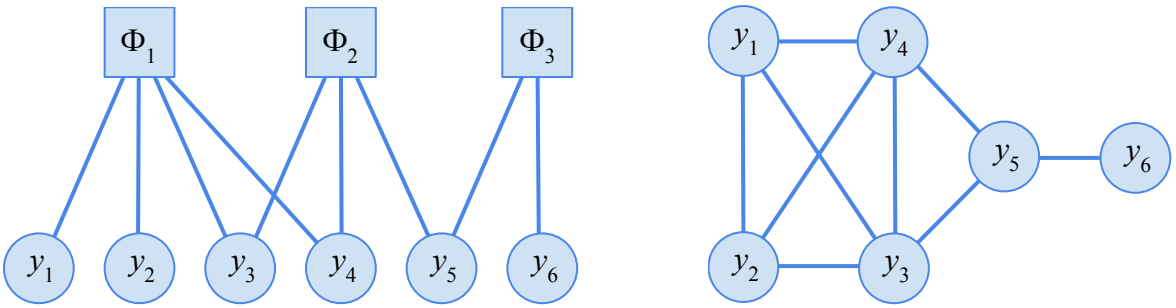
Для восстановления факторизованной плотности распределения по обучающей выборке её часто представляют в параметрическом виде. Далее эти параметры находятся как оптимальные значения некоторой целевой функции. Мы рассмотрим различные варианты целевых функций и методов оптимизации.

1.1 Марковские сети и связанные задачи

Пусть некоторый объект задан своим признаковым описанием $\mathbf{x} \in \mathcal{X}$, а также заданы некоторые параметры $\mathbf{w}$. Тогда можно определить апостериорное распределение над разметками $\mathbf{y} \in \mathcal{Y}$: $\mathsf{P}(\mathbf{y} \mid \mathbf{x}, \mathbf{w})$. В этом разделе мы предполагаем параметры $\mathbf{w}$ уже известными, а описание объекта $\mathbf{x}$ — фиксированным. Тогда можно не учитывать обуславливание распределения, и для краткости писать $\mathsf{P}(\mathbf{y})$, подразумевая апостериорное распределение. Мы предполагаем, что разметка $\mathbf{y}$ — вектор из V дискретных компонент, т.е. $\mathcal{Y} = \{1, \ldots, K\}^V$.

Определение 1.1. Пусть $\mathcal{C}_f \subset \{1, \ldots, V\}$ для $f \in \{1, \ldots, F\}$. *Распределение Гиббса* над вектором случайных переменных $\mathbf{y}$, параметризованным факторами $\{\Phi_1(\mathbf{y}_{\mathcal{C}_1}), \ldots, \Phi_F(\mathbf{y}_{\mathcal{C}_F})\}$ задаётся следующим образом:

$$\mathsf{P}(\mathbf{y}) = \frac{1}{Z} \prod_{f=1}^{F} \Phi_f(\mathbf{y}_{\mathcal{C}_f}), \tag{1.1}$$



(a) Фактор-граф распределения

(b) Марковская сеть

Рисунок 1.1: Различные графические представления распределения $\mathsf{P}(y_1, y_2, y_3, y_4, y_5, y_6) \propto \Phi_1(y_1, y_2, y_3, y_4)\Phi_2(y_3, y_4, y_5)\Phi_3(y_5, y_6)$: (a) фактор-граф, на котором круги соответствуют переменным, а квадраты — факторам; (b) марковская сеть, соответствующая распределению.

где $\mathbf{y}_{\mathcal{C}_f}$ — вектор из элементов $\mathbf{y}$ с индексами $\mathcal{C}_f$, Z — нормировочная константа:

$$Z = \sum_{\bar{\mathbf{y}} \in \mathcal{Y}} \prod_{f=1}^{F} \Phi_f(\bar{\mathbf{y}}_{\mathcal{C}_f}), \tag{1.2}$$

а *фактор* Φ_f — произвольная неотрицательная функция $|\mathcal{C}_f|$ переменных; величина $|\mathcal{C}_f|$ называется *порядком* фактора Φ_f.

Благодаря нормировочной константе Z выполняется свойство $\sum_{\bar{\mathbf{y}}} \mathsf{P}(\bar{\mathbf{y}}) = 1$.

Определение 1.2. *Фактор-графом*, соответствующим данному распределению, называется двудольный граф, у которого вершины одной доли соответствуют переменным-компонентам $\mathbf{y}$, а другой — факторам; вершины-факторы соединены с теми и только с теми вершинами-переменными, которые входят в фактор. Пример фактор-графа показан на рис. 1.1a.

Определение 1.3. *Марковской сетью* (англ. *Markov network, или Markov random field, MRF*), соответствующей строго положительному распределению Гиббса ($\forall \mathbf{y} : \mathsf{P}(\mathbf{y}) > 0$), называется граф, вершины которого соответствуют компонентам $\mathbf{y}$, и на каждом из множеств вершин $\mathcal{C}_f$ образован полный подграф. В таком случае говорят, что распределение Гиббса факторизуется на данную марковскую сеть. Пример марковской сети показан на рис. 1.1b.

Замечание. В литературе марковская сеть обычно определяется через предположения об условной независимости входящих в неё случайных величин, зависящие от структуры графа [36, раздел 19.2.1], а определение 1.3 выводится как их следствие (теорема Хаммерсли–Клиффорда [36, теорема 19.3.1]). Поскольку в данном обзоре мы не касаемся вероятностного моделирования, будем считать это определение марковской сети основным.

В литературе также используется *энергетическая нотация*. Можно записать эквивалентное определение:

$$\mathsf{P}(\mathbf{y}) = \frac{1}{Z} \exp(-E(\mathbf{y})), \tag{1.3}$$

где

$$E(\mathbf{y}) = \sum_{f=1}^{F} \phi_f(\mathbf{y}_{\mathcal{C}_f}), \quad \phi_i(\mathbf{y}_{\mathcal{C}_f}) = -\log \Phi_f(\mathbf{y}_{\mathcal{C}_f}). \tag{1.4}$$

Функция $E(\mathbf{y})$ называется *энергией*, а функции $\phi_f(\mathbf{y}_{\mathcal{C}_f})$ — *потенциалами* марковской сети.

Распределение Гиббса — дискретное, определено на конечном домене, но его табличная запись содержала бы K^V значений, поэтому с ним обычно работают в неявном виде. Двумя важными задачами являются вывод моды распределения и вывод маргинальных распределений.

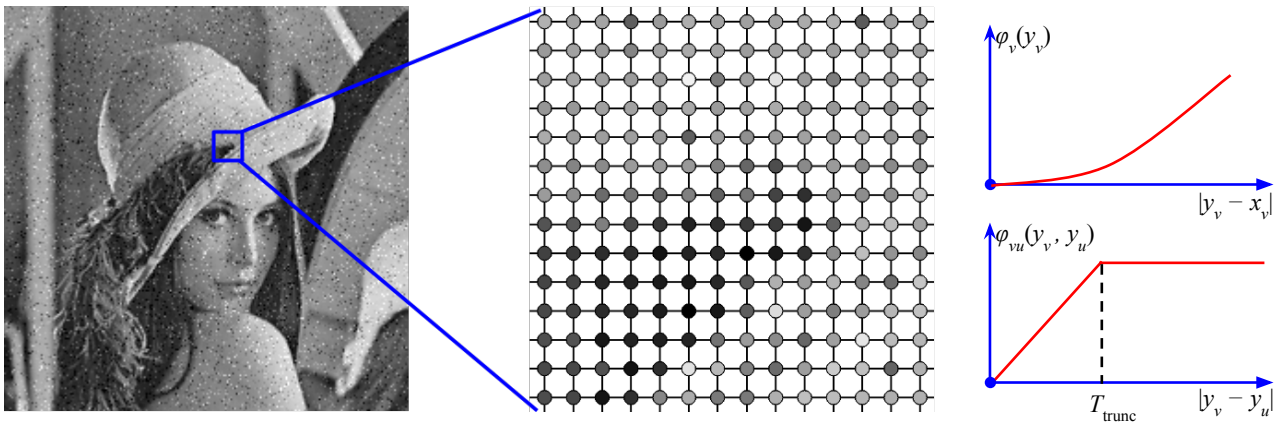
Определение 1.4. *Модой* распределения $\mathsf{P}(\mathbf{y})$, или MAP-оценкой (англ. *maximum a posteriori*), называется его самый вероятный элемент: $\mathbf{y}_{\text{MAP}} = \operatorname{argmax}_{\mathbf{y}} \mathsf{P}(\mathbf{y})$. Поскольку максимизация не зависит от нормировочной константы Z, MAP-оценка также является *минимумом энергии* марковской сети: $\mathbf{y}_{\text{MAP}} = \operatorname{argmin}_{\mathbf{y}} E(\mathbf{y})$.

В задачах разметки часто берут MAP-оценку в качестве ответа. Например, в задаче семантической сегментации каждому суперпикселю соответствует элемент вектора $\mathbf{y}$; оптимальная конфигурация находится минимизацией энергии марковской сети специального вида. В практических задачах множества $\mathcal{C}_f$ переменных, входящих в различные факторы, пересекаются, поэтому приходится осуществлять глобальную оптимизацию.

Байесовская теория принятия решений позволяет учитывать функцию потерь, задаваемую из экспертных соображений [36, §5.7]. Например, в задаче семантической сегментации предпочтительнее предсказать разметку, отличающуюся в одном пикселе, а не в половине изображения. Пусть $\bar{\mathbf{y}}$ — верная разметка, тогда необходимо определить функцию $\mathcal{L} : \mathcal{Y} \to \mathbb{R}$, определяющую штраф за несоответствие разметки $\mathbf{y}$ верной разметке. Тогда вектор $\mathbf{y}$ выводится как минимум математического ожидания функции потерь по апостериорному распределению: $\mathbf{y}_{\text{B}} = \operatorname{argmin}_{\mathbf{y}} \mathsf{E}_{\mathsf{P}(\bar{\mathbf{y}})} \mathcal{L}(\mathbf{y}; \bar{\mathbf{y}}) = \operatorname{argmin}_{\mathbf{y}} \sum_{\bar{\mathbf{y}} \in \mathcal{Y}} \mathcal{L}(\mathbf{y}; \bar{\mathbf{y}}) \mathsf{P}(\bar{\mathbf{y}})$. Заметим, что эта схема является обобщением MAP-оценивания: при использовании бинарной функции потерь $\mathcal{L}(\mathbf{y}; \bar{\mathbf{y}}) = [\![\mathbf{y} \neq \bar{\mathbf{y}}]\!]$ оптимальное Байесовское решение совпадает с MAP-оценкой. На практике использование нетривиальных функций потерь сопряжено с вычислительными трудностями, поэтому они используются редко, однако при настройке параметров использование некоторых функций потерь помогает улучшить обобщающую способность модели, при этом существует выпуклая верхняя оценка на соответствующую целевую функцию, см. раздел 1.3.2.

Также в некоторых задачах приходится оценивать *маргинальные распределения* на индивидуальные переменные $\mathsf{P}(y_v) \propto \sum_{\mathbf{y} \setminus y_v} \exp(-E(\mathbf{y}))$ или их группы $\mathsf{P}(\mathbf{y}_{\mathcal{C}}) \propto \sum_{\mathbf{y} \setminus \mathbf{y}_{\mathcal{C}}} \exp(-E(\mathbf{y}))$. Существуют алгоритмы, позволяющие найти приближённые значения маргиналов эффективнее явного суммирования. Помимо непосредственного интереса к распределению на переменные, ненормированные маргиналы могут быть использованы для эффективного вычисления математического ожидания признаков факторов, что требуется в некоторых методах обучения параметров (раздел 1.3.1).

Рассмотрим класс марковских сетей, наиболее часто используемый на практике.



(a) Структура марковской сети (b) Задание потенциалов

Рисунок 1.2: Пример использования 4-связной парно-сепарабельной марковской сети для подавления шума на изображении. (a) Зашумлённое изображение, в котором каждый пиксель соответствует вершине марковской сети, и структура сети для части изображения. Исходные интенсивности x_v служат для задания унарных потенциалов. (b) Пример задания унарных и парных потенциалов. Значение парного потенциала не зависит от исходных интенсивностей. Оно поощряет близкие значения интенсивности восстановленного изображения в соседних пикселях, при этом выше порога T_{trunc} значение потенциала не наращивается: штраф для возможных границ на изображении постоянен.

Определение 1.5. *Парно-сепарабельные* марковские сети — такие марковские сети, в которых используются только потенциалы порядка один и два. Рассмотрим граф $G = (\mathcal{V}, \mathcal{E})$, где вершины $\mathcal{V} = \{1, \ldots, V\}$ соответствуют переменным, а рёбра $\mathcal{E} \subset \mathcal{V}^2$ определяют факторы. Тогда энергия парно-сепарабельной марковской сети определяется как:

$$E(\mathbf{y}) = \sum_{v \in \mathcal{V}} \phi_v(y_v) + \sum_{(v,u) \in \mathcal{E}} \phi_{vu}(y_v, y_u). \tag{1.5}$$

В таком случае потенциалы ϕ_v называют *унарными*, а ϕ_{vu} — *парными*.

Заметим, что если в графе G нет изолированных вершин, то унарные потенциалы избыточны — модификацией парных потенциалов можно получить эквивалентную функцию энергии. Однако их часто моделируют отдельно, поскольку они имеют интерпретируемое значение, а также могут быть важнее парных, поэтому при настройке параметров их параметры регуляризуются слабее.

Рассмотрим пример. Парно-сепарабельная марковская сеть может использоваться для подавления некоторых видов шумов на изображении. Вершины $\mathcal{V}$ могут индексировать пиксели, а рёбра $\mathcal{E}$ — задавать 4-связную систему соседства над ними, переменные y_v кодируют восстановленные значения цвета соответствующих пикселей (рис. 1.2). Тогда унарные потенциалы задаются так, чтобы штрафовать отклонение от цвета пикселя зашумлённого изображения, а парные — чтобы штрафовать разность цветов соседних пикселей (используется априорное предположение, что границы областей постоянных цветов занимают малую часть площади

изображения). В этой задаче унарные потенциалы имеют естественный смысл, поэтому их удобно моделировать отдельно.

Большинство эффективных алгоритмов минимизации работают с парно-сепарабельными энергиями, однако в последнее время стали активно изучаться методы оптимизации вывода в марковских сетях с факторами высоких порядков, а также их приложения. Например, в задаче подавления шумов такие факторы могут поощрять участки восстановленного изображения, похожие на ранее встретившиеся в обучающей выборке [10]. В задаче семантической сегментации факторы высоких порядков, построенные над кластерами пикселей, позволяют повысить качество разметки [6, 37]. В данной работе алгоритмы минимизации энергии с потенциалами высоких порядков используются в алгоритме настройки параметров парно-сепарабельных марковских сетей по слабой аннотации, см. главу 2.

1.2 Алгоритмы вывода MAP-оценки

Задача вывода MAP-оценки (минимизации энергии) является одной из ключевых задач теории графических вероятностных моделей. В этом разделе мы проводим обзор основных групп методов, используемых в данной работе. Более подробный обзор можно найти в специализированных учебниках [36, 38]. Описание методов приводится для минимизации энергии парно-сепарабельной марковской сети вида (1.5), в конце раздела затрагивается вопрос минимизации энергии с потенциалами высоких порядков.

В общем случае задача минимизации энергии марковской сети является NP-трудной (к ней сводится задача 3-выполнимости [39]), поэтому большинство описываемых подходов дают приближённое решение. Нет общей теории, описывающей точность разных аппроксимаций для специальных задач, поэтому проводятся экспериментальные сравнения алгоритмов [40]. Ниже также описаны два частных случая, для которых существуют полиномиальные алгоритмы минимизации.

1.2.1 Как задача математического программирования

Задачу безусловной оптимизации энергии (1.5) можно переписать в эквивалентном виде как задачу целочисленного линейного программирования (ЦЛП):

Оптимизационная задача 1.1 (минимизация энергии как задача ЦЛП)**.**

$$\min_{\boldsymbol{\Upsilon}} \sum_{v\in\mathcal{V}}\sum_{k=1}^{K}\phi_v(k)\Upsilon_{v,k} + \sum_{(v,u)\in\mathcal{E}}\sum_{k=1}^{K}\sum_{l=1}^{K}\phi_{vu}(k,l)\Upsilon_{vu,kl}, \tag{1.6}$$

при условиях
$$\sum_{k=1}^{K}\Upsilon_{v,k} = 1, \quad \forall v\in\mathcal{V}, \tag{1.7}$$

$$\sum_{k=1}^{K}\Upsilon_{vu,kl} = \Upsilon_{u,l},\ \forall l, \quad \sum_{l=1}^{K}\Upsilon_{vu,kl} = \Upsilon_{v,k},\ \forall k, \quad \forall (v,u)\in\mathcal{E}, \tag{1.8}$$

$$\boldsymbol{\Upsilon}\in\{0,1\}^{K|\mathcal{V}|+K^2|\mathcal{E}|}. \tag{1.9}$$

Представление решения в виде вектора бинарных переменных $\boldsymbol{\Upsilon}$ называется переопределённым (англ. *overcomplete representation*). Ограничения (1.7) гарантируют, что ровно одна из бинарных переменных $\Upsilon_{v,k}$, соответствующих фиксированной исходной переменной y_v, равна 1. Ограничения (1.8) задают согласованность между бинарными переменными для унарных и парных потенциалов. Переход от решения задачи 1.1 к минимуму энергии (1.5) осуществляется следующим образом: $\Upsilon_{v,k} = 1 \iff y_v = k$.

В общем виде задача целочисленного линейного программирования является NP-трудной (задача выполнимости логических выражений является её частным случаем). Поэтому на практике она применима только для небольших задач; в остальных случаях можно ослабить ограничение целочисленности и решать задачу линейного программирования (LP-релаксацию исходной задачи):

Оптимизационная задача 1.2 (LP-релаксация минимизации энергии)**.**

$$\min_{\boldsymbol{\tau}\in\mathbb{R}^{K|\mathcal{V}|+K^2|\mathcal{E}|}} \sum_{v\in\mathcal{V}}\sum_{k=1}^{K}\phi_v(k)\tau_{v,k} + \sum_{(v,u)\in\mathcal{E}}\sum_{k=1}^{K}\sum_{l=1}^{K}\phi_{vu}(k,l)\tau_{vu,kl}, \tag{1.10}$$

при условиях
$$\sum_{k=1}^{K}\tau_{v,k} = 1, \quad \forall v\in\mathcal{V}, \tag{1.11}$$

$$\sum_{k=1}^{K}\tau_{vu,kl} = \tau_{u,l},\ \forall l, \quad \sum_{l=1}^{K}\tau_{vu,kl} = \tau_{v,k},\ \forall k, \quad \forall (v,u)\in\mathcal{E}, \tag{1.12}$$

$$\boldsymbol{\tau}\geq\mathbf{0}. \tag{1.13}$$

Из ограничений (1.11) и (1.13) следует, что каждая из компонент допустимого решения $\boldsymbol{\tau}$ лежит на отрезке $[0,1]$. В общем случае оптимальное решение может содержать дробные решения. Дробные компоненты обычно округляют, используя некоторый жадный алгоритм. Таким образом получается приближённое значение минимума энергии.

1.2.2 Передача сообщений

Один из наиболее общих способов вывода MAP-оценки и маргинальных распределений дают методы на основе передачи сообщений. Общей идеей этих методов является постро-

ение итеративного процесса, в рамках которого между переменными и факторами пересылаются *сообщения*, обновляющие *убеждения* (англ. *beliefs*) о маргинальных распределениях или элементах MAP-оценки для отдельных переменных на основе влияния их «соседей» по графической модели.

Мы определим формулы передачи сообщений для определения маргинальных распределений и затем покажем, как их можно модифицировать для нахождения MAP-оценки.

Определение 1.6. *Сообщением* $\mu_{f\to v}$ из фактора с индексом f в вершину с индексом v называется вектор из K элементов, которые вычисляются следующим образом:

$$\mu_{f\to v}(y) = \sum_{\mathbf{y}'_f : y'_v = y} \Phi_f(\mathbf{y}'_f) \prod_{v' \in \mathcal{C}_f \setminus \{v\}} \mu_{v'\to f}(y'_{v'}), \tag{1.14}$$

Здесь $\mu_{v'\to f}$ — сообщение из вершины с индексом v' в фактор с индексом f называется вектор из K элементов, которые в свою очередь вычисляются по предыдущим значениям сообщений из фактора в вершину:

$$\mu_{v\to f}(y) = \prod_{f' :\ v\in\mathcal{C}_{f'}, f'\neq f} \mu_{f'\to v}(y). \tag{1.15}$$

При фиксированных значениях сообщений, *убеждение* о маргинальном распределении в вершине с индексом v вычисляется так:

$$b_v(y) \propto \prod_{f' :\ v\in\mathcal{C}_{f'}} \mu_{f'\to v}(y), \quad \sum_{y=1}^{k} b_v(y) = 1. \tag{1.16}$$

Для того чтобы определить конкретный алгоритм, осталось определить инициализацию сообщений и порядок их пересчёта. В случае когда фактор-граф не содержит циклов, существует алгоритм *распространения убеждений* (англ. *belief propagation*), позволяющий получить точные оценки маргиналов. У фактор–графа выбирается корневая вершина, а сообщения из висячих вершин-переменных в соответствующие факторы инициализируются единичными векторами. Затем асинхронно пересчитываются все сообщения по направлению к корню. После этого пересчитываются сообщения из корня к листьям. Показано, что после этих двух проходов процесс пересчёта сообщений сходится, а полученные оценки убеждений (1.16) являются точными оценками маргинальных распределений [36, §20.2.1]. Если же операцию суммирования в (1.14) заменить на взятие максимума, то MAP-оценку можно будет получить конкатенацией аргмаксимумов полученных убеждений (1.16).

Аналогичный процесс можно применить к циклическому графу, однако в этом случае нельзя установить порядок пересчёта такой, чтобы алгоритм гарантировано сходился. Более того, даже если алгоритм сходится, найденные убеждения могут не соответствовать маргинальным распределениям. Однако на практике метод оказывается полезным, даже если приходится останавливать передачу сообщений по числу итераций. Для возникающих на практике моделей метод часто возвращает близкое к оптимальному решение, особенно если фактор-граф не содержит коротких циклов.

1.2.3 Двойственное разложение

Группа методов на основе двойственного разложения (двойственной декомпозиции) рассматривает двойственную оптимизационную задачу к задаче поиска минимума энергии марковской сети, которая является вогнутой, поэтому в ней возможно найти глобальный максимум, являющийся нижней оценкой на значение энергии в прямой задаче [41]. Поскольку рассматривается дискретная задача, в общем случае существует ненулевой зазор между минимумом прямой и максимумом двойственной задачи, однако метод поиска максимума в двойственной задаче позволяет выписать приближённое решение прямой задачи, а также оценить сверху субоптимальность любого решения (разницу между энергией в точке и минимальной энергией). Преимуществом данного метода является возможность использования потенциалов высоких порядков [42].

Рассмотрим переопределённое представление $\boldsymbol{\Upsilon}$ вектора переменных $\mathbf{y}$:

$$\Upsilon_{v,k} = \begin{cases} 1, & \text{если } y_v = k, \\ 0, & \text{иначе,} \end{cases} \qquad \forall v \in \mathcal{V},\ \forall k \in \mathcal{K}. \tag{1.17}$$

Отождествим значение энергии на переопределённой разметке с соответствующей энергией: $E(\boldsymbol{\Upsilon}) \equiv E(\mathbf{y})$. Предполагается, что энергия представима в следующем виде:

$$E(\boldsymbol{\Upsilon}) = \sum_i E^i(\boldsymbol{\Upsilon}), \tag{1.18}$$

где минимизация отдельных $E^i(\boldsymbol{\Upsilon})$ может быть выполнена с низкими вычислительными затратами (как правило, используются алгоритмы со сложностью не более линейной по числу вершин в марковской сети). Самым простым примером такого представления является декомпозиция на отдельные факторы: $E^i(\boldsymbol{\Upsilon}) \equiv \phi_i(\mathbf{y}_{\mathcal{C}_i}), \forall i \in F$, применимая, когда максимальный порядок фактора ограничен сверху некоторой константой; тогда сложность минимизации $E^i(\boldsymbol{\Upsilon})$ не зависит от общего числа переменных. С учётом (1.18) задача минимизации энергии (1.4) эквивалентна следующей задаче:

Оптимизационная задача 1.3.

$$\min_{\boldsymbol{\Upsilon}} \sum_i E^i(\boldsymbol{\Upsilon}), \tag{1.19}$$

$$\text{при условии} \quad \boldsymbol{\Upsilon} \in \mathrm{M}, \tag{1.20}$$

где $\mathrm{M} = \left\{ [\boldsymbol{\Upsilon}_v]_{v \in \mathcal{V}} \in \{0,1\}^{K \cdot V} \mid \sum_{k \in \mathcal{K}} \Upsilon_{v,k} = 1, \forall v \in \mathcal{V} \right\}$ — ограничение, накладываемое видом переопределённого представления.

Наша цель — построить релаксацию задачи 1.3, чтобы затем найти двойственную к ней. Для этого введём дополнительные переменные $\bar{\boldsymbol{\Upsilon}}^i$, соответствующие подзадачам E^i, и запишем эквивалентную задачу минимизации:

Оптимизационная задача 1.4 (минимизация разложенной энергии)**.**

$$\min_{\boldsymbol{\Upsilon},\{\bar{\boldsymbol{\Upsilon}}^i\}} \sum_i E^i(\boldsymbol{\Upsilon}^i), \tag{1.21}$$

$$\text{при условиях} \quad \boldsymbol{\Upsilon} \in \mathrm{M}, \quad \bar{\boldsymbol{\Upsilon}}^i \in \mathrm{M}, \quad \forall i, \tag{1.22}$$

$$\bar{\boldsymbol{\Upsilon}}^i = \boldsymbol{\Upsilon}, \quad \forall i. \tag{1.23}$$

Запишем частичную функцию Лагранжа по ограничениям (1.23) для этой задачи:

$$L(\boldsymbol{\lambda}, \boldsymbol{\Upsilon}, \{\bar{\boldsymbol{\Upsilon}}^i\}) = \sum_i E^i(\bar{\boldsymbol{\Upsilon}}^i) + \sum_i (\bar{\boldsymbol{\Upsilon}}^i - \boldsymbol{\Upsilon})^\intercal \boldsymbol{\lambda}^i = \sum_i \left(E^i(\bar{\boldsymbol{\Upsilon}}^i) + \bar{\boldsymbol{\Upsilon}}^{i\intercal} \boldsymbol{\lambda}^i \right) - \boldsymbol{\Upsilon}^\intercal \sum_i \boldsymbol{\lambda}^i. \tag{1.24}$$

При любом фиксированном значении $\boldsymbol{\lambda}$ минимизация $L(\boldsymbol{\lambda}, \boldsymbol{\Upsilon}, \{\bar{\boldsymbol{\Upsilon}}^i\})$ по $\boldsymbol{\Upsilon}, \{\bar{\boldsymbol{\Upsilon}}^i\}$ при условиях (1.22)–(1.23) эквивалентна задаче 1.4, а значит и исходной задаче минимизации энергии $E(\boldsymbol{\Upsilon})$. Если же опустить ограничения целостности (1.23), минимум функции Лагранжа при любом значении $\boldsymbol{\lambda}$ будет служить нижней оценкой на минимум исходной энергии:

$$\min_{\boldsymbol{\Upsilon} \in \mathrm{M}, \{\bar{\boldsymbol{\Upsilon}}^i \in \mathrm{M}\}} L(\boldsymbol{\lambda}, \boldsymbol{\Upsilon}, \{\bar{\boldsymbol{\Upsilon}}^i\}) \leq \min_{\boldsymbol{\Upsilon} \in \mathrm{M}} E(\boldsymbol{\Upsilon}), \quad \forall \boldsymbol{\lambda}. \tag{1.25}$$

Заметим, что при отсутствии ограничений целостности минимизация Лагранжиана по $\boldsymbol{\Upsilon}, \{\bar{\boldsymbol{\Upsilon}}^i\}$ может быть выполнена независимо по отдельным группам переменных, что может быть реализовано вычислительно эффективно по предположению, принятому при построении разложения (1.18).

Заменим всеобщность в (1.25) на максимум, получим эквивалентное условие:

$$\max_{\boldsymbol{\lambda}} \min_{\boldsymbol{\Upsilon} \in \mathrm{M}, \{\bar{\boldsymbol{\Upsilon}}^i \in \mathrm{M}\}} L(\boldsymbol{\lambda}, \boldsymbol{\Upsilon}, \{\bar{\boldsymbol{\Upsilon}}^i\}) \leq \min_{\boldsymbol{\Upsilon} \in \mathrm{M}} E(\boldsymbol{\Upsilon}). \tag{1.26}$$

Идея алгоритмов двойственного разложения заключается в максимизации этой нижней оценки на минимум энергии. Её можно осуществлять с помощью блочно-координатного подъёма [43,44] или субградиентного подъёма [41]. Получим выражение для компонент субградиента:

$$\nabla_{\boldsymbol{\lambda}^i} \left[\min_{\boldsymbol{\Upsilon} \in \mathrm{M}, \{\bar{\boldsymbol{\Upsilon}}^i \in \mathrm{M}\}} L(\boldsymbol{\lambda}, \boldsymbol{\Upsilon}, \{\bar{\boldsymbol{\Upsilon}}^i\}) \right] = \nabla_{\boldsymbol{\lambda}^i} \min_{\bar{\boldsymbol{\Upsilon}}^i \in \mathrm{M}} \left\{ E^i(\bar{\boldsymbol{\Upsilon}}^i) + \bar{\boldsymbol{\Upsilon}}^{i\intercal} \boldsymbol{\lambda}^i \right\} - \nabla_{\boldsymbol{\lambda}^i} \max_{\boldsymbol{\Upsilon} \in \mathrm{M}} \boldsymbol{\Upsilon}^\intercal \textstyle\sum_j \boldsymbol{\lambda}^j = \dot{\bar{\boldsymbol{\Upsilon}}}^{i\intercal} - \dot{\boldsymbol{\Upsilon}}^\intercal, \tag{1.27}$$

где

$$\dot{\bar{\boldsymbol{\Upsilon}}}^i = \operatorname*{argmin}_{\bar{\boldsymbol{\Upsilon}}^i \in \mathrm{M}} \left\{ E^i(\bar{\boldsymbol{\Upsilon}}^i) + \bar{\boldsymbol{\Upsilon}}^{i\intercal} \boldsymbol{\lambda}^i \right\}, \quad \dot{\boldsymbol{\Upsilon}} = \operatorname*{argmax}_{\boldsymbol{\Upsilon} \in \mathrm{M}} \boldsymbol{\Upsilon}^\intercal \textstyle\sum_j \boldsymbol{\lambda}^j, \qquad \forall i. \tag{1.28}$$

Таким образом, алгоритм субградиентного подъёма поочерёдно выполняет два шага:

1. Производится оптимизация в подзадачах (1.28). В первой группе подзадач при вычислении $\dot{\bar{\boldsymbol{\Upsilon}}}^i$ в задаче минимизации относительно энергии $E^i(\bar{\boldsymbol{\Upsilon}}^i)$ изменяются только унарные потенциалы; как правило, для оптимизации может использоваться тот же алгоритм,

что и для минимизации $E^i(\bar{\boldsymbol{\Upsilon}}^i)$ без дополнительных вычислительных затрат. Во второй группе при вычислении $\dot{\boldsymbol{\Upsilon}}$ максимизация может проводиться независимо по векторам, отвечающим разным переменным марковской сети v.

2. Вычисляется субградиент минимума Лагранжиана согласно (1.27), делается шаг по субградиенту: $\boldsymbol{\lambda}_{n+1} \leftarrow \boldsymbol{\lambda}_n + \gamma_n \nabla_{\boldsymbol{\lambda}} \left[\min_{\boldsymbol{\Upsilon}\in\mathrm{M},\{\bar{\boldsymbol{\Upsilon}}^i\in\mathrm{M}\}} L(\boldsymbol{\lambda}_n, \boldsymbol{\Upsilon}, \{\bar{\boldsymbol{\Upsilon}}^i\})\right]$, где $\{\gamma_n\}$ — убывающая последовательность длин шагов.

Существуют различные способы представить энергию в виде суммы (1.18) помимо декомпозиции на отдельные факторы. Например, граф парно-сепарабельной марковской сети можно разбить на пересекающиеся поддеревья — энергию в ациклической марковской сети можно эффективно минимизировать с помощью алгоритма передачи сообщений (раздел 1.2.2). Алгоритм передачи сообщений с перевзвешиванием по деревьям (англ. *tree-reweighted message passing, TRW*) [43] представляет собой блочно-координатную оптимизацию двойственного функционала при декомпозиции графа на поддеревья. За счёт сравнительно небольшого числа подзадач метод требует меньше вычислительных ресурсов, чем при разбиении на отдельные факторы. Другой способ декомпозиции получается при разложении энергии на сумму так называемых субмодулярных функций [45], точная минимизация которых возможна с применением алгоритмов разрезов на графах (см. раздел 1.2.4).

1.2.4 Разрезы на графах

Эта группа методов сводит минимизацию энергии к классической комбинаторно-оптимизационной задаче построения минимального разреза в ориентированном графе. Такие методы, как правило, являются наиболее вычислительно эффективными методами минимизации энергии, однако область их применения ограничена. Рассмотрим сначала минимизацию парно-сепарабельной энергии (1.5) с бинарными переменными ($K = 2$), для которой в некоторых случаях удаётся найти точный минимум [46].

Определение 1.7. *Обобщённым потенциалом Поттса* называется парный потенциал вида

$$\phi_{vu}(y_v, y_u) = \begin{cases} 0, & \text{если } y_v = y_u, \\ \delta_{vu}, & \text{иначе,} \end{cases} \tag{1.29}$$

где $\delta_{vu} \geq 0$. Такой вид потенциалов поощряет назначение смежным вершинам одной и той же метки.

Покажем, как свести задачу минимизации энергии с обобщёнными потенциалами Поттса к задаче нахождения минимального разреза в ориентированном графе [36, §22.6.3.1]. Так как прибавление константы к любому из потенциалов не влияет на точку минимума функции энергии, вычтем из каждого унарного потенциала $\phi_v(y_v)$ величину $\min_y \phi_v(y)$, таким образом, среди $\phi_v(1)$ и $\phi_v(2)$ один будет нулевым, другой — неотрицательным. Рассмотрим граф, содержащий $|\mathcal{V}| + 2$ вершины — по одной вершине на переменные $v \in \mathcal{V}$, а также две дополнительные вершины s и t, и $|\mathcal{V}| + 2|\mathcal{E}|$ дуг. Если $\phi_v(1) = 0$, в графе присутствует дуга $s \to v$

с пропускной способностью $\phi_v(2)$, иначе — дуга $v \to t$ с пропускной способностью $\phi_v(1)$. Также каждому ребру марковской сети $(v,u) \in \mathcal{E}$ соодветствуют две дуги: $u \to v$ и $v \to u$ с одинаковой пропускной способностью δ_{vu}. Легко показать, что любой конфигурации переменных марковской сети соответствует разрез на графе между вершинами s и t: вершины, попавшие в разрез со стороны s, получают метку 1, остальные — метку 2, причём величина разреза равняется энергии марковской сети. Таким образом, минимальный s–t-разрез соответствует разметке, минимизирующей энергию марковской сети.

Определим более широкий класс парно-сепарабельных бинарных энергий, минимизация которых сводится к разрезам на графах.

Определение 1.8. Вещественная функция двух бинарных аргументов $f : \{1,2\}^2 \to \mathbb{R}$ называется *субмодулярной*, если $f(1,1) + f(2,2) \leq f(1,2) + f(2,1)$. Парно-сепарабельная бинарная энергия называется субмодулярной, если все её парные потенциалы — субмодулярные функции.

Покажем, как свести задачу минимизации произвольной субмодулярной энергии $E(\mathbf{y})$ к задаче нахождения минимального разреза в ориентированном графе. Преобразуем парные потенциалы следующим образом:

$$\phi'_{vu}(1,1) = \phi'_{vu}(1,2) = \phi'_{vu}(2,2) = 0, \tag{1.30}$$

$$\phi'_{vu}(2,1) = \phi_{vu}(2,1) + \phi_{vu}(1,2) - \phi_{vu}(1,1) - \phi_{vu}(2,2), \quad \forall (v,u) \in \mathcal{E}. \tag{1.31}$$

Из субмодулярности потенциалов исходной энергии следует неотрицательность выражения (1.31). Далее преобразуем унарные потенциалы:

$$\phi'_v(2) = \phi_v(2) + \sum_{\substack{(\bar{v},\bar{u})\in\mathcal{E}:\\ v=\bar{v}}} \big(\phi_{\bar{v}\bar{u}}(2,1) - \phi_{\bar{v}\bar{u}}(1,1)\big) + \sum_{\substack{(\bar{u},\bar{v})\in\mathcal{E}:\\ v=\bar{v}}} \big(\phi_{\bar{u}\bar{v}}(2,2) - \phi_{\bar{u}\bar{v}}(2,1)\big), \tag{1.32}$$

$$\phi'_v(1) = \phi_v(1), \quad \forall v \in \mathcal{V}. \tag{1.33}$$

Процедура сведения к задаче построения минимального разреза аналогична предыдущему случаю. Необходимо обнулить одно из значений унарных потенциалов:

$$\phi'_v(y) \leftarrow \phi'_v(y) - \min_{\bar{y}\in\mathcal{K}} \phi'_v(\bar{y}), \quad \forall v \in \mathcal{V}, \;\; \forall y \in \{1,2\}. \tag{1.34}$$

Таким образом, унарный потенциал соответствует либо дуге из s, либо дуге в t. В отличие от предыдущего случая, ребру (v,u) марковской сети соответствует одна дуга в графе, имеющая пропускную способность $\phi'_{vu}(2,1)$. Минимальный разрез в полученном графе также соответствует разметке, минимизирующей энергию марковской сети [46].

Минимизация небинарных энергий

Группа методов приближённой минимизации небинарных энергий марковских сетей опирается на пошаговый вызов алгоритма минимизации бинарной энергии. На каждом шаге оп-

тимизация проводится только по подмножеству переменных марковской сети, причём каждая переменная может либо оставить предыдущее значение, либо изменить его на некоторое другое, фиксированное на данном шаге. На каждом шаге значение энергии на текущей разметке уменьшается, поэтому методы находят локальный минимум относительно соответствующего вида шага.

Одним из таких алгоритмов является α-*расширение* [47]. На каждом шаге выбирается одна из меток α (случайно или поочерёдно), и над тем же графом $G(\mathcal{V}, \mathcal{E})$ строится дополнительная марковская сеть с бинарной энергией, в которой метке 1 соответствует сохранение предыдущего значения, а метке 2 — изменение значения на α. Если текущая разметка равна $\mathbf{y}$, потенциалы назначаются следующим образом:

$$\phi'_v(1) = \phi_v(y_v), \tag{1.35}$$
$$\phi'_v(2) = \phi_v(\alpha), \tag{1.36}$$
$$\phi'_{vu}(1,1) = \phi_{vu}(y_v, y_u), \tag{1.37}$$
$$\phi'_{vu}(1,2) = \phi_{vu}(y_v, \alpha), \tag{1.38}$$
$$\phi'_{vu}(2,1) = \phi_{vu}(\alpha, y_u), \tag{1.39}$$
$$\phi'_{vu}(2,2) = \phi_{vu}(\alpha, \alpha), \quad \forall v \in \mathcal{V}, \ \ \forall (v,u) \in \mathcal{E}. \tag{1.40}$$

Чтобы энергию дополнительных марковских сетей можно было оптимизировать с помощью алгоритмов разрезов на графах, необходимо потребовать её субмодулярность. Это ведёт к следующему ограничению на парные потенциалы:

$$\phi_{vu}(\beta,\gamma) + \phi_{vu}(\alpha,\alpha) \leq \phi_{vu}(\beta,\alpha) + \phi_{vu}(\alpha,\gamma), \quad \forall \alpha,\beta,\gamma \in \mathcal{K}, \ \ \forall (v,u) \in \mathcal{E}. \tag{1.41}$$

Для выполнения этого условия достаточно, чтобы парные потенциалы удовлетворяли аксиомам метрики (а при $\phi_{vu}(\alpha,\alpha) = 0$, $\phi_{vu}(\alpha,\beta) \geq 0$ условие становится эквивалентным определению метрики).

Другой метод из этой группы — $\alpha\beta$-*замена* [47]. Он отличается тем, что на каждом шаге выбирается пара меток (α и β), и рассматриваются только те вершины, которые в текущей разметке $\mathbf{y}$ уже имеют метку α или β. Назначение переменной метки 1 в дополнительной задаче соответствует сохранению метки, а метки 2 — изменению на противоположную (α на β, и наоборот). Метод $\alpha\beta$-замены применим к более широкому классу энергий — не требуется выполнение парными потенциалами неравенства треугольника (1.41), однако в случае применимости обоих вариантов, он как правило находит худший локальный минимум, чем α-расширение.

Потенциалы высоких порядков

В прикладных задачах бывает полезно моделировать факторы высоких порядков в марковских сетях. MAP-оценка в них может быть найдена приближённо с помощью алгоритмов передачи сообщений на фактор-графе (раздел 1.2.2) или двойственного разложения (раз-

дел 1.2.3), однако в некоторых случаях возможно свести задачу к построению минимального разреза в графе с помощью введения дополнительных вершин, что является предпочтительным из-за более высокой эффективности таких методов. Ниже мы конструктивно охарактеризуем класс потенциалов высокого порядка, допускающих такое сведение, и приведём примеры функций, полезных на практике.

Определение 1.9. Пусть $\mathcal{C} \subset \mathcal{V}$ — подмножество индексов переменных марковской сети. Функция $\phi^q : \mathcal{K}^{|\mathcal{C}|} \to \mathbb{R}$ называется *линейной*, если она представима в виде

$$\phi^q(\mathbf{y}_\mathcal{C}) = \omega_0^q + \sum_{v \in \mathcal{C}} \sum_{k \in \mathcal{K}} \omega_{v,k}^q [\![y_v = k]\!] = \omega_0^q + \sum_{v \in \mathcal{C}} \omega_{v,y_v}^q, \tag{1.42}$$

где $\boldsymbol{\omega}^q \in \mathbb{R}^{K|\mathcal{C}|+1}$ — набор параметров функции [48].

Определение 1.10. Представлением потенциальной функции в виде *нижней огибающей множества линейных функций* называется следующая запись:

$$\phi(\mathbf{y}_\mathcal{C}) = \min_{q \in \mathcal{Q}} \phi^q(\mathbf{y}_\mathcal{C}), \tag{1.43}$$

где $\{\phi^q(\cdot)\}_{q \in \mathcal{Q}}$ — множество линейных функций.

Для минимизации энергии, содержащей подобный потенциал, необходимо ввести дополнительную переменную $\bar{v}$, принимающую значения из $\mathcal{Q}$. Тогда потенциал может быть эквивалентно переформулирован:

$$\phi(\mathbf{y}_\mathcal{C}) = \min_{q \in \mathcal{Q}} \left\{ \phi_{\bar{v}}(q) + \sum_{v \in \mathcal{C}} \phi_{\bar{v},v}(q, y_v) \right\}, \tag{1.44}$$

где $\phi_{\bar{v}}(q) = \omega_0^q$, а $\phi_{\bar{v},v}(q, y_v) = \omega_{v,y_v}^q$, $\forall q \in \mathcal{Q}$, $\forall y_v \in \mathcal{K}$. При подстановке этого выражения в задачу минимизации энергии, минимизацию по всем дополнительным переменным можно вынести, таким образом задача превращается в совместную минимизацию модифицированной энергии по целевым и дополнительным переменным. Если для парного потенциала выполняется условие (1.41), можно применить α-расширение для минимизации энергии с потенциалами высокого порядка.

Любая потенциальная функция может быть представлена как нижняя огибающая множества линейных функций, однако для этого в общем случае требуется $|\mathcal{Q}| = |\mathcal{K}|^{|\mathcal{C}|}$ функций, и столько же значений дополнительной переменной. Каждой допустимой конфигурации переменных ставится в соответствие функция, которая конечна только в данной конфигурации, а в остальных точках равна $+\infty$. Минимум по таким функциям будет достигаться на функции, соответствующей данной конфигурации, для неё можно задать произвольное конечное значение потенциала. Из-за громоздкости этого представления и сложности соответствующих алгоритмов минимизации на практике используют другие, разреженные представления.

Для бинарных задач (при $|\mathcal{K}| = 2$) возможно представить потенциальную функцию вида (1.43) в виде суммы $|\mathcal{Q}| - 1$ минимумов из двух линейных функций [49]. Поскольку каж-

дый из таких минимумов может быть учтён при минимизации энергии с помощью добавления одной бинарной переменной, а соответствующие ей парные потенциалы оказываются субмодулярными [49, утв. 3.5], модифицированная функция энергии может быть эффективно минимизирована с помощью алгоритма построения разреза в графе. Следствием этого является тот факт, что любая вогнутая функция от $\sum_{v\in\mathcal{C}}[\![y_v = k]\!]$ может быть использована для задания потенциала высокого порядка, и при этом будет возможно применение разрезов на графе.

Для небинарных задач важным частным случаем (1.43) является модель Поттса в классе $\mathcal{P}^n$ и её робастный вариант. Они используются в задаче сегментации изображений, чтобы получить сегментацию более подробную, чем сегментацию на уровне суперпикселей. Для этого марковская сеть строится над пикселями изображения (а не над суперпикселями), а для задания потенциалов высокого порядка используются перекрывающиеся пересегментации, каждому сегменту которых соответствует фактор высокого порядка, поощряющий назначение всем пикселям соответствующего сегмента одной и той же метки [50, 51].

Определение 1.11. *Потенциалом Поттса в классе* $\mathcal{P}^n$ называется потенциальная функция, представимая в виде

$$\phi(\mathbf{y}_\mathcal{C}) = \begin{cases} \delta^k, & \text{если } \exists k \in \mathcal{K} : \forall v \in \mathcal{C}, y_v = k, \\ \delta_\mathcal{C}, & \text{иначе,} \end{cases} \tag{1.45}$$

где δ^k — значение потенциала, когда все переменные имеют одно и то же значение k, $\delta_\mathcal{C}$ — штраф, если не все переменные имеют одинаковое значение, причём $\delta_\mathcal{C} \geq \delta^k,\ \forall k \in \mathcal{K}$. Заметим, что это определение является обобщением потенциала Поттса для парных потенциалов (1.29).

Определение 1.12. *Робастным потенциалом Поттса в классе* $\mathcal{P}^n$ называется потенциальная функция, представимая в виде

$$\phi(\mathbf{y}_\mathcal{C}) = \min\left\{ \min_{k\in\mathcal{K}}\left\{ \left(|\mathcal{C}| - \textstyle\sum_{v\in\mathcal{C}}[\![y_v = k]\!]\right)\dfrac{\delta_\mathcal{C} - \delta^k}{T} + \delta^k \right\}, \delta_\mathcal{C} \right\}, \tag{1.46}$$

где T — параметр отсечения, $\delta_\mathcal{C} \geq \delta^k,\ \forall k \in \mathcal{K}$. В отличие от (1.45), значение потенциала будет меньше, чем $\delta_\mathcal{C}$, если небольшое число переменных (меньше T) будет принадлежать доминирующей метке, при этом значение потенциала на участке от 0 до T ошибок задаётся линейной функцией. Это накладывает ограничение $2T < |\mathcal{C}|$. При $T = 1$ модель вырождается в неробастный потенциал Поттса в классе $\mathcal{P}^n$.

Потенциалы определённых выше классов могут быть приближённо минимизированы с помощью разрезов на графах. При запуске на энергии с такими потенциалами алгоритма α-расширения или $\alpha\beta$-замены возникающие на итерациях бинарные задачи могут быть решены с помощью построения разреза в графе с двумя дополнительными переменными на каждый потенциал высокого порядка [50].

Другим обобщением потенциала Поттса в классе $\mathcal{P}^n$ является потенциал, штрафующий количество различных меток, использованных при разметке подмножества переменных. Он часто используется в компьютерном зрении для регуляризации: например, в задаче восстановления геометрии сцены вероятнее конфигурация с меньшим числом плоскостей, а в задаче сегментации изображений — с меньшим числом классов или кластеров.

Определение 1.13. *Потенциалом, штрафующим наличие меток* называется потенциальная функция, представимая в виде

$$\phi(\mathbf{y}_\mathcal{C}) = \sum_{k \in \mathcal{K}} \delta^k [\![\exists v \in \mathcal{C} : y_v = k]\!], \tag{1.47}$$

где δ^k — штраф за присутствие метки k среди значений переменных с индексами $\mathcal{C}$.

Для минимизации энергии с потенциалами такого вида также может использоваться алгоритм α-расширения [52]. Пусть на некоторой итерации алгоритма необходимо сделать шаг расширения по метке α. Тогда для группы переменных $\mathcal{C}^\beta \in \mathcal{C}$, объединяющей все переменные, имеющие значение β в текущей разметке, необходимо добавить штраф δ^β, если хотя бы одна из этих переменных не изменит метку на α. При построении дополнительного графа для всех меток β, отличных от α и присутствующих в текущей разметке, в граф добавляется дополнительная вершина $\bar{v}^\beta$. Также добавляются дуги $v \to \bar{v}^\beta$, $\forall v \in \mathcal{C}^\beta$, и $\bar{v}^\beta \to t$, где t — вершина-сток, имеющие пропускную способность δ^β. Минимальный разрез в этом графе, так же как и ранее, соответствует точке минимума энергии в дополнительной бинарной задаче, которая трансформируется в изменение разметки на данном шаге.

В данной работе потенциалы такого вида используются при настройке параметров марковской сети по слабоаннотированным данным, то есть таким, для которых известна не разметка, а лишь некоторые её статистики (глава 2). Некоторые виды аннотации изображений приводят к свойствам генерируемых в процессе оптимизации «негативных» разметок, которые необходимо поддерживать: отсутствие меток в разметке изображения, «пустые» (относительно конкретной метки) строки или столбцы в разметке изображения. Хотя в обучаемой марковской сети нет факторов высокого порядка, при настройке параметров используется описанный алгоритм.

1.3 Обучение марковских сетей

Во многих приложениях марковских сетей потенциальные функции могут быть заданы экспертом, например, в задаче поиска стереосоответствия или восстановления изображений. Однако с усложнением структуры моделей [11], а также в задачах со сложными зависимостями между признаками и разметкой [2], приходится настраивать потенциалы по размеченной выборке. Этот процесс называется *обучением*.

Пусть задана некоторая обучающая выборка $\{(\mathbf{x}^j, \mathbf{y}^j)\}_{j=1}^J \in (\mathcal{X} \times \mathcal{Y})^J$. Поскольку при обучении признаки $\mathbf{x}$ и параметры $\mathbf{w}$ уже не фиксированы, нельзя игнорировать обуславливающие переменные в модели $\mathsf{P}(\mathbf{y} \mid \mathbf{x}, \mathbf{w})$. Дискриминативное обучение восстанавливает

это распределение в параметрическом виде. Таким образом, моделирование состоит из двух шагов: 1) формулировка условного распределения в параметрическом виде, 2) настройка параметров $\mathbf{w}^*$ так, что функции вероятностей $\mathsf{P}(\mathbf{y} \mid \mathbf{x}^j, \mathbf{w}^*)$ достигают бóльших значений на верных разметках $\mathbf{y}^j$, и меньших — на остальных, которая обычно осуществляется минимизацией некоторой целевой функции (эмпирического риска).

Определение 1.14. *Логлинейной* зависимостью условной вероятности от параметров называют следующую параметризацию:

$$\mathsf{P}(\mathbf{y} \mid \mathbf{x}, \mathbf{w}) = \frac{1}{Z(\mathbf{x}, \mathbf{w})} \exp\left(\sum_{f=1}^{F} \mathbf{w}^\intercal \boldsymbol{\psi}^{t(f)}(\mathbf{y}_{\mathcal{C}_f}; \mathbf{x}_f)\right) = \frac{1}{Z(\mathbf{x}, \mathbf{w})} \exp\big(\mathbf{w}^\intercal \boldsymbol{\psi}(\mathbf{y}; \mathbf{x})\big), \tag{1.48}$$

где, как и в (1.1), Z — нормировочная константа, $\mathbf{y}_{\mathcal{C}_f}$ — проекция $\mathbf{y}$ на индексы $\mathcal{C}_f$, $\mathbf{x}_f$ — часть вектора признаков, имеющая отношение к фактору f, $\boldsymbol{\psi}^{t(f)}(\mathbf{y}_{\mathcal{C}_f}; \mathbf{x}_f)$ — вектор *обобщённых признаков* фактора f, длина которого равна длине вектора параметров $\mathbf{w}$, а $\boldsymbol{\psi}(\mathbf{y}; \mathbf{x}) = \sum_{f=1}^{F} \boldsymbol{\psi}^{t(f)}(\mathbf{y}_{\mathcal{C}_f}; \mathbf{x}_f)$. Тип фактора $t(f)$ определяет тип зависимости, моделируемой вектором обобщённых признаков; например, он может разделять унарные и парные потенциалы. Логлинейная параметризация чаще всего используется на практике из-за простоты работы с ней.

Рассмотрим пример задания вектора обобщённых признаков и семантики параметров, который может быть использован в задаче семантической сегментации изображений, разбитых на суперпиксели. Разметка суперпикселей изображения моделируется парно-сепарабельной марковской сетью (1.5), то есть все факторы имеют порядок не более двух. Тогда имеются два типа факторов — унарные и парные: $t(f) \in \{\mathrm{v}, \mathrm{e}\}$. Пусть также заданы признаки суперпикселей $\{\mathbf{x}^{\mathrm{v}}_v\}_{v \in \mathcal{V}} \in \mathbb{R}^{d_{\mathrm{v}} \times |\mathcal{V}|}$ и их попарного взаимодействия $\{\mathbf{x}^{\mathrm{e}}_{vw}\}_{(v,w) \in \mathcal{E}} \in \mathbb{R}^{d_{\mathrm{e}} \times |\mathcal{E}|}$, которые все вместе при конкатенации дают вектор $\mathbf{x}$. Здесь d_{v} и d_{e} — размерности соответствующих векторов признаков. Вектор параметров разделяется на две части: параметры $\mathbf{w}^{\mathrm{v}}$ являются общими для всех унарных потенциалов, а $\mathbf{w}^{\mathrm{e}}$ — общими для всех парных потенциалов. $\mathbf{w}^{\mathrm{v}} \in \mathbb{R}^{K d_{\mathrm{v}}}$ содержит коэффициенты скалярного произведения для каждого из K значений, которые может принимать y_v. Функция $\boldsymbol{\psi}^{\mathrm{v}}(y_v; \mathbf{x}_v)$ возвращает элементы вектора $\mathbf{x}^{\mathrm{v}}_v$ в нужных позициях, чтобы они соответствовали признакам для назначения y_v, остальные позиции заполняются нулями. Аналогичная операция может быть проделана для парных потенциалов, с той разницей, что назначение (y_v, y_u) может принимать не K, а K^2 значений (рис. 1.3).

Возможны и другие варианты задания семантики параметров, даже для задачи семантической сегментации. Например, некоторым назначениям в потенциалах могут не соответствовать параметры, тем самым неявно предполагается нулевое значение потенциала. В некоторых задачах параметры могут не быть общими для разных факторов одного порядка, например, это полезно в задаче категоризации документов (раздел 2.3).

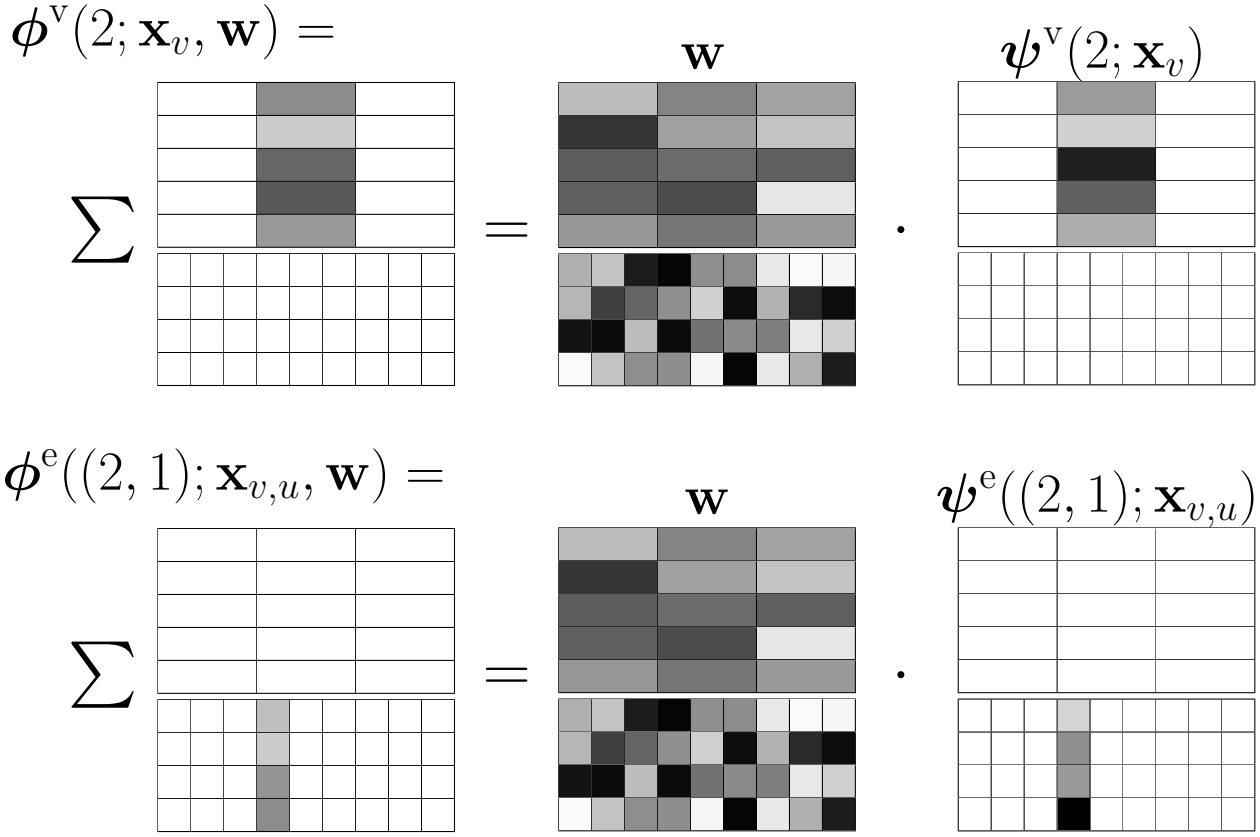


Рисунок 1.3: Пример определения унарных (верхний ряд) и парных (нижний ряд) потенциалов при логлинейной параметризации при количестве категорий $K = 3$, количестве признаков унарных потенциалов $d_{\mathrm{v}} = 5$ и количестве признаков парных потенциалов $d_{\mathrm{e}} = 4$ для конфигураций $y_v = 2$ и $y_v = 2, y_u = 1$. Векторы обобщённых признаков принимают ненулевые значения только в соответствующих «колонках», куда записываются значения $\mathbf{x}_v$ и $\mathbf{x}_{v,u}$, соответственно. Значение потенциала вычисляется как скалярное произведение параметров $\mathbf{w}$ на соответствующий вектор обобщённых признаков.

1.3.1 Максимизация правдоподобия и его приближений

Определение 1.15. *Функцией правдоподобия* параметров $\mathbf{w}$ семейства распределений $\mathsf{P}(\mathbf{y} \mid \mathbf{x}, \mathbf{w})$ на выборке $\{(\mathbf{x}^j, \mathbf{y}^j)\}_{j=1}^J$ называется следующий функционал:

$$\mathsf{L}(\mathbf{w}) = \prod_{j=1}^{J} \mathsf{P}(\mathbf{y}^j \mid \mathbf{x}^j, \mathbf{w}). \tag{1.49}$$

Метод максимального правдоподобия предлагает брать в качестве оценки параметров такую, которая максимизирует правдоподобие на обучающей выборке: $\mathbf{w}_{\mathrm{ML}} = \operatorname{argmax}_{\mathbf{w}} \mathsf{L}(\mathbf{w})$. На практике проще искать максимум логарифма правдоподобия: он достигается в той же точки из-за того, что логарифм монотонно возрастает на всей области определения.

Найдём градиент логарифма функции правдоподобия:

$$\frac{\partial \log \mathsf{L}}{\partial \mathbf{w}} = \sum_{j=1}^{J} \left[\boldsymbol{\psi}(\mathbf{y}^j; \mathbf{x}^j) - \frac{1}{Z(\mathbf{x}^j, \mathbf{w})} \frac{\partial Z(\mathbf{x}^j, \mathbf{w})}{\partial \mathbf{w}} \right] = \tag{1.50}$$

$$\sum_{j=1}^{J} \left[\boldsymbol{\psi}(\mathbf{y}^j; \mathbf{x}^j) - \frac{1}{Z(\mathbf{x}^j, \mathbf{w})} \sum_{\bar{\mathbf{y}}} \frac{\partial \exp\left(\mathbf{w}^\intercal \boldsymbol{\psi}(\bar{\mathbf{y}}; \mathbf{x}^j)\right)}{\partial \mathbf{w}} \right] =$$

$$\sum_{j=1}^{J} \left[\boldsymbol{\psi}(\mathbf{y}^j; \mathbf{x}^j) - \sum_{\bar{\mathbf{y}}} \boldsymbol{\psi}(\bar{\mathbf{y}}; \mathbf{x}^j) \mathsf{P}(\bar{\mathbf{y}} \mid \mathbf{x}^j, \mathbf{w}) \right] =$$

$$J \mathsf{E}_{\text{data}} \boldsymbol{\psi}(\mathbf{y}; \mathbf{x}) - \sum_{j=1}^{J} \mathsf{E}_{\text{model}} \boldsymbol{\psi}(\mathbf{y}; \mathbf{x}^j).$$

В итоговой формуле E_{data} обозначает выборочное математическое ожидание по обучающей выборке, а $\mathsf{E}_{\text{model}}$ — математическое ожидание по оцениваемому распределению при условии текущих параметров $\mathbf{w}$ и признаков данного объекта $\mathbf{x}^j$. Первое представляет собой вектор, элементы которого равны усреднённым по факторам обучающей выборки обобщённым признакам. Элементы второго — суммы по K^V разметкам, которые состоят из обобщённых признаков, вычисленных для данных разметок, с коэффициентами, равными вероятностям получить соответствующие разметки. Чтобы избежать суммирования экспоненциального числа слагаемых, перепишем выражение:

$$\mathsf{E}_{\text{model}} \boldsymbol{\psi}(\mathbf{y}; \mathbf{x}) = \sum_{\bar{\mathbf{y}}} \boldsymbol{\psi}(\bar{\mathbf{y}}; \mathbf{x}^j) \mathsf{P}(\bar{\mathbf{y}} \mid \mathbf{x}^j, \mathbf{w}) = \sum_{\bar{\mathbf{y}}} \sum_{f=1}^{F} \boldsymbol{\psi}^{t(f)}(\mathbf{y}_{\mathcal{C}_f}; \mathbf{x}_f) \mathsf{P}(\bar{\mathbf{y}} \mid \mathbf{x}^j, \mathbf{w}) = \tag{1.51}$$

$$\sum_{f=1}^{F} \sum_{\bar{\mathbf{y}}_{\mathcal{C}_f}} \sum_{\bar{\mathbf{y}}_{\mathcal{V} \setminus \mathcal{C}_f}} \boldsymbol{\psi}^{t(f)}(\mathbf{y}_{\mathcal{C}_f}; \mathbf{x}_f) \mathsf{P}(\bar{\mathbf{y}} \mid \mathbf{x}^j, \mathbf{w}) = \sum_{f=1}^{F} \sum_{\bar{\mathbf{y}}_{\mathcal{C}_f}} \boldsymbol{\psi}^{t(f)}(\mathbf{y}_{\mathcal{C}_f}; \mathbf{x}_f) \mathsf{P}(\bar{\mathbf{y}}_{\mathcal{C}_f} \mid \mathbf{x}^j, \mathbf{w}).$$

Маргинальные распределения $\mathsf{P}(\bar{\mathbf{y}}_{\mathcal{C}_f} \mid \mathbf{x}^j, \mathbf{w})$ могут быть эффективно рассчитаны для некоторых видов марковских сетей (см. раздел 1.1). Таким образом, если порядок факторов ограничен сверху небольшой константой, матожидание по модели может быть оценено довольно быстро (за линейное по числу факторов время, независимо от количества переменных в марковской сети), при условии что известны значения всех маргинальных распределений на факторы.

Найдём теперь гессиан правдоподобия:

$$\frac{\partial^2 \log \mathsf{L}}{\partial \mathbf{w} \partial \mathbf{w}^\intercal} = -\sum_{j=1}^{J} \sum_{\bar{\mathbf{y}}} \boldsymbol{\psi}(\bar{\mathbf{y}}; \mathbf{x}^j) \frac{\frac{\partial \exp(\mathbf{w}^\intercal \boldsymbol{\psi}(\bar{\mathbf{y}}; \mathbf{x}^j))}{\partial \mathbf{w}^\intercal} Z(\mathbf{x}^j, \mathbf{w}) - \frac{\partial Z(\mathbf{x}^j, \mathbf{w})}{\partial \mathbf{w}^\intercal} \exp\left(\mathbf{w}^\intercal \boldsymbol{\psi}(\bar{\mathbf{y}}; \mathbf{x}^j)\right)}{Z^2(\mathbf{x}^j, \mathbf{w})} = \tag{1.52}$$

$$-\sum_{j=1}^{J} \sum_{\bar{\mathbf{y}}} \boldsymbol{\psi}(\bar{\mathbf{y}}; \mathbf{x}^j) \frac{Z(\mathbf{x}^j, \mathbf{w}) \exp\left(\mathbf{w}^\intercal \boldsymbol{\psi}(\bar{\mathbf{y}}; \mathbf{x}^j)\right) \left[\boldsymbol{\psi}^\intercal(\bar{\mathbf{y}}; \mathbf{x}^j) - \mathsf{E}_{\text{model}} \boldsymbol{\psi}^\intercal(\mathbf{y}; \mathbf{x}^j)\right]}{Z^2(\mathbf{x}^j, \mathbf{w})} =$$

$$-\sum_{j=1}^{J} \left[\sum_{\bar{\mathbf{y}}} \frac{\exp(\mathbf{w}^\intercal \boldsymbol{\psi}(\bar{\mathbf{y}}; \mathbf{x}^j))}{Z(\mathbf{x}^j, \mathbf{w})} \boldsymbol{\psi}(\bar{\mathbf{y}}; \mathbf{x}^j) \boldsymbol{\psi}^\intercal(\bar{\mathbf{y}}; \mathbf{x}^j) - \mathsf{E}_{\text{model}} \boldsymbol{\psi}(\mathbf{y}; \mathbf{x}^j) \cdot \mathsf{E}_{\text{model}} \boldsymbol{\psi}^\intercal(\mathbf{y}; \mathbf{x}^j) \right] =$$

$$-\sum_{j=1}^{J} \left[\mathsf{E}_{\text{model}} \boldsymbol{\psi}(\mathbf{y}; \mathbf{x}^j) \boldsymbol{\psi}^\intercal(\mathbf{y}; \mathbf{x}^j) - \mathsf{E}_{\text{model}} \boldsymbol{\psi}(\mathbf{y}; \mathbf{x}^j) \cdot \mathsf{E}_{\text{model}} \boldsymbol{\psi}^\intercal(\mathbf{y}; \mathbf{x}^j) \right] =$$

$$-\sum_{j=1}^{J} \text{var}\left[\boldsymbol{\psi}(\mathbf{y}; \mathbf{x}^j)\right].$$

Таким образом, гессиан логарифма функции правдоподобия с точностью до знака равен сумме матриц ковариаций векторов обобщённых признаков по моделируемому распределению, следовательно, отрицательно определён. Логарифм правдоподобия вогнут, поэтому локальный максимум единственен, и может быть найден методами выпуклой оптимизации, такими как градиентный подъём, ньютоновские или квази-ньютоновские методы [36, §8.3]. На практике, однако же, при пересчёте градиента возникают вычислительные трудности при пересчёте $\mathsf{E}_{\text{model}} \boldsymbol{\psi}(\mathbf{y}; \mathbf{x}^j)$, которое предполагает суммирование $|\mathcal{Y}| = K^V$ слагаемых.

Один из выходов — оценивать это математическое ожидание на каждой итерации приближённо, с помощью выборки по значимости (англ. *importance sampling*) [36, §23.4]. Это метод сэмплирования, который концентрируется на слагаемых, вносящий наибольший вклад в математическое ожидание. Для его выполнения нужно генерировать значения из ненормированного распределения Гиббса, для чего используется метод Монте Карло с Марковскими цепями (англ. *Markov Chains Monte Carlo, MCMC*).

Вместо того, чтобы приближённо максимизировать логарифм правдоподобия, можно определить другую целевую функцию. Одним из вариантов является **псевдоправдоподобие** (англ. *pseudo likelihood*):

$$\mathsf{L}_{\text{P}}(\mathbf{w}) = \prod_{j=1}^{J} \prod_{v \in \mathcal{V}} \mathsf{P}(y_v^j \mid \mathbf{y}_{\setminus v}^j, \mathbf{x}^j, \mathbf{w}), \tag{1.53}$$

где $\mathbf{y}_{\setminus v}^j$ — вектор, состоящий из всех компонент $\mathbf{y}^j$, кроме y_v^j. Таким образом, полное правдоподобие заменяется на произведение условных вероятностей индивидуальных переменных при известных верных значениях остальных. Вычислить такие условные распределения проще, так как нормировочные константы теперь рассчитываются не для всех переменных, а для групп смежных вершин марковской сети каждой из переменных, как правило, небольших по размеру. Используя факторизацию марковской сети, имеем:

$$\mathsf{P}(y_v^j = k \mid \mathbf{y}_{\setminus v}^j, \mathbf{x}^j, \mathbf{w}) = \frac{p_v^j(k)}{\sum_{\bar{k}=1}^{K} p_v^j(\bar{k})}, \tag{1.54}$$

где ненормированное условное распределение равно:

$$p_v^j(k) = \prod_{f: v \in \mathcal{C}_f} \exp\left(\mathbf{w}^\intercal \boldsymbol{\psi}^{t(f)}\left(\mathbf{y}^j_{\mathcal{C}_f}|_{y_v=k}; \mathbf{x}^j_f\right)\right), \tag{1.55}$$

где под $\mathbf{y}^j_{\mathcal{C}_f}|_{y_v=k}$ понимается вектор, в котором компонента, соответствующая индексу переменной v, заменена на k.

Покажем, что вычисление такой функции и её градиента значительно проще на примере парно-сепарабельной марковской сети над графом, заданным 4-связной решёткой (такие марковские сети часто используются в низкоуровневой обработке изображений, см. рис. 1.2a). Каждая вершина графа входит не более чем в 4 фактора, поэтому в знаменателе (1.54) необходимо сложить не более K^4 слагаемых для каждой из переменных, то есть совершить порядка VK^4 операций для подсчёта градиента по одному объекту, что значительно меньше K^V.

Данный метод при обучении считает известными метки других переменных, что может привести к получению смещённой оценки. Например, в обучающей выборке значение переменной может точно определяться значением одной из «соседних» переменных, в результате чего распределение будет моделировать детерминистическую связь между ними, при этом не учитывая другие факторы, например локальные признаки. Однако на практике оценки максимума псевдоправдоподобия часто близки к оценкам максимума правдоподобия [36, §19.5.4].

К логарифму правдоподобия или его аппроксимациям часто добавляют квадратичный регуляризатор на параметры $\mathbf{w}^\intercal\mathbf{w}$, что эквивалентно введению нормального априорного распределения с центром в нуле. Эта техника предотвращает чрезмерный рост параметров, соответствующих наиболее сильным признакам, и таким образом предотвращает переобучение. Тогда оптимум такой регуляризованной функции можно считать MAP-оценкой на параметры $\mathbf{w}$. Градиент квадратичного регуляризатора линеен, поэтому не доставляет трудностей при оптимизации.

1.3.2 Максимизация отступа

Градиент логарифма правдоподобия (1.50) равен разности математического ожидания обобщённых признаков по данным и по модели. Таким образом, максимизация правдоподобия стремится увеличить значение функции распределения в точках, присутствующих в обучающей выборке, и уменьшить во всех остальных. Если на этапе вывода интерес представляет не распределение само по себе, а лишь его MAP-оценка, подбор параметров можно проводить из других соображений: объекты обучающей выборки должны иметь вероятность моды распределения, причём эта вероятность должна иметь как можно больший отступ от второй по вероятности точки. При использовании такого критерия игнорируются значения функции распределения во всех точках, кроме этих двух, что напоминает концепцию *опорных векторов* в методе опорных векторов (англ. *support vector machine, SVM*).

Второй важной особенностью метода является использование нетривиальной функции потерь при обучении. В разделе 1.1 было определено Байесовское решающее правило, позволяющее выводить разметку, минимизирующую математическое ожидание функции потерь по

апостериорному распределению. Хотя на этапе принятия решения это правило используется редко из-за его трудоёмкости, оказывается выгодно использовать его в целевой функции при обучении. Предположим, что задана функция потерь $\mathcal{L}(\bar{\mathbf{y}}; \mathbf{y})$, задающая отклонение некоторой разметки $\bar{\mathbf{y}}$ от верной разметки $\mathbf{y}$. Тогда задача обучения ставится как минимизация по параметрам $\mathbf{w}$ следующей целевой функции:

$$R(\mathbf{w}) = \prod_{j=1}^{J} \sum_{\bar{\mathbf{y}} \in \mathcal{Y}} \mathcal{L}(\bar{\mathbf{y}}; \mathbf{y}^j) \mathsf{P}(\bar{\mathbf{y}} \mid \mathbf{x}^j, \mathbf{w}). \tag{1.56}$$

Заметим, что при $\mathcal{L}(\bar{\mathbf{y}}; \mathbf{y}^j) = [\![\bar{\mathbf{y}} = \mathbf{y}^j]\!]$, $R(\mathbf{w})$ эквивалентна правдоподобию (1.49). Обозначим $\Delta(\bar{\mathbf{y}}; \mathbf{y}) = \log \mathcal{L}(\bar{\mathbf{y}}; \mathbf{y})$ и преобразуем логарифм $R(\mathbf{w})$:

$$\log R(\mathbf{w}) = \sum_{j=1}^{J} \log \left[\sum_{\bar{\mathbf{y}} \in \mathcal{Y}} \mathcal{L}(\bar{\mathbf{y}}; \mathbf{y}^j) \frac{\exp\left(\mathbf{w}^\intercal \boldsymbol{\psi}(\bar{\mathbf{y}}; \mathbf{x}^j)\right)}{Z(\mathbf{x}^j, \mathbf{w})} \right] = \tag{1.57}$$

$$\sum_{j=1}^{J} \left[\log \sum_{\bar{\mathbf{y}} \in \mathcal{Y}} \exp\left(\Delta(\bar{\mathbf{y}}; \mathbf{y}^j) + \mathbf{w}^\intercal \boldsymbol{\psi}(\bar{\mathbf{y}}; \mathbf{x}^j)\right) - \log \sum_{\bar{\mathbf{y}} \in \mathcal{Y}} \exp\left(\mathbf{w}^\intercal \boldsymbol{\psi}(\bar{\mathbf{y}}; \mathbf{x}^j)\right) \right].$$

Каждое слагаемое является суммой выпуклой и вогнутой по $\mathbf{w}$ функций, а также содержит сумму по K^V слагаемым. Построим выпуклую оценку сверху на $\log R(\mathbf{w})$, заменяя суммы по $\bar{\mathbf{y}}$ точечной оценкой. Для этого воспользуемся следующими оценками конструкции $\log \sum_{\bar{\mathbf{y}} \in \mathcal{Y}} \exp\left(f(\bar{\mathbf{y}})\right)$ для произвольных $f : \mathcal{Y} \to \mathbb{R}$ и $\mathbf{y} \in \mathcal{Y}$ [36, (19.85), (19.88)]:

$$f(\mathbf{y}) \leq \log \sum_{\bar{\mathbf{y}} \in \mathcal{Y}} \exp f(\bar{\mathbf{y}}) \leq \log \left[|\mathcal{Y}| \exp\left(\max_{\bar{\mathbf{y}} \in \mathcal{Y}} f(\bar{\mathbf{y}}) \right) \right] \leq \log |\mathcal{Y}| + \max_{\bar{\mathbf{y}} \in \mathcal{Y}} f(\bar{\mathbf{y}}). \tag{1.58}$$

Получим:

$$\log R(\mathbf{w}) \leq \sum_{j=1}^{J} \left[\max_{\bar{\mathbf{y}} \in \mathcal{Y}} \left\{ \Delta(\bar{\mathbf{y}}; \mathbf{y}^j) + \mathbf{w}^\intercal \boldsymbol{\psi}(\bar{\mathbf{y}}; \mathbf{x}^j) \right\} - \mathbf{w}^\intercal \boldsymbol{\psi}(\mathbf{y}^j; \mathbf{x}^j) \right] + J \log |\mathcal{Y}|. \tag{1.59}$$

Максимум конечного числа линейных функций является выпуклой функцией, поэтому полученная верхняя оценка является выпуклой. Последний член не зависит от $\mathbf{w}$, поэтому не влияет на точку, в которой достигается минимум. Добавление квадратичного регуляризатора даёт следующую целевую функцию:

$$\mathsf{L}_{\text{ММ}}(\mathbf{w}) = \frac{1}{2} \mathbf{w}^\intercal \mathbf{w} + C \sum_{j=1}^{J} \left[\max_{\bar{\mathbf{y}} \in \mathcal{Y}} \left\{ \Delta(\bar{\mathbf{y}}; \mathbf{y}^j) + \mathbf{w}^\intercal \boldsymbol{\psi}(\bar{\mathbf{y}}; \mathbf{x}^j) \right\} - \mathbf{w}^\intercal \boldsymbol{\psi}(\mathbf{y}^j; \mathbf{x}^j) \right], \tag{1.60}$$

где структурный параметр C определяет относительный вклад регуляризатора. Минимизацию этой функции можно также представить как задачу условной оптимизации путём введения фиктивных переменных ξ_j. Полученная задача известна как *структурный метод опорных векторов* (англ. *structural support vector machine, SSVM*) [20]:

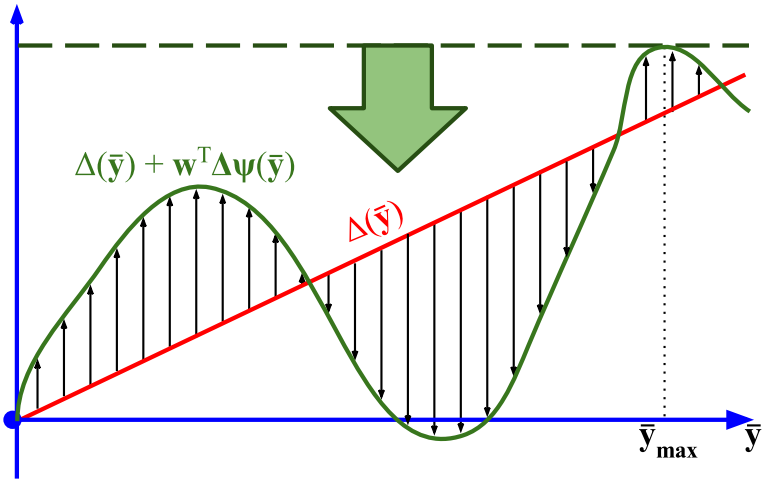


Рисунок 1.4: Пример, поясняющий идею максимизации отступа в структурном обучении для объекта обучающей выборки $(\mathbf{x}, \mathbf{y})$. Горизонтальная ось представляет пространство разметок. Красная кривая задаёт функцию потерь $\Delta(\bar{\mathbf{y}}; \mathbf{y})$, чёрные стрелки задают величину $\mathbf{w}^\intercal \boldsymbol{\Delta\psi}(\bar{\mathbf{y}}; \mathbf{x}) = \mathbf{w}^\intercal\big(\boldsymbol{\psi}(\bar{\mathbf{y}}; \mathbf{x}) - \boldsymbol{\psi}(\mathbf{y}; \mathbf{x})\big)$, а зелёная кривая — их сумму (в подписях опущены постоянные параметры функций). Минимизация отступа стремится минимизировать по $\mathbf{w}$ значение этой суммы в смысле нормы L_∞. На рисунке показана точка максимума этой кривой $\bar{\mathbf{y}}_{\max}$, не совпадающая с точкой максимума функции, показанной чёрными стрелками.

Оптимизационная задача 1.5 (структурный SVM)**.**

$$\min_{\mathbf{w},\boldsymbol{\xi}} \frac{1}{2}\mathbf{w}^\intercal\mathbf{w} + C\sum_{j=1}^{J}\xi_j, \tag{1.61}$$

$$\text{при условиях} \quad \mathbf{w}^\intercal\boldsymbol{\psi}(\mathbf{y}^j; \mathbf{x}^j) \geq \max_{\bar{\mathbf{y}}\in\mathcal{Y}} \big\{\mathbf{w}^\intercal\boldsymbol{\psi}(\bar{\mathbf{y}}; \mathbf{x}^j) + \Delta(\bar{\mathbf{y}}; \mathbf{y}^j)\big\} - \xi_j, \quad \forall j \in \{1, \ldots, J\}. \tag{1.62}$$

Из смысла задачи $\Delta(\mathbf{y}; \mathbf{y}) = 0$, и $\Delta(\bar{\mathbf{y}}; \mathbf{y}) \geq 0, \forall\bar{\mathbf{y}}$. Одним из вариантов выбора функции потерь для задач разметки является расстояние Хэмминга: $\Delta(\bar{\mathbf{y}}; \mathbf{y}) = \sum_v [\![\bar{y}_v \neq y_v]\!]$. В случае выполнения всех условий значение функционала $\mathbf{w}^\intercal\boldsymbol{\psi}(\mathbf{y}^j; \mathbf{x}^j)$ на верной разметке $\mathbf{y}^j$ должно быть больше, чем на любой другой разметке $\bar{\mathbf{y}}$ (с допуском ξ_j), причём отступ должен увеличиваться с удалением разметки $\bar{\mathbf{y}}$ от верной. Поэтому подбор параметров из таких соображений называется *максимизацией отступа* между верной разметкой и второй после неё в соответствии с обученной моделью. Рис. 1.4 иллюстрирует эту идею.

Оптимизация методом секущей плоскости

Задачу 1.5 можно представить в форме стандартной задачи квадратичного программирования с линейными ограничениями:

Алгоритм 1.1 Обучение SSVM методом секущей плоскости

1: **Вход:** обучающая выборка $\{(\mathbf{x}^j, \mathbf{y}^j)\}_{j=1}^J$, гиперпараметры C, ε.
2: **Выход:** параметры $\mathbf{w}$.
3: $\mathcal{W}_j \leftarrow \varnothing$, $\xi_j \leftarrow 0,\ \forall j \in \{1, \ldots, J\}$
4: **repeat**
5: **for all** $j \in \{1, \ldots, J\}$ **do**
6: $\bar{\mathbf{y}} \leftarrow \operatorname{argmax}_{\mathbf{y} \in \mathcal{Y}} \{\mathbf{w}^\intercal \boldsymbol{\psi}(\mathbf{y}; \mathbf{x}^j) + \Delta(\mathbf{y}; \mathbf{y}^j)\}$
7: $v_j \leftarrow \mathbf{w}^\intercal \boldsymbol{\psi}(\bar{\mathbf{y}}; \mathbf{x}^j) - \mathbf{w}^\intercal \boldsymbol{\psi}(\mathbf{y}^j; \mathbf{x}^j) + \Delta(\bar{\mathbf{y}}; \mathbf{y}^j) - \xi_j$
8: **if** $v_j \geq \varepsilon$ **then**
9: $\mathcal{W}_j \leftarrow \mathcal{W}_j \cup \{\bar{\mathbf{y}}\}$
10: $(\mathbf{w}, \boldsymbol{\xi}) \leftarrow \operatorname{argmin}_{\mathbf{w}, \boldsymbol{\xi} \geq 0} \frac{1}{2} \mathbf{w}^\intercal \mathbf{w} + C \sum_{j=1}^J \xi_j$
11: п. у. $\mathbf{w}^\intercal \boldsymbol{\psi}(\mathbf{y}^i; \mathbf{x}^i) \geq \mathbf{w}^\intercal \boldsymbol{\psi}(\bar{\mathbf{y}}^i; \mathbf{x}^i) + \Delta(\bar{\mathbf{y}}^i; \mathbf{y}^i) - \xi_j,\ \forall \bar{\mathbf{y}}^i \in \mathcal{W}_i,\ \forall i \in \{1, \ldots, J\}$
12: **end if**
13: **end for**
14: **until** $v_j < \varepsilon,\ \forall j \in \{1, \ldots, J\}$

Оптимизационная задача 1.6 (SSVM как задача квадратичного программирования)**.**

$$\min_{\mathbf{w}, \boldsymbol{\xi}} \frac{1}{2} \mathbf{w}^\intercal \mathbf{w} + C \sum_{j=1}^J \xi_j, \tag{1.63}$$

при условиях $\quad \mathbf{w}^\intercal \boldsymbol{\psi}(\mathbf{y}^j; \mathbf{x}^j) \geq \mathbf{w}^\intercal \boldsymbol{\psi}(\bar{\mathbf{y}}; \mathbf{x}^j) + \Delta(\bar{\mathbf{y}}; \mathbf{y}^j) - \xi_j,\ \forall \bar{\mathbf{y}} \in \mathcal{Y},\ \forall j \in \{1, \ldots, J\}. \tag{1.64}$

Для задач квадратичного программирования общего вида существует множество стандартных решателей, однако на практике с их помощью невозможно решить задачу 1.6, так как она содержит JK^V линейных ограничений. Поэтому разработаны специальные алгоритмы для решения задач такого вида. Одним из подходов является применение *метода секущей плоскости* (англ. *cutting-plane method*). В подобных алгоритмах (см. алгоритм 1.1) многогранник, определяющий допустимое множество, уточняется итеративно. На итерации t находится точка $\mathbf{w}_t$, минимизирующая сокращённую задачу квадратичного программирования на рабочем наборе ограничений, являющемся подмножеством полного набора. Это можно выполнить с помощью стандартного решателя. Затем ограничивающий политоп уточняется путём добавления набора наиболее нарушаемых ограничений полной задачи при текущих параметрах $\mathbf{w}_t$. Алгоритм прекращает работу, когда все ограничения выполняются с точностью до ε. Показано, что при фиксированном $\varepsilon > 0$ алгоритм сходится за полиномиальное число итераций [20].

Наиболее нарушаемое ограничение в строке 6 алгоритма 1.1 находится посредством алгоритма оптимизации, называемого *выводом, дополненным функцией потерь*. Часто оптимизация может проводиться теми же средствами, которые применяются при выводе MAP-оценки. Для этого функция потерь должна иметь вид, пригодный для такого вывода. Например, если функция потерь разделяется на унарные потенциалы (как расстояние Хэмминга), вывод, дополненный функцией потерь не становится сложнее вывода MAP-оценки. Другим примером могут являться функции потерь высоких порядков специального вида, допускающие эффективный вывод с помощью алгоритмов на основе разрезов графов (раздел 1.2.4). В главе 2 приведены примеры нетривиальных функций потерь, возникающих в практических задачах.

Субградиентные методы оптимизации

Целевую функцию (1.60) можно оптимизировать и напрямую. Она является выпуклой, но недифференцируемой. Поэтому можно применить метод субградиентного спуска. Субградиент может быть вычислен по формуле:

$$\frac{\partial \mathsf{L}_{\text{ММ}}}{\partial \mathbf{w}} \ni \mathbf{g}(\mathbf{w}) = \mathbf{w} + C \sum_{j=1}^{J} \left[\boldsymbol{\psi}(\bar{\mathbf{y}}^j(\mathbf{w}); \mathbf{x}^j) - \boldsymbol{\psi}(\mathbf{y}^j; \mathbf{x}^j) \right], \tag{1.65}$$

где $\bar{\mathbf{y}}^j(\mathbf{w}) = \operatorname{argmax}_{\bar{\mathbf{y}} \in \mathcal{Y}} \{\Delta(\bar{\mathbf{y}}; \mathbf{y}^j) + \mathbf{w}^\intercal \boldsymbol{\psi}(\bar{\mathbf{y}}; \mathbf{x}^j)\}$ при текущем значении $\mathbf{w}$. Инициализировав вектор параметров некоторым значением $\mathbf{w}^0$, метод итеративно обновляет его значения по формуле

$$\mathbf{w}^{n+1} = \mathbf{w}^n - \gamma_n \mathbf{g}(\mathbf{w}^n), \tag{1.66}$$

где γ_n — убывающий размер шага. Поскольку целевая функция выпукла, существует такая последовательность $\{\gamma_n\}$, при которой оптимизация сходится к глобальному оптимуму. В частности, достаточно, чтобы $\gamma_n \to 0$, но $\sum_{n=0}^{\infty} \gamma_n \to +\infty$ [53]. Например, такому свойству удовлетворяет последовательность $\gamma_n = \frac{1}{n+1}$. В практических задачах важна скорость сходимости, которая сильно зависит от выбора конкретной последовательности размеров шагов.

На практике бывает полезно ограничивать множество $\mathbf{w}$. Например, при использовании ассоциативных марковских сетей приходится полагать $\mathbf{w} \geq \mathbf{0}$ (см. раздел 2.2.1). Если на каждой итерации брать проекцию $\mathbf{w}^{n+1}$ на некоторое выпуклое множество, то метод сходится к оптимуму целевой функции на этом выпуклом множестве [53].

Лакост-Жулие и др. [54] рассмотрели субградиентный метод для оптимизации двойственной функции к (1.60). Формулы пересчёта, выраженные через целевые переменные прямой задачи, совпали с (1.66), однако удалось получить в аналитическом виде оптимальный размер шага γ_n на каждой итерации n. Кроме того, появилась возможность вычислять текущий интервал двойственности, который является верхней оценкой отклонения значения целевой функции в текущей точке от оптимума.

Исследования других применений субградиентного метода [55,56] показали, что неэффективно оценивать градиент точно на каждой итерации. Сумму по J объектам в (1.65) можно приблизить суммой по их случайному подмножеству:

$$\tilde{\mathbf{g}}(\mathbf{w}) = \mathbf{w} + C \frac{|\mathcal{J}|}{J} \sum_{j \in \mathcal{J}} \left[\boldsymbol{\psi}(\bar{\mathbf{y}}^j(\mathbf{w}); \mathbf{x}^j) - \boldsymbol{\psi}(\mathbf{y}^j; \mathbf{x}^j) \right], \tag{1.67}$$

где $\mathcal{J} \subset \{1, \ldots, J\}$ — некоторое случайное подмножество. В вырожденном случае $\mathcal{J}$ состоит из одного элемента, тогда метод называют *онлайн-обучением*. При его использовании скорость сходимости субградиентных методов может конкурировать со скоростью метода секущей плоскости [54].

1.3.3 Обучение нелинейных моделей

До этого предполагалась логлинейная зависимость (1.48) правдоподобия от параметров распределения. Далее рассмотрим графические вероятностные модели с более гибкой зависимостью от параметров, однако явно представимые в виде распределения Гиббса (1.1). Существуют также методы, модифицирующие алгоритмы вывода и настраивающие непосредственно их параметры, не используя в явном виде параметризованное распределение [25, 26, 57], в том числе попадающий в этот класс метод, предложенный в главе 4. Более подробный обзор таких методов дан в разделе 4.4.

Функциональный градиентный бустинг

Метод *функционального градиентного бустинга* (англ. *functional gradient boosting*) [58] использует идею субградиентного обучения, однако не рассматривает параметрическое представление потенциалов, а выполняет градиентный спуск непосредственно в функциональном пространстве. Рассмотрим нерегуляризованный аналог целевой функции (1.60) для обучения потенциальных функций энергии (1.4):

$$\mathsf{L}_{\text{FGB}} = \sum_{j=1}^{J} \left[\max_{\bar{\mathbf{y}} \in \mathcal{Y}} \left\{ \Delta(\bar{\mathbf{y}}; \mathbf{y}^j) - \sum_{f=1}^{F} \phi_f(\bar{\mathbf{y}}_{\mathcal{C}_f}; \mathbf{x}^j) \right\} + \sum_{f=1}^{F} \phi_f(\mathbf{y}^j_{\mathcal{C}_f}; \mathbf{x}^j) \right], \tag{1.68}$$

причём потенциал $\phi_f(\mathbf{y}_{\mathcal{C}_f}; \mathbf{x})$ ищется в виде некоторой функции от d обобщённых признаков $\boldsymbol{\psi}^{t(f)}(\mathbf{y}_{\mathcal{C}_f}; \mathbf{x}_f)$ (они могут задаваться так же, как и ранее, см. пример после определения 1.14). Пусть на итерации градиентного подъёма n подобрана функция $g_n : \mathbb{R}^d \to \mathbb{R}$, определяющая потенциалы: $\phi_f(\mathbf{y}_{\mathcal{C}_f}; \mathbf{x}) = g_n\big(\boldsymbol{\psi}^{t(f)}(\mathbf{y}_{\mathcal{C}_f}; \mathbf{x}_f)\big)$, $\forall f$. Чтобы получить функциональный градиент в этой точке, покажем сначала, чему равен градиент функционала $\mathsf{F}_{\mathbf{a}}[g] \equiv g(\mathbf{a})$. Рассмотрим значение функционала при вариации аргумента:

$$\mathsf{F}_{\mathbf{a}}[g + \varepsilon\eta] = g(\mathbf{a}) + \varepsilon\eta(\mathbf{a}) = g(\mathbf{a}) + \varepsilon \int \eta(\mathbf{x})\lambda_{\mathbf{a}}(\mathbf{x})d\mathbf{x} + \mathsf{O}(\varepsilon^2), \tag{1.69}$$

где $\lambda_{\mathbf{a}}(\boldsymbol{\xi}) \equiv \delta(\boldsymbol{\xi} - \mathbf{a})$ — дельта-функция Дирака в точке $(\boldsymbol{\xi} - \mathbf{a})$. Из определения функционального градиента, $\frac{\delta \mathsf{F}_{\mathbf{a}}}{\delta g} = \lambda_{\mathbf{a}}$.

Получим теперь отрицательный функциональный градиент функции (1.68) в точке g_n [6]:

$$-\frac{\delta \mathsf{L}_{\text{FGB}}}{\delta g_n} = \sum_{j=1}^{J} \left[\sum_{f=1}^{F} \frac{\delta}{\delta g_n} g_n \left(\boldsymbol{\psi}^{t(f)}(\mathbf{y}^{*j}_{\mathcal{C}_f}; \mathbf{x}^j_f) \right) - \sum_{f=1}^{F} \frac{\delta}{\delta g_n} g_n \left(\boldsymbol{\psi}^{t(f)}(\mathbf{y}^{j}_{\mathcal{C}_f}; \mathbf{x}^j_f) \right) \right] \tag{1.70}$$

$$= \sum_{j=1}^{J} \sum_{f=1}^{F} \left[\lambda_{\boldsymbol{\psi}^{t(f)}(\mathbf{y}^{*j}_{\mathcal{C}_f}; \mathbf{x}^j_f)} - \lambda_{\boldsymbol{\psi}^{t(f)}(\mathbf{y}^{j}_{\mathcal{C}_f}; \mathbf{x}^j_f)} \right], \tag{1.71}$$

где $\mathbf{y}^{*j} = \operatorname{argmax}_{\bar{\mathbf{y}} \in \mathcal{Y}} \left\{ \Delta(\bar{\mathbf{y}}; \mathbf{y}^j) - \sum_{f=1}^{F} g_n \left(\boldsymbol{\psi}^{t(f)}(\bar{\mathbf{y}}_{\mathcal{C}_f}; \mathbf{x}^j_f) \right) \right\}$.

В случае, когда градиентный подъём выполняется в Евклидовом пространстве, делается шаг по направлению отрицательного градиента, а финальное решение может быть представ-

лено как взвешенная сумма градиентов, найденных в процессе оптимизации. При восстановлении функциональной зависимости g от обобщённых признаков эту стратегию использовать нельзя, так как сумма дельта-функций будет обладать низкой обобщающей способностью: для большинства признаков тестовой выборки она будет равняться нулю, так как такие признаки потенциалов не встречались в обучающей выборке, зато при случайном совпадении признаков одного из потенциалов энергия устремится в бесконечность. Поэтому для регуляризации на каждой итерации функционального градиентного бустинга будем настраивать функцию-предиктор из некоторого множества $\mathcal{H}$, наилучшим образом приближающую отрицательный функциональный градиент (1.70) в смысле скалярного произведения.

Определение 1.16. *Скалярным произведением в* L_2 называют следующий вещественнозначный функционал:

$$\langle f, g \rangle = \int f(\xi) g(\xi) d\xi. \tag{1.72}$$

Найдём функцию $h_n^* : \mathbb{R}^d \to \mathbb{R}$ в классе $\mathcal{H}$, наиболее близкую к отрицательному градиенту в смысле скалярного произведения в L_2:

$$\begin{aligned} h_n^* = \operatorname*{argmax}_{h_n \in \mathcal{H}} \left\langle h_n, -\frac{\delta \mathsf{L}_{\text{FGB}}}{\delta g_n} \right\rangle &= \operatorname*{argmax}_{h_n \in \mathcal{H}} \sum_{j=1}^{J} \sum_{f=1}^{F} \left[\left\langle h_n, \lambda_{\boldsymbol{\psi}^{t(f)}(\mathbf{y}^{*j}_{\mathcal{C}_f}; \mathbf{x}^j_f)} \right\rangle - \left\langle h_n, \lambda_{\boldsymbol{\psi}^{t(f)}(\mathbf{y}^{j}_{\mathcal{C}_f}; \mathbf{x}^j_f)} \right\rangle \right] \\ &= \operatorname*{argmax}_{h_n \in \mathcal{H}} \sum_{j=1}^{J} \sum_{f=1}^{F} \left[h_n\big(\boldsymbol{\psi}^{t(f)}(\mathbf{y}^{*j}_{\mathcal{C}_f}; \mathbf{x}^j_f)\big) - h_n\big(\boldsymbol{\psi}^{t(f)}(\mathbf{y}^{j}_{\mathcal{C}_f}; \mathbf{x}^j_f)\big) \right]. \end{aligned} \tag{1.73}$$

Класс $\mathcal{H}$ может представлять собой семейство бинарных классификаторов, т.е. множество функций, возвращающих $+1$ или -1. Например, это может быть множество линейных классификаторов или решающих деревьев. Тогда максимизация в (1.73) эквивалентна обучению соответствующего классификатора, где $\boldsymbol{\psi}^{t(f)}(\mathbf{y}^{*j}_{\mathcal{C}_f}; \mathbf{x}^j_f)$ представляют собой объекты класса $+1$, а $\boldsymbol{\psi}^{t(f)}(\mathbf{y}^{j}_{\mathcal{C}_f}; \mathbf{x}^j_f)$ — объекты класса -1, для всех объектов обучающей выборки j и их факторов f. Функция h_n^* может быть найдена, например, с помощью алгоритма логистической регрессии или индукции решающих деревьев.

Осталось определить, как обновляется функция g_n на каждом следующем шаге градиентного спуска:

$$g_{N+1}(\boldsymbol{\xi}) = g_N(\boldsymbol{\xi}) + \gamma_N h_N^*(\boldsymbol{\xi}) = \sum_{n=1}^{N} \gamma_n h_n^*(\boldsymbol{\xi}). \tag{1.74}$$

Здесь γ_n — убывающая последовательность длин шагов, а начальное значение можно положить нулевым: $g_0(\boldsymbol{\xi}) \equiv 0$. Финальные значения потенциальных функций определяются значением функции g_n после N итераций:

$$\phi_f(\mathbf{y}_{\mathcal{C}_f}; \mathbf{x}) = g_{N+1}\left(\boldsymbol{\psi}^{t(f)}(\mathbf{y}_{\mathcal{C}_f}; \mathbf{x}_f)\right) = \sum_{n=1}^{N} \gamma_n h_n^*\left(\boldsymbol{\psi}^{t(f)}(\mathbf{y}_{\mathcal{C}_f}; \mathbf{x}_f)\right), \quad \forall f. \tag{1.75}$$

Поле решающих деревьев

В поле решающих деревьев (англ. *decision tree field*) [59] используется другой вид нелинейной зависимости потенциальных функций от признаков. Каждому типу факторов t соответствует решающее дерево $\mathbf{T}_t$, вершинам q которого соответствуют векторы параметров $\mathbf{w}_q^t$, определяющие значение потенциала для каждой из конфигураций меток. Обозначим $path(\mathbf{x}_f; \mathbf{T}_t)$ функцию, возвращающую для признаков фактора $\mathbf{x}_f$ множество вершин решающего дерева $\mathbf{T}_t$, «посещённых» при классификации объекта с признаками $\mathbf{x}_f$. Тогда значение потенциальной функции определяется следующим образом:

$$\phi_f(\mathbf{y}_{\mathcal{C}_f}; \mathbf{x}) \quad = \sum_{q \in path\left(\mathbf{x}_f; \mathbf{T}_{t(f)}\right)} w_q^{t(f)}(\mathbf{y}_{\mathcal{C}_f}), \quad \forall f. \tag{1.76}$$

Обучение модели состоит из двух стадий: сначала определяется структура решающих деревьев, затем настраиваются параметры. Для определения структуры дерева $\mathbf{T}_t$ используется алгоритм индукции решающих деревьев, где в качестве признаков используются все признаки $\mathbf{x}_f$ факторов соответствующего типа t, а в качестве правильных ответов используются векторы их правильных разметок $\mathbf{y}_{\mathcal{C}_f}$, причём каждый из $K^{|\mathcal{C}_f|}$ векторов считается отдельным классом (предполагается, что все факторы одного типа имеют равный порядок). Затем структура дерева фиксируется, а гистограммы категорий в вершинах обнуляются — вместо них на следующей стадии настраиваются коэффициенты $\mathbf{w}_q^t$.

Ключевым наблюдением является линейная зависимость энергии марковской сети от параметров, которая в свою очередь вытекает из линейности потенциалов (1.76) по $\mathbf{w}$. Из этого следует, что правдоподобие (1.49) выпукло и дифференцируемо по $\mathbf{w}$, однако оно не может быть оптимизировано с помощью градиентных методов из-за невозможности эффективно вычислять нормировочную константу. Вместо этого предлагается максимизировать L_2-регуляризованный логарифм псевдоправдоподобия (1.53). Регуляризация важна, так как она ведёт к уменьшению модуля параметров листовых вершин решающих деревьев и близких к ним. Для их настройки используется небольшое количество данных, поэтому предпочтительно для определения потенциалов использовать более близкие к корню вершины, так как их параметры настраиваются надёжнее.

Поскольку число классов, используемых при определении структуры решающих деревьев, экспоненциально зависит от порядка факторов, метод не позволяет использовать потенциалы высоких порядков. Однако использование различных типов факторов позволяет учитывать дальнодействующие зависимости между метками, например с помощью задания регулярной структуры отступов в каждом пикселе изображения, в которой каждому отступу соответствует тип фактора. Подобная идея используется при задании д-факторов в главе 4 данной работы.

Глава 2

Использование различных типов аннотации обучающей выборки

При обучении алгоритмов разметки зачастую представляет сложность аннотация обучающей выборки — она требует значительных человеческих усилий. В отличие от полной (*сильной*) разметки, бывает проще получить *слабую аннотацию*, под которой мы понимаем некоторую статистику от полной разметки. Например, при решении задачи семантического разбора предложений слабая аннотация обучающей выборки может быть представлена разметкой частей речи, а в задаче категоризации документов может использоваться неполная разметка, в которой часть категорий (тэгов) для каждого из документов пропущены. В ряде задач анализа видеопоследовательностей разметка может быть дана только для ключевых кадров.

В этой главе целевыми приложениями являются задачи семантической сегментации изображений и категоризации документов. Примерами слабых аннотаций в первой служат метки изображения, которые отражают присутствие или отсутствие категорий; метки площади, которые содержат число пикселей каждой категории на изображении; набор плотных рамок для объектов, присутствующих в разметке; а также набор зёрен — подмножеств координат пикселей, принадлежащих объектам (рис. 2.1). Использование слабых типов аннотаций в этой задаче обуславливается практической целесообразностью. Например, в наборе данных PASCAL VOC 2012[1] только 2913 из 11540 (25%) изображений размечены полностью, для остальных известны только плотные рамки некоторых категорий объектов. Кроме того, часто оказывается выгодно использовать разнообразные типы слабых аннотаций, поскольку они лучше характеризуют различные семантические категории. Например, категории-объекты (такие как ‘знак’, ‘корова’, ‘автомобиль’) хорошо описываются рамками, а категории-фон (‘небо’, ‘трава’, ‘вода’), которые обычно занимают значительную часть изображения, — метками изображения.

В литературе описаны методы, которые используют слабые аннотации для обучения семантической сегментации, но большинство из них используют только метки изображения в качестве слабых аннотаций. Например, Вежневец и др. [60, 61] используют вероятностную графическую модель над набором изображений, чтобы распространять информацию о раз-

[1] http://pascallin.ecs.soton.ac.uk/challenges/VOC/

(a) Изображение

(b) Полная разметка

(c) Аннотация с помощью рамки

(d) Аннотация с помощью зёрен

небо самолёт дерево трава

(e) Аннотация метками изображения

Рисунок 2.1: Различные типы аннотаций для изображения из набора данных MSRC

метке между изображениями. В этой главе мы представляем метод для обучения семантической сегментации по смеси сильно- и слабоаннотированных изображений. Метод позволяет учитывать разные типы слабой аннотации, даже в рамках одного изображения.

В задаче категоризации документов разметка текстового документа представляет собой подмножество тегов (категорий) некоторого допустимого множества. Например, юридический документ может быть помечен 4 категориями из возможных 201: [‘сельское хозяйство’, ‘торговля’, ‘международные отношения’, ‘Украина’]. При получении такой разметки легко пропустить некоторые категории. Таким образом, слабой аннотацией документа может являться некоторое подмножество этих четырёх категорий. Предлагаемый метод обучает модель, предсказывающую полное множество категорий, имея лишь слабоаннотированную обучающую выборку.

Работа базируется на недавних исследованиях по использованию структурного метода опорных векторов с латентными переменными (англ. *latent-variable structural support vector machine, LV-SSVM*) для задач обучения со слабым наблюдением [62–64]. В отличие от них, предлагаемый метод использует специализированные функции потерь, которые измеряют рассогласованность разметки, предсказанной алгоритмом, с верной (возможно, слабой) аннотацией данного изображения. Мы определяем эти функции потерь так, чтобы они оценивали матожидание расстояния Хэмминга от разметки, предсказанной алгоритмом, до разметок, удовлетворяющих слабой аннотации изображения. Благодаря такому определению, функции, специализированные для разных типов аннотаций, определены в одном масштабе. Таким образом, модель содержит только один гиперпараметр, который регулирует относительный вклад

полностью размеченных и слабо аннотированных данных. Он необходим, поскольку последние обычно менее информативны. В разделе 3.2 эмпирически показано, как балансирование этого параметра может улучшить качество сегментации.

Для того чтобы обучить LV-SSVM с использованием различных типов аннотаций, необходимо определить специализированные функции потерь. Для введённых функций потерь необходимо описать алгоритмы *вывода, дополненного функцией потерь* и *вывода, согласованного с аннотацией*. Первый алгоритм выводит разметку изображения, высоко ранжируемую текущей моделью, но при этом сильно отличающуюся от верной аннотации, а второй выводит разметку, высоко ранжируемую текущей моделью, при этом согласующуюся с верной аннотацией (для слабых аннотаций существует множество разметок, согласующихся с ними). В разделе 2.2 показано, как решать эти оптимизационные задачи для различных функций потерь, используя эффективные комбинаторные алгоритмы, основанные на разрезах в графах.

Новизна работы заключается в следующем:

- мы предлагаем метод структурного обучения со слабым наблюдением, основанный на LV-SSVM, который минимизирует различные функции потерь, специализированные для различных видов аннотаций;
- в задаче обучения семантической сегментации мы определяем функции потерь для трёх популярных типов аннотаций (помимо полной разметки изображения) и их комбинаций: меток изображения, плотных рамок и зёрен объектов;
- в задаче обучения категоризации документов мы определяем функции потерь для полной разметки документа и для частичной разметки, в которой могут быть пропущены некоторые теги;
- мы предлагаем эффективные алгоритмы вывода, необходимые для обучения LV-SSVM с введёнными функциями потерь.

2.1 Обучение со слабыми аннотациями

Пусть необходимо настроить параметры $\mathbf{w}$ логлинейной модели (1.48), в которой за решение принимается максимум апостериорного распределения:

$$\mathbf{y}_{\text{MAP}} = \max_{\mathbf{y} \in \mathcal{Y}} \mathbf{w}^{\intercal} \boldsymbol{\psi}(\mathbf{y}; \mathbf{x}). \tag{2.1}$$

При наличии обучающей выборки $\{(\mathbf{x}^j, \mathbf{y}^j)\}_{j=1}^J$ это можно сделать с помощью структурного SVM (поиск оптимума в задаче 1.5), как показано в разделе 1.3.2.

Определение 2.1. *Слабой аннотацией* экземпляра обучающей выборки будем называть любой такой объект $\mathbf{z}$, для которого однозначно определяется непустое множество разметок $L(\mathbf{z}) \subseteq \mathcal{Y}$, *совместных со слабой аннотацией*.

2.1.1 Обобщённый SSVM

Рассмотрим случай, когда помимо J полностью размеченных объектов, обучающая выборка содержит I слабо аннотированных: $\{(\mathbf{x}^i, \mathbf{z}^i)\}_{i=J+1}^{J+I}$. Обобщим стандартную формулировку SSVM на случай присутствия в обучающей выборке полностью размеченных и слабо аннотированных данных.

Оптимизационная задача 2.1 (Обобщённый SSVM)**.**

$$\min_{\mathbf{w},\boldsymbol{\xi},\boldsymbol{\eta}} \ \frac{1}{2}\mathbf{w}^\intercal\mathbf{w} + \frac{C}{J+I}\left(\sum_{j=1}^{J}\xi_j + \alpha\sum_{i=J+1}^{J+I}\eta_i\right), \tag{2.2}$$

$$\text{при условиях} \quad \mathbf{w}^\intercal\boldsymbol{\psi}(\mathbf{y}^j;\mathbf{x}^j) \geq \max_{\bar{\mathbf{y}}\in\mathcal{Y}}\left\{\mathbf{w}^\intercal\boldsymbol{\psi}(\bar{\mathbf{y}};\mathbf{x}^j) + \Delta(\bar{\mathbf{y}};\mathbf{y}^j)\right\} - \xi_j, \quad \forall j \in \{1,\ldots,J\}, \tag{2.3}$$

$$\max_{\mathbf{y}\in L(\mathbf{z}^i)} \mathbf{w}^\intercal\boldsymbol{\psi}(\mathbf{y};\mathbf{x}^i) \ \geq \max_{\bar{\mathbf{y}}\in\mathcal{Y}}\left\{\mathbf{w}^\intercal\boldsymbol{\psi}(\bar{\mathbf{y}};\mathbf{x}^i) + \mathrm{K}(\bar{\mathbf{y}};\mathbf{z}^i)\right\} - \eta_i, \quad \forall i \in \{J+1,\ldots,J+I\}. \tag{2.4}$$

Здесь $\mathrm{K}(\bar{\mathbf{y}},\mathbf{z})$ — *слабая функция потерь*, задающая степень несогласованности некоторого ответа $\bar{\mathbf{y}} \in \mathcal{Y}$ со слабой аннотацией $\mathbf{z}$, η_i — минимизируемые нарушения ограничений.

Заметим, что при $I = 0$ эта оптимизационная задача сводится к стандартной постановке SSVM, а при $J = 0$ это частный случай SSVM с латентными переменными (*LV-SSVM*) [65]. Заметим также, что полная разметка $\mathbf{y}^j$ является вырожденным случаем слабой аннотации, где $L(\mathbf{z}^j) = \{\mathbf{y}^j\}$. Таким образом, оптимизационная задача 2.1 эквивалентна LV-SSVM, с тем исключением что она содержит балансирующий коэффициент α.

Если в задаче 2.1 перенести ограничения в целевую функцию, избавившись от фиктивных переменных, эквивалентной задачей безусловной оптимизации будет минимизация следующей целевой функции:

$$\mathsf{L}_{\text{GMM}}(\mathbf{w}) = \frac{1}{2}\mathbf{w}^\intercal\mathbf{w} + \frac{C}{J+I}\left(\sum_{j=1}^{J}\left[\max_{\bar{\mathbf{y}}\in\mathcal{Y}}\left\{\mathbf{w}^\intercal\boldsymbol{\psi}(\bar{\mathbf{y}};\mathbf{x}^j) + \Delta(\bar{\mathbf{y}};\mathbf{y}^j)\right\} - \mathbf{w}^\intercal\boldsymbol{\psi}(\mathbf{y}^j;\mathbf{x}^j)\right] + \right. \tag{2.5}$$

$$\left. \alpha\sum_{i=J+1}^{J+I}\left[\max_{\bar{\mathbf{y}}\in\mathcal{Y}}\left\{\mathbf{w}^\intercal\boldsymbol{\psi}(\bar{\mathbf{y}};\mathbf{x}^i) + \mathrm{K}(\bar{\mathbf{y}};\mathbf{z}^i)\right\} - \max_{\mathbf{y}\in L(\mathbf{z}^i)}\mathbf{w}^\intercal\boldsymbol{\psi}(\mathbf{y};\mathbf{x}^i)\right]\right) =$$

$$\frac{1}{2}\mathbf{w}^\intercal\mathbf{w} + \frac{C}{J+I}\left(\sum_{j=1}^{J}\left[\max_{\bar{\mathbf{y}}\in\mathcal{Y}}\left\{\mathbf{w}^\intercal\boldsymbol{\psi}(\bar{\mathbf{y}};\mathbf{x}^j) + \Delta(\bar{\mathbf{y}};\mathbf{y}^j)\right\} - \mathbf{w}^\intercal\boldsymbol{\psi}(\mathbf{y}^j;\mathbf{x}^j)\right] + \right. \tag{2.6}$$

$$\left. \alpha\sum_{i=J+1}^{J+I}\max_{\bar{\mathbf{y}}\in\mathcal{Y}}\left\{\mathbf{w}^\intercal\boldsymbol{\psi}(\bar{\mathbf{y}};\mathbf{x}^i) + \mathrm{K}(\bar{\mathbf{y}};\mathbf{z}^i)\right\}\right) - \frac{C\alpha}{J+I}\sum_{i=J+1}^{J+I}\max_{\mathbf{y}\in L(\mathbf{z}^i)}\mathbf{w}^\intercal\boldsymbol{\psi}(\mathbf{y};\mathbf{x}^i).$$

Первые два слагаемых в (2.6) выпуклы, а последнее, с учётом знака «минус», вогнуто по $\mathbf{w}$. Эти факты следуют из того, что максимум конечного числа линейных функций является выпуклым, так же как и сумма произвольных выпуклых функций. Следуя Йу и Йоахимсу [65], мы используем эту специфическую структуру задачи — сумму выпуклой и вогнутой функции. Это позволяет применить выпукло-вогнутую процедуру (англ. *convex-concave procedure, CCCP*) [66] для приближённой минимизации. Идея этого алгоритма заключается в том, что-

бы итеративно минимизировать сумму выпуклой функции и линеаризации вогнутой в точке минимума с предыдущей итерации. Таким образом, на n-й итерации значение параметров пересчитывается по формуле

$$\mathbf{w}_n = \operatorname*{argmin}_{\mathbf{w}} \left\{ \frac{1}{2}\mathbf{w}^\intercal\mathbf{w} + \frac{C}{J+I}\left(\sum_{j=1}^{J} \left[\max_{\bar{\mathbf{y}}\in\mathcal{Y}} \left\{ \mathbf{w}^\intercal\boldsymbol{\psi}(\bar{\mathbf{y}};\mathbf{x}^j) + \Delta(\bar{\mathbf{y}};\mathbf{y}^j) \right\} - \mathbf{w}^\intercal\boldsymbol{\psi}(\mathbf{y}^j;\mathbf{x}^j) \right] + \right. \right. \tag{2.7}$$
$$\left. \left. \alpha \sum_{i=1}^{I} \max_{\bar{\mathbf{y}}\in\mathcal{Y}} \left\{ \mathbf{w}^\intercal\boldsymbol{\psi}(\bar{\mathbf{y}};\mathbf{x}^i) + \mathrm{K}(\bar{\mathbf{y}};\mathbf{z}^i) \right\} \right) - \frac{C\alpha}{J+I} \sum_{i=J+1}^{J+I} \mathbf{w}^\intercal\boldsymbol{\psi}(\mathbf{y}_n^i;\mathbf{x}^i) \right\},$$

где $\mathbf{y}_n^i = \operatorname{argmax}_{\mathbf{y}\in L(\mathbf{z}^i)} \mathbf{w}_{n-1}^\intercal\boldsymbol{\psi}(\mathbf{y};\mathbf{x}^i)$. В (2.7) необходимо минимизировать выпуклую функцию, которая фактически совпадает с целевой функцией структурного SVM, для чего могут применяться методы, описанные в разделе 1.3.2. Заметим, что эта функция зависит от $\mathbf{w}_{n-1}$ не напрямую, а через выведенную мнимую разметку $\mathbf{y}_n^i,\ \forall i$. Таким образом, алгоритм поочерёдно пересчитывает значения $\mathbf{y}_n^i$ и $\mathbf{w}_n$. Гарантируется, что метод сходится к локальному минимуму или седловой точке.

Определение 2.2. Задача поиска $\operatorname{argmax}_{\mathbf{y}\in L(\mathbf{z})} \mathbf{w}^\intercal\boldsymbol{\psi}(\mathbf{y};\mathbf{x})$ на множестве, ограниченном аннотацией $\mathbf{z}$, возникающая в выпукло-вогнутой процедуре, называется *выводом, согласованным с аннотацией* (англ. *annotation-consistent inference*).

Таким образом, при оптимизации в обобщённом SSVM необходимо помимо вывода, дополненного функцией потерь Δ в (2.3), необходимо также эффективно выполнять вывод, дополненный слабой функцией потерь K в (2.4), а также вывод, согласованный с аннотацией. Последние две задачи зависят от используемого типа аннотаций. В разделе 2.2 описаны конкретные алгоритмы для трёх типов аннотаций.

2.1.2 Обобщённый SSVM и максимизация неполного правдоподобия

В разделе 1.3.1 описан алгоритм настройки параметров марковской сети с помощью максимизации правдоподобия. В случае присутствия в модели латентных переменных говорят о максимизации неполного правдоподобия, для чего обычно используют EM-алгоритм. Введём вероятностную модель для задачи разметки и покажем, как EM-алгоритм, применённый к этой модели, связан с выпукло-вогнутой процедурой оптимизации в обобщённом SSVM.

Поскольку полная разметка является частным случаем аннотации, будем без потери общности рассматривать только объекты со слабыми аннотациями $\{(\mathbf{x}^j,\mathbf{z}^j)\}_{j=1}^J$. Определим неполное правдоподобие как

$$\mathsf{L}(\mathbf{w}) = \prod_{j=1}^{J} \mathsf{P}(\mathbf{z}^j \mid \mathbf{x}^j,\mathbf{w}) = \prod_{j=1}^{J} \sum_{\bar{\mathbf{y}}\in\mathcal{Y}} \mathsf{P}(\bar{\mathbf{y}},\mathbf{z}^j \mid \mathbf{x}^j,\mathbf{w}). \tag{2.8}$$

Распишем полное правдоподобие модели (т. е. правдоподобие при известных латентных переменных) по определению условной вероятности:

$$\mathsf{P}(\mathbf{y}, \mathbf{z} \mid \mathbf{x}, \mathbf{w}) = \mathsf{P}(\mathbf{y} \mid \mathbf{x}, \mathbf{w})\mathsf{P}(\mathbf{z} \mid \mathbf{y}, \mathbf{x}, \mathbf{w}). \tag{2.9}$$

Будем рассматривать традиционную параметризацию апостериорного распределения: $\mathsf{P}(\mathbf{y} \mid \mathbf{x}, \mathbf{w}) \propto \exp\big(\mathbf{w}^\intercal \boldsymbol{\psi}(\mathbf{y}; \mathbf{x})\big)$. Также для простоты будем рассматривать случай, когда аннотация $\mathbf{z}$ однозначно определяется по разметке $\mathbf{y}$, что верно, например, для аннотации изображения множеством присутствующих на нём меток (см. определение 2.4). Тогда распределение на аннотации будет иметь вид $\mathsf{P}(\mathbf{z} \mid \mathbf{y}, \mathbf{x}, \mathbf{w}) = \mathsf{P}(\mathbf{z} \mid \mathbf{y}) = [\![\mathbf{y} \in L(\mathbf{z})]\!]$. Рассмотрим также нормальное априорное распределение на параметры, не зависящее от признаков: $\mathsf{P}(\mathbf{w} \mid \mathbf{x}) = \mathsf{P}(\mathbf{w}) = \mathcal{N}(\mathbf{w}; \mathbf{0}, C\mathbf{I})$, где $\mathbf{I}$ — единичная матрица. Тогда апостериорное распределение на параметры пропорционально произведению их правдоподобия на априорное распределение:

$$\mathsf{P}\left(\mathbf{w} \mid \{(\mathbf{x}^j, \mathbf{z}^j)\}_{j=1}^J\right) \propto \mathsf{P}(\mathbf{w})\mathsf{L}(\mathbf{w}) \propto \exp\left(-\frac{1}{2C}\mathbf{w}^\intercal\mathbf{w}\right)\mathsf{L}(\mathbf{w}). \tag{2.10}$$

Для обоснования связи двух методов нам понадобится точечная оценка нормировочной константы. Докажем вспомогательное утверждение.

Лемма 2.1. *Пусть* $(\mathbf{x}, \mathbf{z})$ *— слабоаннотированный объект обучающей выборки. Если слабая функция потерь неотрицательна:* $\mathrm{K}(\tilde{\mathbf{y}}; \mathbf{z}) \geq 0,\ \forall \tilde{\mathbf{y}}$*, то для любого вектора* $\mathbf{w}$ *верна следующая оценка:*

$$-\log \sum_{\tilde{\mathbf{y}} \in \mathcal{Y}} \exp\big(\mathbf{w}^\intercal \boldsymbol{\psi}(\tilde{\mathbf{y}}; \mathbf{x})\big) \geq -\max_{\tilde{\mathbf{y}} \in \mathcal{Y}} \big\{\mathbf{w}^\intercal \boldsymbol{\psi}(\tilde{\mathbf{y}}; \mathbf{x}) + \mathrm{K}(\tilde{\mathbf{y}}; \mathbf{z})\big\} + \mathrm{const}, \tag{2.11}$$

где константа не зависит от $\mathbf{w}$.

Доказательство. Запишем цепочку неравенств:

$$-\log \sum_{\tilde{\mathbf{y}} \in \mathcal{Y}} \exp\big(\mathbf{w}^\intercal \boldsymbol{\psi}(\tilde{\mathbf{y}}; \mathbf{x})\big) \geq -\max_{\tilde{\mathbf{y}} \in \mathcal{Y}} \big\{\mathbf{w}^\intercal \boldsymbol{\psi}(\tilde{\mathbf{y}}; \mathbf{x})\big\} - \log|\mathcal{Y}| \geq \tag{2.12}$$

$$-\max_{\tilde{\mathbf{y}} \in \mathcal{Y}} \big\{\mathbf{w}^\intercal \boldsymbol{\psi}(\tilde{\mathbf{y}}; \mathbf{x}) + \mathrm{K}(\tilde{\mathbf{y}}; \mathbf{z})\big\} - \log|\mathcal{Y}|. \tag{2.13}$$

Здесь первое неравенство следует из (1.58), где $f(\tilde{\mathbf{y}}) \equiv \mathbf{w}^\intercal \boldsymbol{\psi}(\tilde{\mathbf{y}}; \mathbf{x})$, а второе — из неотрицательности функции K. Константа $-\log|\mathcal{Y}|$ не зависит от $\mathbf{w}$, что завершает доказательство леммы. □

Теорема 2.1. *Пусть слабая функция потерь неотрицательна:* $\mathrm{K}(\mathbf{y}; \mathbf{z}) \geq 0$*. Тогда, при условии равенства начальных приближений* $\mathbf{w}^0$*, выпукло-вогнутая процедура минимизации целевой функции обобщённого SSVM* (2.5) *сходится к тому же вектору* $\mathbf{w}^*$*, что и EM-алгоритм для максимизации распределения* (2.10) *со следующими модификациями:*

- *на E-шаге оценка матожидания производится не по действительному распределению на латентные переменные, а по его точечной MAP-оценке;*
- *на M-шаге максимизируется не полученная на E-шаге оценка матожидания, а её нижняя оценка, где логарифмы нормировочных констант распределений на латентные разметки слабоаннотированных объектов оцениваются согласно* (2.11).

Доказательство. Построим последовательность значений $[\mathbf{w}_n]_{n=1}^{\infty}$, получаемую на итерациях описанной модификации EM-алгоритма и покажем, что она совпадает с аналогичной последовательностью обобщённого SSVM. Пусть $\mathbf{w}_n$ — значение вектора параметров, полученное на n-й итерации. Получим значение для E-шага следующей итерации:

$$Q(\mathbf{w};\mathbf{w}_n) = \sum_{j=1}^{J} \mathsf{E}_{\mathbf{y}|\mathbf{z}^j,\mathbf{x}^j,\mathbf{w}_n} \log\big(\mathsf{P}(\mathbf{y}\mid\mathbf{x}^j,\mathbf{w})\mathsf{P}(\mathbf{z}^j\mid\mathbf{y})\mathsf{P}(\mathbf{w})\big) = \tag{2.14}$$
$$\sum_{j=1}^{J}\sum_{\mathbf{y}\in\mathcal{Y}} \mathsf{P}(\mathbf{y}\mid\mathbf{z}^j,\mathbf{x}^j,\mathbf{w}_n)\Big[\mathbf{w}^{\mathsf{T}}\boldsymbol{\psi}(\mathbf{y};\mathbf{x}^j) + \log[\![\mathbf{y}\in L(\mathbf{z}^j)]\!] - \log\sum_{\tilde{\mathbf{y}}\in\mathcal{Y}}\exp\big(\mathbf{w}^{\mathsf{T}}\boldsymbol{\psi}(\tilde{\mathbf{y}};\mathbf{x}^j)\big)\Big] -$$
$$\frac{J}{2C}\mathbf{w}^{\mathsf{T}}\mathbf{w} + \text{const}.$$

Константа получается из логарифма нормировочной константы априорного распределения на параметры. Заменим теперь $\mathsf{P}(\mathbf{y}\mid\mathbf{z}^j,\mathbf{x}^j,\mathbf{w}_n)$ на $\delta[\operatorname{argmax}_{\mathbf{y}}\mathsf{P}(\mathbf{y}\mid\mathbf{z}^j,\mathbf{x}^j,\mathbf{w}_n)]$ и получим точечную оценку матожидания:

$$\dot{Q}(\mathbf{w};\mathbf{w}_n) = \sum_{j=1}^{J}\Big[\mathbf{w}^{\mathsf{T}}\boldsymbol{\psi}(\bar{\mathbf{y}}^j;\mathbf{x}^j) - \log[\![\bar{\mathbf{y}}^j\in L(\mathbf{z}^j)]\!] - \log\sum_{\tilde{\mathbf{y}}\in\mathcal{Y}}\exp\big(\mathbf{w}^{\mathsf{T}}\boldsymbol{\psi}(\tilde{\mathbf{y}};\mathbf{x}^j)\big)\Big] - \frac{J}{2C}\mathbf{w}^{\mathsf{T}}\mathbf{w} + \text{const}, \tag{2.15}$$

где $\bar{\mathbf{y}}^j = \operatorname{argmax}_{\mathbf{y}\in\mathcal{Y}}\mathsf{P}(\mathbf{y}\mid\mathbf{z}^j,\mathbf{x}^j,\mathbf{w}_n) = \operatorname{argmax}_{\mathbf{y}\in L(\mathbf{z}^j)}\mathsf{P}(\mathbf{y}\mid\mathbf{x}^j,\mathbf{w}_n),\ \forall j$. Из $\bar{\mathbf{y}}^j\in L(\mathbf{z}^j)$ следует, что $\log[\![\bar{\mathbf{y}}^j\in L(\mathbf{z}^j)]\!] = 0$. Зависимость от предыдущего значения $\mathbf{w}_n$ в этой функции выражается лишь через значения $\bar{\mathbf{y}}^j$.

На M-шаге итерации $n+1$ необходимо максимизировать эту функцию по $\mathbf{w}$, однако она содержит в себе экспоненциальное число слагаемых, с которым трудно работать. Получим нижнюю оценку, используя результат леммы 2.1:

$$\dot{Q}(\mathbf{w};\mathbf{w}_n) \geq \sum_{j=1}^{J}\Big[\mathbf{w}^{\mathsf{T}}\boldsymbol{\psi}(\bar{\mathbf{y}}^j;\mathbf{x}^j) - \max_{\tilde{\mathbf{y}}\in\mathcal{Y}}\big\{\mathbf{w}^{\mathsf{T}}\boldsymbol{\psi}(\tilde{\mathbf{y}};\mathbf{x}^j) + \mathrm{K}(\tilde{\mathbf{y}};\mathbf{z}^j)\big\}\Big] - \frac{J}{2C}\mathbf{w}^{\mathsf{T}}\mathbf{w} + \text{const}. \tag{2.16}$$

Функция (2.16) с точностью до аффинного преобразования с отрицательным коэффициентом совпадает с целевой функцией SSVM (1.60). На каждой итерации EM-алгоритма обновляется значение параметров $\mathbf{w}_n$ с помощью максимизации (2.16), причём $\bar{\mathbf{y}}^j$ находятся с помощью процедуры, эквивалентной выводу, согласованному с аннотацией. Этот шаг таким образом совпадает с формулой поиска точки минимума в выпукло-вогнутой процедуре (2.7). Из равенства начальных приближений следует, что последовательности $[\mathbf{w}_n]_{n=1}^{\infty}$ в обоих методах совпадают, а значит сходятся к одной и той же точке $\mathbf{w}^*$, что и требовалось доказать. □

Доказательство теоремы 2.1 позволяет лучше понять свойства описанного метода. Процедура оптимизации аналогична той, что происходит в EM-алгоритме с жёстким присваиванием. В этом алгоритме на каждом шаге максимизируется нижняя оценка логарифма апостериорного распределения (2.10). Покажем, что аналогичное свойство выполняется и в рассмотренной модификации. Рассмотрим некоторое распределение на латентные переменные $q(\tilde{\mathbf{y}})$ для j-го объекта обучающей выборки. Благодаря неравенству Йенсена, при любом выборе $q(\tilde{\mathbf{y}})$ справедлива следующая верхняя оценка [36, (11.85)]:

$$\log \mathsf{P}(\mathbf{z}^j \mid \mathbf{x}^j, \mathbf{w}) = \log \sum_{\tilde{\mathbf{y}}} q(\tilde{\mathbf{y}}) \frac{\mathsf{P}(\mathbf{z}^j, \tilde{\mathbf{y}} \mid \mathbf{x}^j, \mathbf{w})}{q(\tilde{\mathbf{y}})} \geq \sum_{\tilde{\mathbf{y}}} q(\tilde{\mathbf{y}}) \log \frac{\mathsf{P}(\mathbf{z}^j, \tilde{\mathbf{y}} \mid \mathbf{x}^j, \mathbf{w})}{q(\tilde{\mathbf{y}})}. \tag{2.17}$$

В классе дельта-функций наиболее плотную верхнюю оценку обеспечивает мода распределения на латентные переменные при текущих параметрах $\mathbf{w}$. Если подставить это значение в целевую функцию (2.10), то она будет равна функции $\dot{Q}(\mathbf{w}; \mathbf{w}_n)$ (2.15) с точностью до константы — энтропии распределения $q(\tilde{\mathbf{y}})$. Функция (2.16) является её нижней оценкой с точностью до константы, значит, она является и нижней оценкой целевой функции.

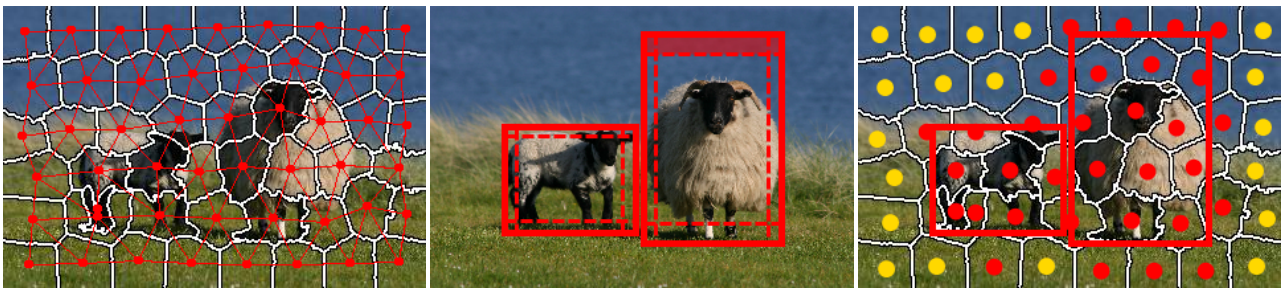
Переход к точечной оценке позволяет избежать суммирования по всевозможным разметкам, что делает метод применимым у более широкому кругу задач, в том числе для настройки параметров Марковских сетей. Классический EM-алгоритм игнорирует эмпирическую функцию потерь K, которую можно определить, исходя из знания предметной области, таким образом, метод одинаково трактует все неправильные разметки. Как и в случае стандартного SSVM, обобщённый позволяет настраивать параметры с целью максимизации отступа, делая вероятность разметки тем выше, чем ближе она к аннотации (см. рис. 1.4). Это позволяет повысить робастность настройки параметров.

2.2 Типы аннотаций для обучения сегментации изображений

В этом разделе формально определяются три типа слабой аннотации в задаче семантической сегментации изображений, затем описывается стандартная схема обучения структурного SVM, если доступна разметка всех изображений обучающей выборки. Вводятся сопутствующие функции потерь обобщённого SSVM и необходимые алгоритмы вывода.

Пусть на изображении задано разбиение пикселей на *суперпиксели* $\mathcal{V}$ — группы соседних пикселей, сходных по цвету и текстуре (рис. 2.2a).

Определение 2.3. Рассмотрим дискретное изображение высоты H и ширины W. *Разбиением на суперпиксели* назовём функционал $v : \{1, \ldots, W\} \times \{1, \ldots, H\} \to \mathcal{V}$, относящий каждый пиксель к одному из суперпикселей. Прообразы элементов $\mathcal{V}$ образуют связные сегменты в пространстве изображения.



(a) Разбиение на суперпиксели (b) Плотность рамок (c) Множества $\mathcal{V}_k$ и $\mathcal{V}_0$

Рисунок 2.2: Примеры пересегментации изображения и аннотации рамками. (a) Разбиение изображения на суперпиксели и структура парно-сепарабельной марковской сети. (b) Пример плотной и неплотной рамок для $r = 0.1$. Рамка слева является r-плотной для класса ‘овца’, так как образ объекта «касается» каждой из 4 сторон рамки. Рамка справа не является r-плотной, так как в регионе $[\mathit{left}(\bar{z}), \mathit{right}(\bar{z})] \times [\mathit{top}(\bar{z}), \mathit{top}(\bar{z}) + r(\mathit{bottom}(\bar{z}) - \mathit{top}(\bar{z}))]$ нет пикселей категории ‘овца’. (c) Разбиение множества суперпикселей на подмножества. Красным показано множество $\mathcal{V}_k$, где k соответствует категории ‘овца’, жёлтым — $\mathcal{V}_0$.

Определение 2.4. *Аннотацией метками изображения* называется множество $\mathbf{z} \subset \mathcal{K}$ категорий, присутствующих на изображении. Пусть $\mathbf{y}$ — разметка изображения, тогда уникальные метки изображения $\mathbf{z} = \{y_v \mid v \in \mathcal{V}\}$ (рис. 2.1e).

Дальнейшие типы аннотации оперируют понятием *объектов* реального мира, таких как конкретный автомобиль или человек. На изображениях им соответствуют *образы объектов* — множества пикселей, получившихся проектированием этого объекта в пространство изображения. Не все категории в задаче семантической сегментации соответствуют объектам — такие категории как ‘трава’, ‘небо’ являются фоновыми, поэтому для них не подходят соответствующие типы аннотации. С формальной точки зрения, будем считать образом объекта связную область пикселей изображения одной категории.

Определение 2.5. *Рамкой*, аннотирующей объект категории k, называется структура $\bar{z}$, задающая прямоугольник на изображении, включающий в себя образ этого объекта. Для $\bar{z}$ определены функции $\mathit{label}(\bar{z})$, а также $\mathit{left}(\bar{z}), \mathit{right}(\bar{z}), \mathit{top}(\bar{z}), \mathit{bottom}(\bar{z})$, определяющие границы прямоугольника. Пусть $\mathbf{y}$ — разметка изображения, а $\mathcal{P}'_k$ — некоторое подмножество пикселей, получивших метку k: $\mathcal{P}'_k \subset \{\mathbf{p} \mid y_{v(\mathbf{p})} = k\}$. Рамка $\bar{z}$ описывает множество $\mathcal{P}'_k$, если $\mathcal{P}'_k \subset [\mathit{left}(\bar{z}), \mathit{right}(\bar{z})] \times [\mathit{top}(\bar{z}), \mathit{bottom}(\bar{z})]$, а также $\mathit{label}(\bar{z}) = k$ (см. рис. 2.1c).

Определение 2.6. Пусть задано число $r \in [0, 0.5)$. Будем называть рамку $\bar{z}$ *r-плотной* по отношению к множеству пикселей $\mathcal{P}'_k$, если выполняются следующие предположения о пересечении множеств:

$$\mathcal{P}'_k \cap \big([\mathit{left}(\bar{z}), \mathit{left}(\bar{z}) + r(\mathit{right}(\bar{z}) - \mathit{left}(\bar{z}))] \times [\mathit{top}(\bar{z}), \mathit{bottom}(\bar{z})] \big) \neq \varnothing, \tag{2.18}$$

$$\mathcal{P}'_k \cap \big([\mathit{right}(\bar{z}) - r(\mathit{right}(\bar{z}) - \mathit{left}(\bar{z})), \mathit{right}(\bar{z})] \times [\mathit{top}(\bar{z}), \mathit{bottom}(\bar{z})] \big) \neq \varnothing, \tag{2.19}$$

$$\mathcal{P}'_k \cap \big([\mathit{left}(\bar{z}), \mathit{right}(\bar{z})] \times [\mathit{top}(\bar{z}), \mathit{top}(\bar{z}) + r(\mathit{bottom}(\bar{z}) - \mathit{top}(\bar{z}))] \big) \neq \varnothing, \tag{2.20}$$

$$\mathcal{P}'_k \cap \big([\mathit{left}(\bar{z}), \mathit{right}(\bar{z})] \times [\mathit{bottom}(\bar{z}) - r(\mathit{bottom}(\bar{z}) - \mathit{top}(\bar{z})), \mathit{bottom}(\bar{z})] \big) \neq \varnothing. \tag{2.21}$$

Будем обозначать это отношение следующим образом: $\bar{z} \sqsupseteq_r \mathcal{P}'_k$.

Согласно этому определению, расстояние от множества $\mathcal{P}'_k$ до каждой из сторон рамки не превосходит некоторого порога, зависящего от измерений рамки (рис. 2.2b). Согласно исследованиям типичных аннотаций, производимых пользователями, большинство рамок оказываются r-плотными с $r = 0.06$ [67]. Поэтому в дальнейшем под плотной рамкой мы будем понимать 0.06-плотную рамку.

Определение 2.7. *Аннотацией плотными рамками* категорий $\mathcal{K}' \subset \mathcal{K}$ на некотором изображении называют множество рамок, плотных по отношению к образам каждого из объектов категорий из $\mathcal{K}'$. Пусть $\mathbf{y}$ — разметка изображения, и для каждой категории $k \in \mathcal{K}'$ задано покрытие $\{\mathcal{P}_k^i\}_i$ множества пикселей, отнесённых к этой категории: $\bigcup_i \mathcal{P}_k^i = \{\mathbf{p} \mid y_{v(\mathbf{p})} = k\}$, причём все $\mathcal{P}_k^i$ представляют собой связные множества. Тогда аннотация плотными рамками — это множество $\mathbf{z}^{\text{bb}} = \{\bar{z}_k^i\}_{i,k}$, таких что $\bar{z}_k^i \sqsupseteq_r \mathcal{P}_k^i,\ \mathit{label}(\bar{z}_k^i) = k,\ \forall i, \forall k \in \mathcal{K}'$.

Заметим, что аннотация плотными рамками определяется по полной разметке неоднозначно из-за неединственности покрытия $\{\mathcal{P}_k^i\}_i$ и определения r-плотной рамки при $r > 0$.

Определение 2.8. *Зерном*, аннотирующим объект категории $\dot{k}$, называется пара $\dot{z} = (\dot{\mathbf{p}}, \dot{k})$, задающая пиксель изображения, принадлежащий образу этого объекта. Пусть $\mathbf{y}$ — разметка изображения, а $\mathcal{P}'_k$ — некоторое подмножество пикселей, получивших метку k: $\mathcal{P}'_k \subset \{\mathbf{p} \mid y_{v(\mathbf{p})} = k\}$. Зерно $\dot{z} = (\dot{\mathbf{p}}, \dot{k})$ описывает множество $\mathcal{P}'_k$, если $\dot{\mathbf{p}} \in \mathcal{P}'_k$, а также $\dot{k} = k$ (рис. 2.1d).

Определение 2.9. *Аннотацией зёрнами* категорий $\mathcal{K}' \subset \mathcal{K}$ на некотором изображении называют множество зёрен, принадлежащих образам каждого из объектов категорий из $\mathcal{K}'$. Пусть $\mathbf{y}$ — разметка изображения, и для каждой категории $k \in \mathcal{K}'$ задано покрытие $\{\mathcal{P}_k^i\}_i$ множества пикселей, отнесённых к этой категории: $\bigcup_i \mathcal{P}_k^i = \{\mathbf{p} \mid y_{v(\mathbf{p})} = k\}$, причём все $\mathcal{P}_k^i$ представляют собой связные множества. Тогда аннотация зёрнами — это множество $\mathbf{z}^{\text{os}} = \{\dot{z}_k^i = (\dot{\mathbf{p}}_{k,i}, k)\}_{i,k}$, таких что $\dot{\mathbf{p}}_{k,i} \in \mathcal{P}_k^i,\ \forall i, \forall k \in \mathcal{K}'$.

Аннотация зёрнами также определяется по полной разметке неоднозначно, однако предполагается, что зерно находится в центре образа объекта.

Чтобы использовать конкретный вид слабой аннотации при обучении, необходимо определить функцию потерь K для данного типа аннотации, которая допускает эффективный вывод, дополненный функций потерь и вывод, согласованный с аннотацией. Первый должен быть очень эффективным, поскольку он вызывается на каждой итерации обучения, и, как правило, является основным источником вычислительной сложности. Также будет показано, что некоторые типы аннотаций могут комбинироваться в рамках одного изображения.

2.2.1 Обучение сегментации по полной разметке

В случае, когда вся обучающая выборка $\{(\mathbf{x}^j, \mathbf{y}^j)\}_{j=1}^J$ размечена полностью, параметры марковской сети можно искать, решая оптимизационную задачу 1.5 (стандартный структурный SVM, без ограничений (2.4)). Ниже описано, как задача семантической сегментации изображений формулируется в терминах структурного SVM.

Будем моделировать сегментацию изображения с помощью парно-сепарабельной марковской сети над графом $G = (\mathcal{V}, \mathcal{E})$, переменные которой $\mathbf{y} \in \mathbb{R}^{|\mathcal{V}|}$ соответствуют суперпикселям изображения (далее для упрощения нотации будем отождествлять переменные с соответствующими им суперпикселями). На этапе вывода переменным назначаются метки категорий. Это означает, что все пиксели, относящиеся к данному суперпикселю, получают его метку. Парные потенциалы объединяют все пары суперпикселей, имеющих общую границу (рис. 2.2a). Обозначим $\mathbf{x}_v^{\mathrm{v}} \in \mathbb{R}^{d_{\mathrm{v}}}$ вектор признаков суперпикселя $v \in \mathcal{V}$, $\mathbf{x}_{uv}^{\mathrm{e}} \in \mathbb{R}^{d_{\mathrm{e}}}$ — вектор признаков, описывающий сходство соседних суперпикселей u и v, а $\mathbf{x} = \bigoplus_{v \in \mathcal{V}} \mathbf{x}_v^{\mathrm{v}} \oplus \bigoplus_{(u,v) \in \mathcal{E}} \mathbf{x}_{uv}^{\mathrm{e}}$ — их конкатенацию. Каждая переменная y_v, соответствующая суперпикселю v, принимает значение одной из меток категорий из множества $\mathcal{K} = \{1, \dots, K\}$. Пространство $\mathcal{X}$ содержит всевозможные признаки изображения $\mathbf{x}$, а пространство $\mathcal{Y}$ — всевозможные разметки $\mathbf{y}$ (на практике изображения могут иметь разное число суперпикселей и разное число их соседних пар, однако в нотации этит факт игнорируется для простоты; обобщение на общий случай тривиально). Будем использовать логлинейную параметризацию (1.48) потенциалов марковской сети:

$$\log \mathsf{P}(\mathbf{y} \mid \mathbf{x}, \mathbf{w}) - \text{const} = \mathbf{w}^{\intercal} \Psi(\mathbf{x}, \mathbf{y}) = \sum_{v \in \mathcal{V}} \sum_{k=1}^{K} [\![y_v = k]\!] (\mathbf{x}_v^{\intercal} \mathbf{w}_k^{\mathrm{v}}) + \sum_{(v,u) \in \mathcal{E}} [\![y_v = y_u]\!] (\mathbf{x}_{vu}^{\intercal} \mathbf{w}^{\mathrm{e}}). \quad (2.22)$$

Здесь $\mathbf{w} = \bigoplus_{k=1}^{K} \mathbf{w}_k^{\mathrm{v}} \oplus \mathbf{w}^{\mathrm{e}}$ — вектор параметров модели, $\mathbf{w}_k^{\mathrm{v}} \in \mathbb{R}^{d_{\mathrm{v}}}$, $\mathbf{w}^{\mathrm{e}} \in \mathbb{R}^{d_{\mathrm{e}}}$. Мы полагаем парные веса $\mathbf{w}^{\mathrm{e}}$ и парные признаки $\mathbf{x}_{uv}^{\mathrm{e}}$ неотрицательными числами, и таким образом получаем *ассоциативную* фуннкцию энергии [68]. В этом случае задача вычисления MAP-оценки, хотя и является NP-трудной, может быть эффективно решена приближённо (раздел 1.2.4).

В задаче сегментации в качестве функции потерь часто используется расстояние Хэмминга (число неправильно распознанных пикселей):

$$\Delta(\bar{\mathbf{y}}, \mathbf{y}^j) = \sum_{v \in \mathcal{V}} c_v^j [\![\bar{y}_v \neq y_v^j]\!], \quad (2.23)$$

где c_v^j — площадь v-го суперпикселя j-го изображения. На практике в разметке суперпикселя может встретиться несколько меток (такие суперпиксели называют *гетерогенными*). В этом случае функция потерь также равна числу неверно распознанных пикселей. Чтобы не загромождать нотацию, мы рассматриваем только гомогенные суперпиксели. Вывод тривиально обобщается на гетерогенный случай.

Эта функция потерь декомпозируется по переменным. Это значит, что вывод, дополненный функцией потерь, вычислительно не сложнее, чем нахождение MAP-оценки, и также может быть выполнен с помощью α-расширения. Известны также некоторые частные случаи функций потерь высоких порядков (т.е. не разделяющуюся на функции от переменных или их пар), которые допускают эффективный приближённый вывод [21, 22, 52].

2.2.2 Учёт аннотации метками изображений

Определение 2.10. Назовём сильной функцией потерь *по метке изображения* следующую функцию:

$$\Delta_{\text{il}}(\bar{\mathbf{y}}, \mathbf{y}) = \sum_{v \in \mathcal{V}} c_v [\![\nexists u \in \mathcal{V} : y_u = \bar{y}_v \ \vee \ \nexists u \in \mathcal{V} : \bar{y}_u = y_v]\!]. \tag{2.24}$$

Эта функция штрафует суперпиксели, помеченные метками, которых нет в $\mathbf{y}$, а также суперпиксели, верные метки которых не присутствуют в $\bar{\mathbf{y}}$.

Определение 2.11. Пусть $\mathbf{z}$ — метка изображения. Назовём *слабой функцией потерь по метке изображения* следующую функцию, параметризованную числами s_k, для $k \in \mathbf{z}$:

$$\text{K}_{\text{il}}(\bar{\mathbf{y}}, \mathbf{z}) = \text{K}_{\text{il}}(\bar{\mathbf{y}}, \mathbf{z}; s_k) = \sum_{k \notin \mathbf{z}} \sum_{v \in \mathcal{V}} c_v [\![\bar{y}_v = k]\!] + \sum_{k \in \mathbf{z}} s_k \prod_{v \in \mathcal{V}} [\![\bar{y}_v \neq k]\!]. \tag{2.25}$$

Лемма 2.2. *Пусть $\mathbf{z}$ — множество меток категорий, присутствующих в $\bar{\mathbf{y}}$, а s_k — число пикселей в каждой из них. Тогда слабая функция потерь по меткам изображения является верхней оценкой сильной с мультипликативным коэффициентом не более 2:*

$$\frac{1}{2} \text{K}_{il}(\bar{\mathbf{y}}, \mathbf{z}) \leq \Delta_{il}(\bar{\mathbf{y}}, \mathbf{y}) \leq \text{K}_{il}(\bar{\mathbf{y}}, \mathbf{z}). \tag{2.26}$$

Доказательство. Преобразуем K_{il}, учитывая определение $\mathbf{z}$:

$$\begin{aligned} \text{K}_{\text{il}}(\bar{\mathbf{y}}, \mathbf{z}) = \sum_{v \in \mathcal{V}} c_v [\![\bar{y}_v \notin \mathbf{z}]\!] + \sum_{k \in \mathbf{z}} \sum_{v \in \mathcal{V}} c_v [\![y_v = k]\!] \prod_{u \in \mathcal{V}} [\![\bar{y}_u \neq k]\!] = \\ \sum_{v \in \mathcal{V}} c_v [\![\nexists u \in \mathcal{V} : y_u = \bar{y}_v]\!] + \sum_{v \in \mathcal{V}} c_v [\![\nexists u \in \mathcal{V} : \bar{y}_u = y_v]\!] = \\ \sum_{v \in \mathcal{V}} c_v \left([\![\nexists u \in \mathcal{V} : y_u = \bar{y}_v]\!] + [\![\nexists u \in \mathcal{V} : \bar{y}_u = y_v]\!] \right). \end{aligned} \tag{2.27}$$

Верность (2.26) следует из того факта, что для любых $a \in \{0, 1\}$, $b \in \{0, 1\}$ верно $\frac{1}{2}(a + b) \leq \max\{a, b\} \leq a + b$, что может быть проверено непосредственно. □

На практике значение коэффициентов s_k в определении слабой функции потерь неизвестно. Обозначим число пикселей на изображении $s = \sum_{v \in \mathcal{V}} c_v$. Будем считать, что эта величина распределена мультиномиально над допустимыми метками классов: $\{s_k\}_{k \in \mathbf{z}} \sim \mathcal{M}(\mathbf{q}, s)$, где $\mathbf{q}$ — параметры мультиномиального распределения.

Теорема 2.2. *Пусть $\hat{s}_k = q_k s$, тогда $\text{K}_{il}(\bar{\mathbf{y}}, \mathbf{z}; \hat{s}_k) = \mathsf{E}\text{K}_{il}(\bar{\mathbf{y}}, \mathbf{z}; s_k)$, где матожидание берётся по распределению $\{s_k\}_{k \in \mathbf{z}} \sim \mathcal{M}(\mathbf{q}, s)$, то есть $\hat{s}_k$ обеспечивает несмещённую оценку слабой функции потерь.*

Доказательство.

$$\mathsf{E}\mathrm{K}_{\mathrm{il}}(\bar{\mathbf{y}}, \mathbf{z}; s_k) = \sum_{k \notin \mathbf{z}} \sum_{v \in \mathcal{V}} c_v [\![\bar{y}_v = k]\!] + \mathsf{E} \sum_{k \in \mathbf{z}} s_k \prod_{v \in \mathcal{V}} [\![\bar{y}_v \neq k]\!] =$$
$$\sum_{k \notin \mathbf{z}} \sum_{v \in \mathcal{V}} c_v [\![\bar{y}_v = k]\!] + \sum_{k \in \mathbf{z}} \mathsf{E} s_k \prod_{v \in \mathcal{V}} [\![\bar{y}_v \neq k]\!] = \sum_{k \notin \mathbf{z}} \sum_{v \in \mathcal{V}} c_v [\![\bar{y}_v = k]\!] + \sum_{k \in \mathbf{z}} \hat{s}_k \prod_{v \in \mathcal{V}} [\![\bar{y}_v \neq k]\!]. \quad (2.28)$$

□

Параметры распределения $\mathbf{q}$ могут быть оценены по полностью размеченной части выборки. Однако на практике размеченных изображений мало, и оценка получается неустойчивой. В этом случае разумно предположить равномерные $\mathbf{q}$. Таким образом, в экспериментах используется следующая слабая функцию потерь по метке изображений:

$$\mathrm{K}_{\mathrm{il}}(\mathbf{y}, \mathbf{z}) = \sum_{k \notin \mathbf{z}} \sum_{v \in \mathcal{V}} c_v [\![y_v = k]\!] + \sum_{k \in \mathbf{z}} \frac{s}{|\mathbf{z}|} \prod_{v \in \mathcal{V}} [\![y_v \neq k]\!]. \quad (2.29)$$

При заданной слабой функции потерь K_{il} необходимо продемонстрировать алгоритмы для задач вывода в (2.4). Для вывода, согласованного с аннотацией $\max_{\mathbf{y} \in L(\mathbf{z}^i)} \mathbf{w}^\intercal \boldsymbol{\psi}(\mathbf{y}; \mathbf{x}^i)$ используется α-расширение только над метками из $\mathbf{z}^i$. Это может привести к несогласованной разметке — некоторые метки из $\mathbf{z}^i$ могут отсутствовать в $\mathbf{y}$. Предлагается использовать следующую эвристику для того, чтобы сделать найденную разметку удовлетворяющей ограничению. Для каждой метки k, такой что $k \in \mathbf{z}^i$ и $k \notin \mathbf{y}$, находится суперпиксель $\hat{v} = \operatorname{argmax}_{v \in \mathcal{V}} \mathbf{w}^\intercal \boldsymbol{\psi}(\mathbf{y}|_{y_v = k}; \mathbf{x}^i)$, где под $\mathbf{y}|_{y_v = k}$ понимается вектор, в котором компонента, соответствующая индексу переменной v, заменена на k. В качестве новой разметки выбирается $\mathbf{y}|_{y_{\hat{v}} = k}$. На практике применение этой эвристики не даёт значимого улучшения по сравнению с использованием несогласованных разметок.

Вывод, дополненный потерями, теперь не разделяется на унарные и парные потенциалы. Покажем, что его можно осуществлять с помощью метода минимизации энергии с штрафами за использование меток [52].

Лемма 2.3. *Вывод, дополненный слабой функцией потерь* K_{il}*, может быть выполнен как минимизация парно-сепарабельной энергии с дополнительными потенциалами вида* (1.47)*.*

Доказательство. Преобразуем выражение:

$$\operatorname*{argmax}_{\bar{\mathbf{y}} \in \mathcal{Y}} \{\mathbf{w}^\intercal \boldsymbol{\psi}(\bar{\mathbf{y}}; \mathbf{x}) + \mathrm{K}(\bar{\mathbf{y}}; \mathbf{z})\} =$$
$$\operatorname*{argmin}_{\bar{\mathbf{y}} \in \mathcal{Y}} \left\{ - \mathbf{w}^\intercal \boldsymbol{\psi}(\bar{\mathbf{y}}; \mathbf{x}) - \sum_{k \notin \mathbf{z}} \sum_{v \in \mathcal{V}} c_v [\![\bar{y}_v = k]\!] + \sum_{k \in \mathbf{z}} \frac{s}{|\mathbf{z}|} [\![\exists v \in \mathcal{V} : \bar{y}_v = k]\!] + \mathrm{const} \right\}. \quad (2.30)$$

Первые два члена под минимумом разделяются на унарные и ассоциативные парные потенциалы, а третий — потенциал, штрафующий присутствие меток в глобальном факторе. Для минимизации может использоваться эффективный алгоритм на основе α-расширения [52]. □

2.2.3 Плотные рамки

Объекты на изображении удобно аннотировать плотными рамками. С другой стороны, сегменты фоновых категорий не соответствуют объектам, аморфны и часто их плотная рамка близка к границам изображения, поэтому рамки добавили бы мало информации к метке изображения. В этом разделе рассматриваются аннотации, которые состоят одновременно из рамок и меток изображения. Например, для изображения могут быть заданы рамки для автомобилей и пешеходов, а также известно, что ещё присутствуют пиксели зданий, дороги, неба. Будем предполагать, что в рамках конкретного изображения категория может быть задана либо рамками, либо меткой изображения, хотя тип аннотаций для категории может меняться от изображения к изображению (см. в разделе 2.5.3 пример, демонстрирующий когда это может быть полезно).

Определение 2.12 (*слабая функция потерь при наличии рамок*)**.** Пусть слабая аннотация изображения $\mathbf{z}$ задана парой $(\mathbf{z}^{\text{il}}, \mathbf{z}^{\text{bb}})$ метки изображения и множества рамочных аннотаций $\mathbf{z}^{\text{bb}}$. Разобьём множество меток $\mathcal{K}$ на три подмножества в соответствии со слабой аннотацией $\mathbf{z}$: метки, которые определены рамками ($\mathcal{K}_{\text{b}} = \bigcup_{z \in \mathbf{z}^{\text{bb}}} \mathit{label}(z)$), метки, которые присутствуют в других местах ($\mathcal{K}_{\text{p}} = \mathbf{z}^{\text{il}}$) и метки, которые отсутствуют на изображении ($\mathcal{K}_{\text{a}} = \mathcal{K} \setminus (\mathcal{K}_{\text{b}} \cup \mathcal{K}_{\text{p}})$). Множество суперпикселей $\mathcal{V}$ также разбивается: $\mathcal{V}_k = \left\{ v \in \mathcal{V} : \exists \mathbf{p} \in \bigcup_{\bar{z} \in \mathbf{z}^{\text{bb}}: \mathit{label}(\bar{z}) = k} \mathit{box}(\bar{z}) : v = v(\mathbf{p}) \right\}$ — объединение суперпикселей, находящихся хотя бы частично в рамках с меткой $k \in \mathcal{K}_{\text{b}}$, и $\mathcal{V}_0 = \mathcal{V} \setminus \bigcup_{k \in \mathcal{K}_{\text{b}}} \mathcal{V}_k$ (рис. 2.2c). Тогда объединённая слабая функция потерь выглядит так:

$$\begin{aligned} \mathrm{K}_{\text{il-bb}}(\mathbf{y}, \mathbf{z}) = \sum_{k \in \mathcal{K}_{\text{a}}} \sum_{v \in \mathcal{V}} c_v [\![y_v = k]\!] + \sum_{k \in \mathcal{K}_{\text{p}}} \sigma_k \prod_{v \in \mathcal{V}} [\![y_v \neq k]\!] + \\ \beta \sum_{\bar{z} \in \mathbf{z}^{\text{bb}}} \left(\sum_{p=\mathit{top}(\bar{z})}^{\mathit{bottom}(\bar{z})} \nu_p^{\bar{z}} \prod_{q=\mathit{left}(\bar{z})}^{\mathit{right}(\bar{z})} [\![y_{v(p,q)} \neq \mathit{label}(\bar{z})]\!] + \sum_{q=\mathit{left}(\bar{z})}^{\mathit{right}(\bar{z})} \omega_q^{\bar{z}} \prod_{p=\mathit{top}(\bar{z})}^{\mathit{bottom}(\bar{z})} [\![y_{v(p,q)} \neq \mathit{label}(\bar{z})]\!] \right) \\ + \sum_{k \in \mathcal{K}_{\text{b}}} \sum_{v \in \mathcal{V}_0} c_v [\![y_v = k]\!]. \end{aligned} \tag{2.31}$$

Первые два слагаемых несут такой же смысл, как в (2.29). Третье слагаемое штрафует *пустые* строки и столбцы внутри рамок, т.е. те, которые не содержат ни одного пикселя, выведенного как метка рамки (см. рис. 2.3). Последнее слагаемое штрафует метки рамок вне соответствующих рамок. Оценим параметры этой функции σ_k, β, $\nu_p^{\bar{z}}$, $\omega_p^{\bar{z}}$, предполагая, что половина каждой из рамок занята объектом соответствующей категории.

Теорема 2.3. *Предположим, что в неизвестной разметке изображения каждый пиксель внутри рамки z_i независимо принимает метку $\mathit{label}(\bar{z})$ с вероятностью 0.5, иначе принимает одну из меток в $\mathcal{K}_p$. Предположим снова, что количество пикселей для меток из $\mathcal{K}_p$ распределено мультиномиально с равномерными параметрами. Тогда, если рамки не пересекаются, при следующих параметрах оценка функции $\mathrm{K}_{\text{il-bb}}$ является несмещённой: $\nu_p^{\bar{z}} = (\mathit{right}(\bar{z}) - \mathit{left}(\bar{z}))/2$, $\omega_q^{\bar{z}} = (\mathit{bottom}(\bar{z}) - \mathit{top}(\bar{z}))/2$, $\sigma_k = (s + \sum_{v \in \mathbf{v}_0} c_v)/2|\mathbf{z}^{il}|$, $\beta = 1$.*

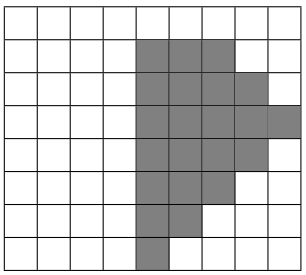


Рисунок 2.3: Пример разметки внутри рамки. Клетки соответствуют пикселям. Серые клетки помечены меткой, равной метке рамки, белые — остальными метками. Разметка не является плотной, так как верхняя строка и четыре левых столбца — пустые. Таким образом, в функции потерь 5 ненулевых слагаемых, соответствующих этой рамке.

Доказательство. Пусть $s^{\bar{z}}$ — количество пикселей внутри рамки $\bar{z}$, принадлежащих категории $\mathit{label}(\bar{z})$. По предположению теоремы оно распределено по биномиальному закону: $s^{\bar{z}} \sim \mathcal{B}(0.5, |\mathit{box}(\bar{z})|)$. Математическое ожидание этой величины равно $|\mathit{box}(\bar{z})|/2$. Пусть s^{il} — число пикселей изображения, относящихся к категориям из $\mathcal{K}_{\text{p}}$. Зная $s^{\bar{z}}$, можно оценить $s^{\text{il}} = s - \sum_{\bar{z}\in\mathbf{z}^{\text{bb}}} s^{\bar{z}}$. Рассуждая аналогично доказательству теоремы 2.3, получим оценку $\hat{\sigma}_k = \mathsf{E}s^{\text{il}}/|\mathbf{z}^{\text{il}}|$, которая позволяет несмещённо оценить $\text{K}_{\text{il-bb}}$. Поскольку s^{il} линейно зависит от $s^{\bar{z}}$, можно заменить последнее на его оценку. Отсюда

$$\hat{\sigma}_k = \frac{s - \sum_{\bar{z}\in\mathbf{z}^{\text{bb}}} |\mathit{box}(\bar{z})|/2}{|\mathbf{z}^{\text{il}}|} = \frac{s - \frac{1}{2}\sum_{k\in\mathcal{K}_{\text{b}}}\sum_{v\in\mathcal{V}_k} c_v}{|\mathbf{z}^{\text{il}}|} = \frac{s + \sum_{v\in\mathcal{V}_0} c_v}{2|\mathbf{z}^{\text{il}}|}. \tag{2.32}$$

Покажем несмещённость оценки, задаваемой третьим слагаемым на примере штрафа за пустые строки; для столбцов доказательство аналогично. Пусть $\hat{\nu}_p^{\bar{z}}$ — математическое ожидание числа пикселей категории $\mathit{label}(\bar{z})$ в строке p. Согласно модели, $\hat{\nu}_p^{\bar{z}} = (\mathit{right}(\bar{z}) - \mathit{left}(\bar{z}))/2$. Рассмотрим строки, в которых не найдено ни одного пикселя категории $\mathit{label}(\bar{z})$. Математическое ожидание ошибки на них равно $\hat{\nu}_p^{\bar{z}}$. Строки, в которых выведен хотя бы один пиксель категории $\mathit{label}(\bar{z})$, не штрафуются. Таким образом, при $\nu_p^{\bar{z}} = \hat{\nu}_p^{\bar{z}}$, третье слагаемое даёт несмещённую оценку на число неправильно классифицированных пикселей категории $\mathit{label}(\bar{z})$ в пустых строках рамки $\mathit{box}(\bar{z})$. □

Ещё более точную оценку можно получив, явно учтя в модели неравномерность распределения пикселей внутри рамки, для которых метка равна $\mathit{label}(\bar{z})$. Коэффициенты $\nu_p^{\bar{z}}$ и $\omega_q^{\bar{z}}$ позволяют варьировать штраф за пустые строки и столбцы соответственно, в зависимости от их расположения в рамке. При достаточном количестве полностью размеченных изображений можно обучить специфичные для категорий профили $\boldsymbol{\nu}^{\bar{z}}$ и $\boldsymbol{\omega}^{\bar{z}}$.

В предыдущем подразделе мы показали, как обрабатывать первые два слагаемых в выводе, дополненном функцией потерь — первое разделяется на унарные потенциалы, а второе представляет собой штраф за наличие метки. Последнее слагаемое также разделяется на унарные потенциалы. Третье слагаемое — сумма потенциалов высокого порядка. Для каждой рамки $\bar{z}$ каждая её строка и каждый столбец порождает потенциал над вершинами, соответствующими суперпикселям, которые пересекает эта строка/столбец. Так же как и в преды-

Алгоритм 2.1 Модификация алгоритма акцентирования для случая многоклассовой сегментации с ограничениями, задаваемыми рамочными аннотациями

1: **Вход:** Вектор признаков изображения $\mathbf{x}$, вектор параметров $\mathbf{w}$, множество рамочных аннотаций $\mathbf{z}^{\text{bb}}$, параметр плотности r.
2: **Выход:** разметка $\mathbf{y}$, согласованная с рамочными аннотациями $\mathbf{z}^{\text{bb}}$.

3: инициализировать унарные потенциалы $\phi_v(y_v) \leftarrow -\mathbf{x}_v^{\text{T}}\mathbf{w}^{\text{v}}_{y_v},\ \forall v \in \mathcal{V},\ \forall y_v \in \mathcal{K}$
4: инициализировать парные потенциалы $\phi_{vu}(y_v, y_u) \leftarrow -\mathbf{x}_{vu}^{\text{T}}\mathbf{w}^{\text{e}}_{y_v y_u}, \forall (v,u) \in \mathcal{E}, \forall (y_v, y_u) \in \mathcal{K}^2$
5: найти оптимальную разметку $\mathbf{y} \leftarrow \operatorname*{argmin}_{\mathbf{y}} \left\{ \sum_{v \in \mathcal{V}} \phi_v(y_v) + \sum_{(v,u)\in\mathcal{E}} \phi_{vu}(y_v, y_u) \right\}$
6: $\mathbf{z}^{\text{viol}} \leftarrow \left\{ \bar{z} \in \mathbf{z}^{\text{bb}} \mid \bar{z} \text{ не } r\text{-плотна относительно } \mathbf{y} \right\}$
7: **while** $\mathbf{z}^{\text{viol}} \neq \varnothing$ **do**
8: $\quad \mathcal{V}^{\text{viol}} \leftarrow \left\{ (v,k) \in \mathcal{V} \times \mathcal{K} \ \mid\ \exists \bar{z} \in \mathbf{z}^{\text{viol}},\ \exists \mathbf{p} \in \mathit{box}(\bar{z}) : \left(v = v(\mathbf{p}) \ \&\ k = \mathit{label}(\bar{z}) \ \&\ k \neq y_v \right) \right\}$
9: $\quad (\bar{v}, \bar{k}) \leftarrow \operatorname*{argmin}_{(v,k)\in\mathcal{V}^{\text{viol}}} \left\{ \phi_v(k) - \phi_v(y_v) \right\}$
10: $\quad \phi_v(\bar{k}) \leftarrow -\infty$ [2]
11: $\quad \mathbf{y} \leftarrow \operatorname*{argmin}_{\mathbf{y}} \left\{ \sum_{v \in \mathcal{V}} \phi_v(y_v) + \sum_{(v,u)\in\mathcal{E}} \phi_{vu}(y_v, y_u) \right\}$ # выполняется один шаг $\bar{k}$-расширения
12: $\quad \mathbf{z}^{\text{viol}} \leftarrow \left\{ \bar{z} \in \mathbf{z}^{\text{viol}} \mid \bar{z} \text{ не } r\text{-плотна относительно } \mathbf{y} \right\}$
13: **end while**

дущем разделе, штраф за присутствие метки $\mathit{label}(\bar{z})$ на соответствующих вершинах, но не на всём графе, представим в виде (1.47), и минимизируется модифицированной процедурой α-расширения [52].

При выводе, согласованном с рамочной аннотацией, необходимо вывести разметку, в которой только суперпиксели внутри рамок могут получать метки соответствующих объектных категорий, причём, в соответствии с определением r-плотной рамки, сегменты объектов должны быть связными и примыкать к рамке плотно, с допуском не более r от соответствующего измерения (напомним, что мы используем постоянное значение $r = 6\%$). Ограничение на метки вне рамок легко удовлетворяется при выводе: можно подавить нежелательные метки вне рамок, установив бесконечные унарные потенциалы.

Чтобы обеспечить плотность рамок, мы используем вариацию алгоритма *акцентирования* (англ. *pinpointing*) [67], модифицированного для работы с многоклассовой сегментацией, формально определённую в алгоритме 2.1. Это эвристический алгоритм, гарантирующий, что разметка будет обеспечивать плотность рамок, однако не гарантируется оптимальность в классе таких разметок. Сначала вывод выполняется без ограничений на плотность (строка 5). Затем, пока все ограничения не выполнены, одна из вершин меняет унарный потенциал (строка 10), и выполняется шаг расширения (строка 11). В нашей реализации выбирается вершина, соответствующую суперпикселю с наименьшим относительным потенциалом за $\mathit{label}(\bar{z})$ из тех, что ещё не получили эту метку (строка 9). Этой вершине назначается бесконечный[2] потенциал за метку $\mathit{label}(\bar{z})$, чтобы гарантировать, что метка вершины поменяется. Процедура

[2]Здесь под $-\infty$ понимается достаточно маленькое конечное число, чтобы при любых фиксированных значениях переменных $y_{\bar{v}},\ \forall \bar{v} \in \mathcal{V} \setminus \{v\}$ минимум энергии будет достигался при $y_v = \bar{k}$

(a) Аннотация самолёта зерном

(b) Штраф за другую метку

Рисунок 2.4: (a) Объект категории ‘самолёт’ аннотирован зерном. (b) Штраф за аннотацию пикселя категорией, отличной от ‘самолёт’, гауссово убывающий в зависимости от расстояния от положения пикселя до положения зерна. Чем ярче пиксель отмечен красным, тем больше соответствующий штраф.

конечна, если ни один суперпиксель не пересекает рамки разных меток, поскольку на каждой итерации хотя бы один суперпиксель внутри некоторой $box(\bar{z})$ меняет метку на $label(\bar{z})$.

Эксперименты показали, что при использовании такого типа аннотаций важна инициализация латентных переменных при обучении LV-SSVM. Наилучший результат имел место, когда изначально *все* суперпиксели внутри $box(\bar{z})$ получили метку $label(\bar{z})$.

Заметим, что Кумар и др. [63] используют другой критерий для вывода, согласованного с аннотацией — они предлагают штрафовать пустые строки и столбцы внутри рамки (точная противоположность того, что предлагаемый алгоритм делает при выводе, дополненном рамочной функцией потерь). Эта эвристика не гарантирует плотность полученных сегментов внутри рамок.

2.2.4 Зёрна объектов

Рассмотрим аннотацию зёрнами, при которой известно, что один пиксель, предположительно располагающийся близко к центру сегмента, принадлежит данной категории. При выводе, согласованном с аннотацией, требуется, чтобы зёрна принадлежали указанному классу. Ассоциативные парные потенциалы обычно распространяют эту метку на соседние суперпиксели.

Определение 2.13 (*слабая функция потерь при наличии зёрен*)**.** Пусть слабая аннотация изображения $\mathbf{z}$ задана парой $(\mathbf{z}^{\text{il}}, \mathbf{z}^{\text{os}})$, где $\mathbf{z}^{\text{os}}$ — это множество аннотаций зёрнами: $\dot{z} = (\dot{\mathbf{p}}, \dot{k})$. Определим объединённую слабую функцию потерь так:

$$\mathrm{K}_{\text{il-os}}(\mathbf{y}, \mathbf{z}) = \sum_{k \in \mathcal{K}_{\text{a}}} \sum_{v \in \mathcal{V}} c_v [\![y_v = k]\!] + \sum_{k \in \mathcal{K}_{\text{p}}} \sigma_k \prod_{v \in \mathcal{V}} [\![y_v \neq k]\!] + \beta \sum_{\substack{(\dot{\mathbf{p}}, \dot{k}) \\ \in \mathbf{z}^{\text{os}}}} \sum_{\mathbf{p}} [\![y_{v(\mathbf{p})} \neq \dot{k}]\!] \exp\left(-\frac{\pi \|\mathbf{p} - \dot{\mathbf{p}}\|^2}{\tau_{\dot{k}}} \right). \tag{2.33}$$

Первые два слагаемых здесь несут тот же смысл, как в функции потерь для меток изображения. Третье слагаемое поощряет назначение метки зерна в его окрестности (рис. 2.4b).

В нём внутренняя сумма берётся по всем пикселям изображения, $\tau_{\dot{k}}$ — параметр, оценивающий количество пикселей категории $\dot{k}$, а π — отношение длины окружности к её диаметру. Покажем, как назначать параметры в этом случае.

Теорема 2.4. *Предположим, что в неизвестной разметке изображения число пикселей, отнесённых к меткам из* $\mathbf{z}^{il}$ *и* $\mathbf{z}^{os}$ *распределено мультиномиально с равномерными параметрами, и что для каждого зерна* $\dot{z} = (\dot{\mathbf{p}}, \dot{k})$ *вероятность пикселя* $\mathbf{p}$ *принять метку* $\dot{k}$ *определяется гауссовым парзеновским окном:* $\exp\left(-\pi\|\mathbf{p} - \dot{\mathbf{p}}\|^2/\tau_{\dot{k}}\right)$. *Тогда при следующих параметрах оценка функции* $\mathrm{K}_{il\text{-}os}$ *является несмещённой:*

$$\tau_{\dot{k}} = \frac{s}{(|\mathbf{z}^{il}| + \#Lab(\mathbf{z}^{os})) \cdot \#Obj(\mathbf{z}^{os}, \dot{k})}, \quad \sigma_k = \frac{s}{|\mathbf{z}^{il}| + \#Lab(\mathbf{z}^{os})}, \quad \beta = 1, \tag{2.34}$$

если при этом зёрна находятся достаточно далеко друг от друга, а именно, $\sum_{(\dot{\mathbf{p}},\dot{k})\in\mathbf{z}^{os}} \exp\left(-\pi\|\mathbf{p} - \dot{\mathbf{p}}\|^2/\tau_{\dot{k}}\right) \leq 1, \forall \mathbf{p}$. *Здесь* $\#Lab(\mathbf{z}^{os})$ *— число различных меток в* $\mathbf{z}^{os}$, *а* $\#Obj(\mathbf{z}^{os}, \dot{k})$ *— число зёрен метки* $\dot{k}$ *в* $\mathbf{z}^{os}$.

Доказательство. Аналогично доказательству теоремы 2.3 можно получить оценку числа пикселей, отнесённых к каждой из категорий при мультиномиальном распределении: $\sigma_k = s/\left(|\mathbf{z}^{\mathrm{il}}| + \#Lab(\mathbf{z}^{\mathrm{os}})\right)$. Согласно условию теоремы, в окрестности зерна $\dot{z} = (\dot{\mathbf{p}}, \dot{k})$ классификация пикселя $\mathbf{p}$ меткой, отличной от $\dot{k}$, влечёт в сильной функции потерь от неизвестной разметки слагаемое с математическим ожиданием $\exp\left(-\pi\|\mathbf{p} - \dot{\mathbf{p}}\|^2/\tau_{\dot{k}}\right)$. Исходя из линейности вхождения всех слагаемых, значение функции $\mathrm{K}_{\text{il-os}}$ равно математическому ожиданию функции потерь по неизвестным сильным разметкам. Остаётся определить масштаб парзеновского окна. При $\tau_{\dot{k}} = \sigma_{\dot{k}}/\#Obj(\mathbf{z}^{\mathrm{os}}, \dot{k})$, ожидаемое число меток в категориях из $\mathbf{z}^{\mathrm{os}}$ равно оценке на $\sigma_{\dot{k}}$ (при условии достаточной удалённости зёрен):

$$\#Obj(\mathbf{z}^{\mathrm{os}}, \dot{k}) \cdot \int_{\mathrm{dom}(v)} \exp\left(-\frac{\pi\|\mathbf{p} - \dot{\mathbf{p}}\|^2}{\dot{\tau}_{\dot{k}}}\right) d\mathbf{p} = \#Obj(\mathbf{z}^{\mathrm{os}}, \dot{k}) \cdot \tau_{\dot{k}} = \sigma_{\dot{k}}. \tag{2.35}$$

Из равенства (2.35) получим искомую оценку $\tau_{\dot{k}}$. □

Последний член функции потерь (2.33) декомпозируется на унарные потенциалы, так что вывод, дополненный функцией потерь, тривиален.

2.3 Обучение категоризации документов по слабой аннотации

В задаче категоризации документов каждый документ должен быть размечен подмножеством категорий (тегов). Эту задачу также можно рассматривать как задачу разметки: структурная метка документа представляет собой бинарный вектор $\mathbf{y} \in \{0, 1\}^K$, где K — общее число категорий. Пусть $\mathbf{x}$ — признаки документа, тогда структурная метка $\mathbf{y}$ определяется следующей максимизацией:

$$\mathbf{y}^* = \operatorname*{argmax}_{\mathbf{y}} \mathbf{w}^\intercal \psi(\mathbf{y}; \mathbf{x}) = \operatorname*{argmax}_{\mathbf{y}} \left\{ \sum_{k \in \mathcal{K}} [\![y_k = 1]\!] (\mathbf{x}^\intercal \mathbf{w}_k^{\mathrm{u}}) + \sum_{(k,l) \in \mathcal{K}^2} [\![y_k = 1]\!] [\![y_l = 1]\!] \mathbf{w}_{kl}^{\mathrm{p}} \right\}, \quad (2.36)$$

Первая сумма соответствует независимой линейной классификации для определения присутствия каждой из категорий, а вторая позволяет моделировать корреляции между ними. В отличие от (2.22), здесь не накладывается ограничение на неотрицательность парных весов $\mathbf{w}_{kl}^{\mathrm{p}}$, что позволяет моделировать отрицательные корреляции. Вывод осуществляется с помощью простого алгоритма итерационного усреднения мод условных распределений (ICM), который эффективно применяется в подобных задачах с небольшими полносвязными графами. В качестве функции потерь для полной разметки будем снова использовать расстояние Хэмминга:

$$\Delta_{\mathrm{ml}}(\bar{\mathbf{y}}, \mathbf{y}^j) = \sum_{k \in \mathcal{K}} [\![\bar{y}_k \neq y_k^j]\!]. \quad (2.37)$$

Определение 2.14. *Частичной разметкой* документа будем называть троичный вектор $\mathbf{z} \in \{0, 1, ?\}^K$, где $z_k = ?$ означает, что информация о присутствии данной категории отсутствует. Пусть $\mathbf{y}$ — структурная метка документа. Частичная разметка $\mathbf{z}$ является корректной слабой аннотацией ($L(\mathbf{z}) \ni \mathbf{y}$), если $\forall k \in \mathcal{K} : z_k = y_k \vee z_k = ?$.

Определение 2.15. Пусть p_k — некоторые параметры $\forall k \in \mathcal{K}$, тогда *слабая функция потерь по частичной разметке* определяется как

$$K_{\mathrm{ml}}(\mathbf{y}, \mathbf{z}) = \sum_{k \in \mathcal{K}} \Big([\![z_k \neq ?]\!] [\![y_k \neq z_k]\!] + [\![z_k = ?]\!] \big(p_k [\![y_k = 0]\!] + (1 - p_k) [\![y_k = 1]\!] \big) \Big), \quad (2.38)$$

Эта функция потерь моделирует ситуацию, при которой оператор, аннотирующий выборку, забывает проставить метку категории, либо вносит лишнюю.

Теорема 2.5. *Пусть p_k — априорные вероятности отнесения документа к каждой из категорий $\forall k \in \mathcal{K}$. Тогда частичная функция потерь* (2.38) *является несмещённой оценкой функции потерь* (2.37).

Доказательство.

$$\begin{aligned} \mathsf{E}_{\bar{\mathbf{y}} \in L(\mathbf{z})} \Delta_{\mathrm{ml}}(\bar{\mathbf{y}}, \mathbf{y}) = & \sum_{\substack{k \in \mathcal{K}: \\ z_k \neq ?}} [\![\bar{y}_k \neq y_k]\!] + \sum_{\substack{k \in \mathcal{K}: \\ z_k = ?}} \big(\mathsf{P}(\bar{y}_k = 1) [\![y_k = 0]\!] + \mathsf{P}(\bar{y}_k = 0) [\![y_k = 1]\!] \big) = \\ & \sum_{k \in \mathcal{K}} \Big([\![z_k \neq ?]\!] [\![z_k \neq y_k]\!] + [\![z_k = ?]\!] \big(p_k [\![y_k = 0]\!] + (1 - p_k) [\![y_k = 1]\!] \big) \Big) = \\ & K_{\mathrm{ml}}(\mathbf{y}, \mathbf{z}). \end{aligned} \quad (2.39)$$

□

Вероятности p_k могут оцениваться по сильной части выборки, либо для каждой категории отдельно, либо одинаковые для всех категорий (это целесообразно, если данных недостаточно

для точной оценки). Вывод, согласованный с аннотацией, производится с помощью максимизации по неизвестным компонентам $\mathbf{z}$ при фиксированных известных. Слабая функция потерь разделяется на унарные потенциалы, поэтому вывод, дополненный функцией потерь, тривиален.

2.4 Обзор литературы

Вежневец и др. [60] решают задачу обучения сегментации по слабоаннотированным данным с помощью парадигмы обучения по нескольким прецедентам (англ. *multiple-instance learning*), в которой объекты обучающей выборки объединены в группы, про которые известно, что как минимум один из элементов принадлежит некоторой категории. Авторы предлагают новую модель нескольких изображений (англ. *multi-image model, MIM*) для регуляризации обучения. Она представляет собой марковскую сеть, в которой похожие суперпиксели *различных* изображений соединены парными связями. При обучении настраиваются её параметры, а при выводе в сеть включаются новые изображения, суперпиксели которых связываются с похожими суперпикселями обучающих и тестовых изображений. В оригинальной модели параметры унарных потенциалов настраиваются независимо, а парные потенциалы задаются вручную. Позже авторы предложили использовать Гауссовские процессы для настройки парных потенциалов [61]. Предложенная модель показывает качество сегментации, сравнимое с обучением по полной разметке. Модель нескольких изображений может быть настроена также с помощью предлагаемого метода.

Лу и др. [64] исследовали несколько вариаций формулировки структурного SVM с латентными переменными и предложили использовать ограничения с предыдущей внешней итерации для ускорения оптимизации. Метод тестировался на задаче отслеживания делящихся клеток, которая сводится к минимизации бинарной энергии марковской сети.

Кумар и др. [63] предложили метод обучения сегментатора по разнородным слабым аннотациям (в их терминологии, по разнородным источникам меток). Метод сначала обучает структурный SVM с латентными переменными по общей аннотации фона или переднего плана, то есть, для некоторых изображений обучающей выборки размечены только суперпиксели фона, для остальных — только суперпиксели переднего плана (т.е. объектных категорий). Назовём такой тип аннотации *регулярным*. Для обучения модели определяется слабая функция потерь, не разделяемая по факторам. После того, как модель обучена на данных с регулярной аннотацией, могут быть использованы данные с другими типами аннотаций: метками уровня изображения и рамками объектов. Для этого при текущих параметрах модели выводится предполагаемая регулярная аннотация, совместная с соответствующими слабыми аннотациями. Данные с полученной регулярной аннотацией добавляются к предыдущим, и процесс обучения запускается снова, при этом за начальное приближение берутся выходные параметры предыдущей стадии. Таким образом, метод не нуждается в специализированных функциях потерь для разных типов аннотаций и соответствующих им алгоритмах вывода, так как все типы аннотации явно приводятся к регулярной аннотации.

Эксперименты показывают, что такое дообучение по данным с новыми типами аннотации лишь немного улучшает качество модели. В данной работе же вводятся специализированные функции потерь для различных типов аннотаций, причём они оптимизируются одновременно. Разработанный метод не нуждается в «разгоночных» данных, в то время как регулярный тип аннотации, как и полная разметка, трудоёмок в получении. В работе Кумара и др. вывод, дополненный функцией потерь, проводится с помощью алгоритма итерационного пересчёта мод условных распределений (англ. *iterated conditional modes, ICM*) с эвристической инициализацией. Все функции потерь, используемые в данной работе, основаны на расстоянии Хэмминга между разметками, поэтому вывод, дополненный аннотациями допускает эффективный точный вывод, либо достаточно точные аппроксимации с помощью алгоритма разрезов на графах. Кроме того, в данной работе используются другие типы слабых аннотаций.

Некоторые из используемых в данной работе функций потерь не разделяется по индивидуальным переменным, так что она связана с работами по структурному обучению сегментации изображений по полной разметке с неразделяемыми функциями потерь [21,22]. Плетчер и Кохли [22] используют функцию потерь с фактором высокого порядка, которая штрафует разницу в площади сегментов целевой категории для двух сегментаций. Они используют алгоритм разрезов на графах для эффективного точного вывода, дополненного функцией потерь. Тарлоу и Цемель [21] используют метод передачи сообщений для вывода, дополненного функцией потерь, при обучении структурного SVM с тремя различными функциями потерь высокого порядка: коэффициент Жаккара для пикселей целевой категории, заполненность рамки сегментом целевой категории, и локальную выпуклость края сегмента.

2.5 Эксперименты

2.5.1 Наборы данных, детали реализации, критерии качества

Наборы данных. Мы протестировали предложенный метод на двух наборах данных: MSRCv2[3] [2, 60] и SIFT-flow[4] [61, 69, 70]. Набор MSRC содержит 276 изображений в обучающей и 256 в тестовой выборке. Пиксели вручную отнесены каждый к одной из 23 категорий, хотя значительная их часть осталась неразмеченной. SIFT-flow содержит 2488 изображений в обучающей и 200 в тестовой выборке, они размечены с использованием 33 меток категорий.

Структура модели и признаки. Для набора MSRC суперпиксели получены с помощью авторской реализации детектора границ *gPb* [71]. Признаки унарных потенциалов следующие: гистограмма визуальных слов на основе дескриптора SIFT [72], построенная с помощью словаря из 512 слов, гистограмма цветов пикселей, построенная на словаре из 128 слов, гистограмма локаций на равномерной сетке 6×6. Объединённые векторы признаков нормализуются и отображаются в пространство более высокой размерности, где скалярное произведение приближает расстояние χ^2 из оригинального пространства (размерность векторов признаков

[3] http://research.microsoft.com/en-us/projects/objectclassrecognition/
[4] http://people.csail.mit.edu/celiu/LabelTransfer/code.html

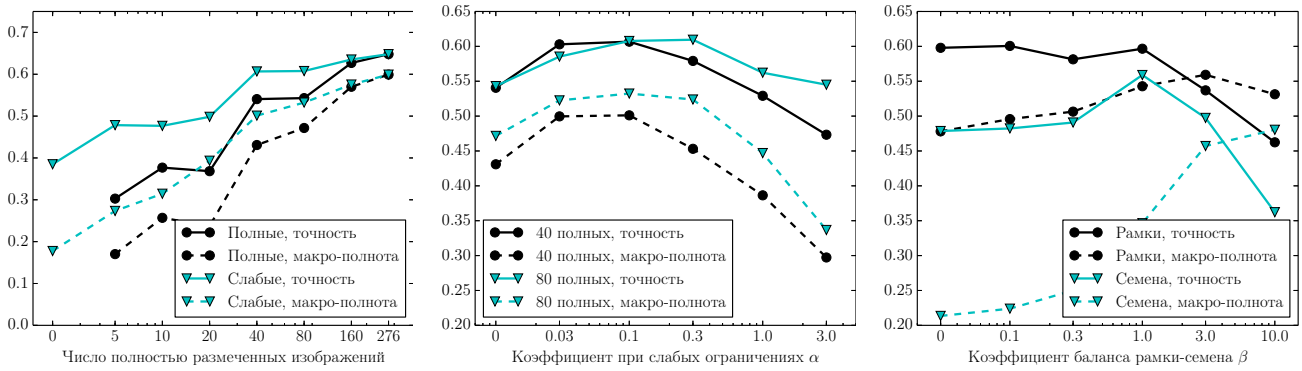
при этом утраивается) [73]. Признаки парных потенциалов состоят из 4 чисел: $\exp(-c_{ij}/10)$, $\exp(-c_{ij}/40)$, $\exp(-c_{ij}/100)$, 1. Здесь c_{ij} — сила границы между суперпикселями, соответствующими вершинам i и j, определённая детектором *gPb*.

Для набора SIFT-flow мы повторяем условия эксперимента Вежневца и др. [61]. Суперпиксели и признаки получены с помощью кода Тая и Лазебник [70]. Он использует графовую сегментацию Фельценсцвальба и Гуттенлохера [74] и затем вычисляет признаки для вычисления потенциальных функций. Унарные потенциалы зависят от формы, положения, текстуры и пиксельной маски суперпикселей и их окрестностей: всего 3115 унарных признаков. Мы также преобразуем их, приближая ядро χ^2, утраивая их размер [73]. Парные признаки вычисляются как расстояния над группами признаков суперпикселей (χ^2-расстояния для гистограмм, евклидовы в противном случае), всего 26 парных признаков.

Критерии качества. Мы используем два объективных критерия качества сегментации, которые вычисляются по размеченной тестовой выборке: *точность* (англ. *accuracy*) и *средняя поклассовая полнота* (англ. *per-class recall*). Пусть TP_k, FP_k, TN_k, FN_k — число истинно-положительных, ложноположительных, истинно-отрицательных и ложноотрицательных обнаружений для категории k, соответственно. Точность — это доля корректно распознанных пикселей тестовой выборки: $\frac{\sum_{k=1}^{K} TP_k}{\sum_{k=1}^{K} TP_k+FP_k}$. Поклассовая полнота — это число корректно размеченных пикселей каждой категории, делённое на суммарную площадь категории в верной разметке, усреднённое по категориям: $\frac{1}{K}\sum_{k=1}^{K} \frac{TP_k}{TP_k+FN_k}$. Следуя принятой практике [3, 60], мы исключили пиксели редких категорий (‘лошадь’ и ‘гора’) из подсчёта полноты для набора MSRC, однако учитываем метку ‘другое’, см. раздел 2.5.2. Аналогично мы не рассматриваем редкие категории (‘корова’, ‘пустыня’, ‘луна’, ‘солнце’) при подсчёте полноты на наборе SIFT-flow.

2.5.2 Метки изображений

Для создания тестовой выборки аннотация метками изображений получается автоматически из полной разметки: для каждого изображения берутся уникальные метки пикселей. Изображение из набора MSRC обычно содержит один или несколько объектов конкретной целевой категории (например, ‘знак’, ‘корова’, ‘автомобиль’) на некотором фоне. Не любую фоновую категорию можно отнести к используемым 23 меткам, так что часть изображения может остаться неразмеченной. На практике некоторые изображения содержат только одну метку категории. В этом случае метка изображения однозначно определяет полную разметку. Чтобы избежать этого знания (нереалистичного при практическом использовании), мы моделируем дополнительную метку ‘другое’, к которой относятся все категории кроме обозначенных 23-х. Обычно разметки имеют нечёткие границы, так что границы между сегментами различных меток также неразмечены (рис. 2.1b). Если мы будем относить их к категории ‘другое’, это может внести лишний шум в обучающую выборку. Поэтому необходимо использовать метку ‘другое’ только для неразмеченных регионов, но не для границ. Мы используем следующий эвристический критерий для получения меток изображения: метка ‘другое’ вклю-



(a) (b) (c)

Рисунок 2.5: Точность (сплошные линии) и поклассовая полнота (штриховые линии) при различных параметрах на наборе данных MSRC. (a) Изменение числа полностью размеченных изображений. Линии с круглыми маркерами показывают точность на тестовой выборке, если используются только полностью размеченные изображения, с треугольными — когда остальная часть обучающей выборки аннотирована метками изображений. (b) Изменение коэффициента слабой функции потерь α. Линии с круглыми маркерами показывают точность сегментации, когда 40 изображений полностью размечены, с треугольными — когда 80 изображений; остальная часть обучающей выборки аннотирована метками изображений. (c) Изменение коэффициента функции потерь β для плотных рамок (круглые маркеры) или зёрен объектов (треугольные маркеры). Все 276 изображений аннотированы метками изображений, а также все объекты аннотированы рамками или зёрнами, соответственно.

чается в список меток изображения тогда и только тогда, когда изображение содержит только одну метку или не менее 30 % его пикселей неразмечены.

В нашей базовой постановке эксперимента имеется (возможно пустая) полностью размеченная часть обучающей выборки, при этом остальные изображения аннотированы метками изображений. Эти подмножества выбраны с помощью эвристического алгоритма так, чтобы пропорции меток в них отражали соответствующие пропорции во всей выборке. С помощью модификации алгоритма Метрополиса–Гастингса с большой принимающей вероятностью находится подмножество изображений заданного размера, такое что распределение меток категорий в нём близко к распределению в полной выборке по расстоянию χ^2. Это даёт хорошую аппроксимацию, но из-за неравномерной представленности категорий в выборке некоторые редкие классы отсутствуют в небольших подмножествах (таким образом, невозможно настроить модель для них, и они сильно уменьшают поклассовую полноту). Например, подмножество из 10 изображений не содержит представителей 4 категорий.

Рис. 2.5a показывает точность и поклассовую полноту для сегментации тестовой выборки для различных размеров полностью размеченной части обучающей выборки, по сравнению с обучением на только сильно размеченной части выборки. В наиболее интересном случае, когда менее 20 % обучающей выборки полностью размечены, слабо аннотированная подвыборка обеспечивает увеличение на 10–15 процентных пунктов и по точности, и по полноте. В случае отсутствия полных разметок, модель производит сегментацию с точностью 38 % и пол-

нотой 18 %, что можно считать хорошим результатом для сегментации на 22 метки (полнота при случайной разметке составила бы 4.5 %).

Когда в обучающей выборке одновременно присутствуют изображения с полной разметкой и со слабыми аннотациями, необходимо установить коэффициент α из (2.2). Рис. 2.5b показывает, что его оптимальное значение лежит ниже 1. Возможным объяснением этого факта является то, что слабо аннотированные изображения несут меньше информации, таким образом должны давать меньший вклад в целевую функцию. Для всех дальнейших экспериментов, где это применимо, мы используем $\alpha = 0.1$.

Поскольку наша реализация требовательна к ресурсам времени и памяти при обучении на наборе данных SIFT-flow (обучение длится до нескольких недель), нет возможности провести настолько же подробный набор экспериментов. Вместо этого здесь сравнивается обучение с полной разметкой со слабым обучением при фиксированной доле слабо аннотированных изображений, а именно при 256 полностью размеченных изображениях и 2232 — с метками изображений (Табл. 2.1). Эта слабообученная модель уступает обученной на полной разметке всего 2 п. п. по точности и 4 п. п. по полноте. Похожие результаты показала на этом наборе данных модель Вежневца и др. [61], которая также достигла полноты 21 % при тех же признаках и том же разбиении на суперпиксели, совсем не используя полностью размеченных изображений. Однако этот метод использует дополнительные эвристики, которые можно включить и в предлагаемую схему: используется хэширующий ансамбль экстремально-рандомизированных решающих деревьев для нелинейного преобразования признаков, дополнительно обучаются априорные распределения «объектности» пикселей и категорий изображения, а также суперпиксели различных изображений соединяются в общую графическую вероятностную модель.

Поскольку задача оптимизации, возникающая в SSVM с латентными переменными, невыпукла, алгоритм может остановиться в локальном минимуме или на плато целевой функции, так что желательна хорошая инициализация. В приведённых экспериментах начальная разметка для слабоаннотированной части выборки выводится с помощью модели, обученной по размеченной части выборки.

Таблица 2.1: Точность и средняя поклассовая полнота на наборе данных SIFT-flow. Первые две строки описывают обучение на подмножестве из 256 полностью размеченных изображений для моделей с парными потенциалами и без них, соответственно. Третья строка описывает обучение на наборе, где остальные 2232 изображения обучающей выборки аннотированы метками изображений. Последняя строка показывает результат обучения на полностью размеченной выборке из 2488 изображений.

эксперимент	точн	полн
256/256 полных, без парных связей (локальная)	0.574	0.167
256/256 полных, инициализация результатом локальной	0.620	0.176
256/2488 полных, инициализация 256/256	0.674	0.208
2488/2488 полных	0.696	0.246

Таблица 2.2: Точность (первое число в каждой ячейке) и поклассовая полнота (второе число) на наборе MSRC, при обучении 1) только с полной разметкой, 2) если метки изображений (il) также доступны для оставшейся части выборки, 3) зёрна объектов (os) также доступны для оставшейся части выборки, 4) плотные рамки (bb) объектов доступны, 5) и зёрна, и плотные рамки доступны. Числа в последней колонке равны между собой, так как при полностью размеченной выборке слабая аннотация не добавляет информации.

il	bb	os	0/276 полных	5/276 полных	276 полных
−	−	−	n/a	0.300/0.170	0.648/0.599
+	−	−	0.385/0.178	0.478/0.273	0.648/0.599
+	−	+	0.559/0.346	0.574/0.370	0.648/0.599
+	+	−	0.597/0.543	0.606/0.546	0.648/0.599
+	+	+	0.531/0.567	0.542/0.564	0.648/0.599

2.5.3 Добавление рамок и зёрен

По полной разметке были сгенерированы ещё два типа аннотаций для обучающих изображений набора MSRC. Плотные рамки и зёрна объектов хорошо описывают объектные категории, но прибавляют мало информации для фоновых. Например, небо может занимать значительную часть изображения, так что его рамка не намного меньше всего изображения. Мы поделили список категорий на две части: фоновые, в т. ч. ‘трава’, ‘небо’, ‘гора’, ‘вода’, ‘дорога’ и ‘другое’, и объектные, в которые вошли все остальные категории. Две категории, ‘здание’ и ‘дерево’, проявляют двойственную природу — они могут отражать как основной объект на фотографии, так и задний фон (например, лес). Используется следующую эвристика: для конкретного изображения здание или дерево считается фоном тогда и только тогда, когда помимо него на изображении есть другие представители объектных категорий. Мы добавляем к меткам изображений обучающей выборки либо плотные рамки объектов, либо их зёрна. Для не-объектных категорий по-прежнему доступны только метки изображений.

Плотные рамки и зёрна определяются по полной разметке неоднозначно. Будем называть сегментами компоненты связности в маске пикселей данной категории, полученной по разметке изображения. Каждому сегменту соответствует одна рамка или зерно, соответственно. В качестве плотной рамки, соответствующей сегменту, берётся максимальная 0.06-плотная рамка сегмента. В качестве зерна, соответствующего сегменту, мы используем его «полюс недоступности»: выполняется преобразование расстояния (англ. *distance transform*), для каждого пикселя внутри сегмента возвращающее расстояние до его границы. Пиксель, максимизирующий это расстояние, и считается зерном.

В таблице 2.2 собраны результаты обучения по разным комбинациям аннотаций. В колонках приведены результаты для разного количества полностью размеченных изображений в обучающей выборке (0, 5, или вся выборка). Строки соответствуют различным комбинациям аннотаций остальной части выборки. Если полная разметка недоступна, и зёрна, и рамки значительно улучшают результат по сравнению с использованием только меток изображений. Рамки особенно сильно повышают поклассовую полноту — они помогают лучше обучать объектные категории, которые обычно занимают меньшую часть изображения, чем фоновые

категории, и соответственно вносят низкий вклад в функцию потерь, основанную на Хэмминговом расстоянии. В целом, обучение лишь по слабой аннотации метками изображений и плотными рамками лишь на 5 % уступает обучению с полной разметкой и по точности, и по полноте. Зёрна объектов дают меньший прирост качества, однако их использование может быть оправдано, так как они проще в получении.

В функциях потерь (2.31) и (2.33) присутствует коэффициент β, отвечающий за относительный вклад в функцию потерь штрафа за нарушение рамочной и зерновой аннотации, соответственно. Теоретически, при $\beta = 1$ функции потерь являются оценками расстояния Хэмминга. Мы измерили качество модели, обученной при различных значениях коэффициента (см. рис. 2.5c). При значение $\beta = 1$ точность оказалась сравнительно высокой, что подтверждает гипотезу.

2.5.4 Категоризация документов

В этом разделе описывается эксперимент на задаче из вычислительной лингвистики. Используется база данных юридических документов *EUR-lex* с метками вида *subject matter* [75], в частности, для разделения на обучающую и тестовую выборки используется первое разделение кросс-валидации из базы. Всего имеются 17413 документов в обучающей и 1935 в тестовой выборках, каждый описан 5000 признаками TF-IDF. Каждому документу присвоены несколько из возможных 201 категории.

В данном эксперименте мы моделируем ситуацию, когда оператор пропускает некоторые категории: для каждого документа известно подмножество единиц, а наличие остальных категорий неизвестно. По аналогии с предыдущими экспериментами мы разделяем обучающую выборку на часть с полной разметкой и часть со слабой аннотацией. Применяется классификатор на основе ансамбля рандомизированных решающих деревьев, чтобы сократить количество признаков документа с 5000 до 2: вероятностный выход данного классификатора и постоянный признак для моделирования смещения. Этот классификатор настраивается на полностью размеченной части обучающей выборки, затем применяется к обучающей и тестовой выборкам для преобразования признаков. Для полностью размеченной части обучающей выборки берутся несмещённые оценки, полученные при обучении (для каждого объекта усредняются результаты решающих деревьев, не использовавших данный объект при обучении). В функции потерь по частичной разметке (2.38) мы устанавливаем постоянное значение $p_k = p$, оценивая p по полностью размеченной части обучающей выборки.

Мы измеряем среднюю по категориям f-меру на тестовой выборке, чтобы оценить качество категоризации [75]. В бинарной классификации f-мера — это среднее гармоническое между точностью и полнотой:

$$F = \frac{2PR}{P+R}; \quad P = \frac{TP}{TP+FP}; \quad R = \frac{TP}{TP+FN}. \tag{2.40}$$

При обучении на 10 % обучающей выборки, f-мера равна 68.6 %. При добавлении остальных 90 % выборки с частичной разметкой, f-мера увеличивается до 71.9 %, что очень близ-

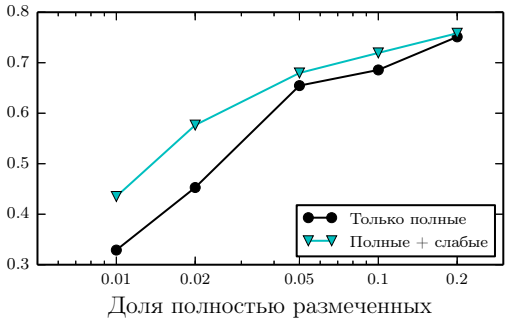


Рисунок 2.6: F-мера категоризации документов EUR-lex в зависимости от доли полностью размеченных документов (круглые маркеры), а также без полностью размеченных документов (треугольные маркеры).

ко к f-мере, полученной с помощью полностью размеченной целой выборки, т. е. 72.8 %. На рис. 2.6 приведены результаты для других соотношений.

2.6 Выводы

Предложен алгоритм структурного обучения по разнообразным аннотациям для задач разметки и общая схема определения функций потерь для различных типов аннотаций. В отличие от существующих, предложенный метод позволяет одновременно оптимизировать соответствующие им функции потерь, не сводя аннотации к более «полным» по жадной схеме. Метод применён для обучения семантической сегментации изображений по различным типам аннотаций, предложены специализированные функции потерь для меток изображений, плотных рамок и зёрен объектов, а также к задаче категоризации документов, для которой предложена функция потерь для неполной разметки. Описаны алгоритмы оптимизации, необходимые для обучения по слабой аннотации. Результаты показывают, что совместная аннотация, где фоновые категории заданы метками изображений, а объектные — плотными рамками, показывают лучшее качество сегментации тестовой выборки с учётом использованных при аннотировании трудозатрат.

Глава 3

Структурное обучение неассоциативных марковских сетей

Большинство прикладных задач разметки используют *ассоциативные* марковские сети. В этом случае поощряется назначение одной и той же метки переменным, соединённым ребром. В наиболее общем определении ассоциативной парно-сепарабельной марковской сети, каждый парный потенциал $\phi_{vu}(y_v, y_u)$ энергии (1.5) обладает свойствами метрики. Однако на практике не очень удобно параметризовывать пространство таких потенциалов, поэтому при необходимости обучения функционала энергии часто его сужают. Например, в главе 2 данной работы, как и в классической статье Таскара и др. [68], используется следующее обобщение модели Поттса: $\phi_{vu}(k, l) = 0$ при $k \neq l$; и $\phi_{vu}(k, k) = -\lambda_{vu}$, $\lambda_{vu} \geq 0$.

Популярность ассоциативных марковских сетей объясняется тем, что такой вид энергии допускает эффективные алгоритмы минимизации на основе разрезов на графах (раздел 1.2.4). Они позволяют выводить оптимальную разметку (с небольшой погрешностью), в том числе и в задаче вывода, дополненного функцией потерь, который выполняется на итерациях обучения структурного SVM, и, как правило, является самым ресурсоёмким местом в алгоритме. При этом ограничение естественного пространства потенциалов усложняет метод обучения. Например, при субградиентной оптимизации на каждой итерации необходимо брать проекцию на область допустимых значений параметров.

Ассоциативность может также служить формой регуляризации. Если известно, что связанные парными связями переменные не могут быть отрицательно коррелированы, то модель не будет настроена на такие шумовые зависимости в данных. Однако это предположение не всегда выполняется. Например, в задачах понимания сцены (т. е. семантической сегментации естественных сцен) марковская сеть может включать связи между удалёнными регионами, про которые известно, что они с большой вероятностью принадлежат разным категориям. Если связь соответствует суперпикселям, находящимся друг над другом вверху и внизу изображения, то они вероятнее принадлежат категориям ‘небо’ (верхний) и ‘трава’ (нижний), чем оба одновременно к любой из этих категорий.

Целевым приложением в этой главе является семантическая сегментация облаков точек, полученных лазерным сканированием естественных сцен (рис. 3.1). Масштаб сцен большой,

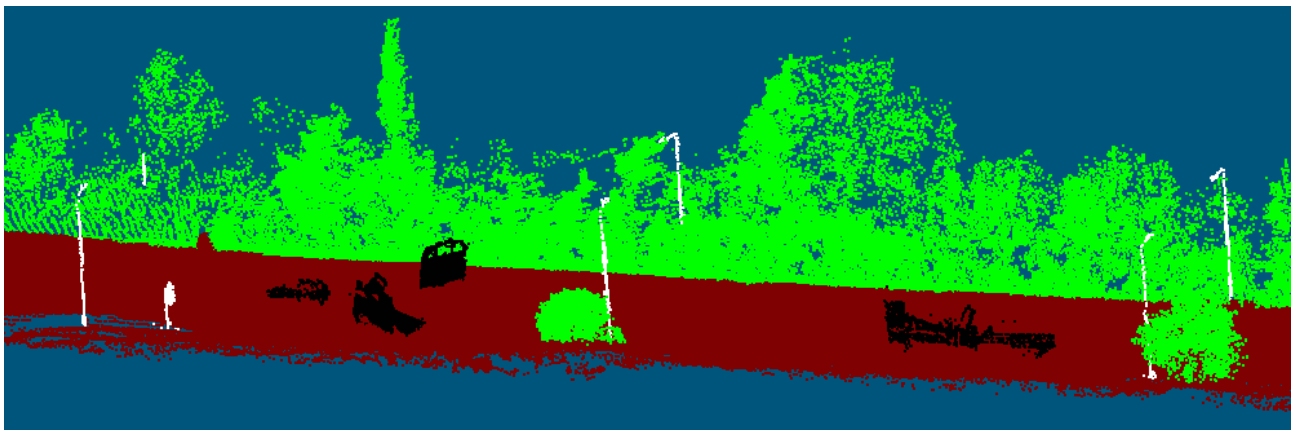


Рисунок 3.1: Типичное в геодезии облако точек, полученное лазерным сканированием. Цветом показана разметка, полученная вручную: красным — ‘земля’, чёрным — ‘автомобили’, зелёным — ‘растительность’, белым — ‘столбы’.

так как сканирование проводилось с движущегося самолёта или автомобиля. В отличие от данных, используемых в главах 4 и 2, данные не содержат цветовой информации — они представляют собой набор точек трёхмерного пространства, приближающих поверхности сцены. Поверхности представлены в данных неточно из-за шумов, загороженных поверхностей и бликов, возникающих при лазерной съёмке. Таким образом, локальные признаки гораздо хуже дискриминируют категории, чем локальные признаки в изображениях или облаках точек с цветовой информацией. Поэтому в данной задаче важнее анализировать геометрию сцены на глобальном уровне (уровне сцены) и на промежуточном уровне (уровне соседних объектов сцены). В настоящей главе предлагается модель для учёта неассоциативных взаимодействий на среднем уровне, таких как «дерево вероятно находится выше земли».

В данной главе предложена параметризация неассоциативной парно-сепарабельной марковской сети, а также предложен алгоритм обучения параметров на основе структурного SVM. Кроме того, предложена модификация функции потерь на основе Хэммингова расстояния, позволяющая настраивать параметры в случае, когда разные категории представлены в данных несбалансированно. Также показано, как обучать нелинейный структурный SVM с гауссовскими ядрами (англ. *gaussian radial basis function, RBF*). Эксперименты показывают, что эти нововведения позволяют улучшить качество сегментации.

3.1 Неассоциативная марковская сеть для сегментации облаков точек

Как и при работе с изображениями, будем считать элементарной единицей сегментации *суперпиксель*.

Определение 3.1. Рассмотрим облако точек в трёхмерном пространстве $\{\mathbf{p}_t\}_{t=1}^T$. Разбиением на суперпиксели назовём функционал $v : \{1, \dots, T\} \to \mathcal{V}$. Это разбиение выполняется так, чтобы прообразы $\mathcal{V}$ образовывали связные сегменты поверхности, приближаемой облаком.

Будем моделировать сегментацию облака точек с помощью парно-сепарабельной марковской сети над графом $G = (\mathcal{V}, \mathcal{E})$, переменные которой $\mathbf{y} \in \mathbb{R}^{d_\text{v}}$ соответствуют суперпикселям (далее для упрощения нотации будем отождествлять переменные с соответствующими им суперпикселями). На этапе вывода переменным назначаются метки категорий. Это означает, что все точки, относящиеся к данному суперпикселю, получают его метку. Парные потенциалы определены для всех пар близких суперпикселей. Более конкретно, для каждого суперпикселя определяется медоид (используется следующая аппроксимация: находится точка суперпикселя, ближайшая к центру масс), затем находятся k ближайших соседей в смысле медоидов. Объединение всех пар суперпикселей с каждым из его k ближайших соседей образуют множество $\mathcal{E}$ (используется значение $k = 5$). Обозначим $\mathbf{x}_v^\text{v} \in \mathbb{R}^{d_\text{v}}$ вектор признаков суперпикселя $v \in \mathbf{v}$, $\mathbf{x}_{uv}^\text{e} \in \mathbb{R}^{d_\text{e}}$ — вектор признаков, описывающий сходство соседних суперпикселей u и v, а $\mathbf{x} = \bigoplus_{v\in\mathcal{V}} \mathbf{x}_v^\text{v} \oplus \bigoplus_{(v,u)\in\mathcal{E}} \mathbf{x}_{vu}^\text{e}$ — их конкатенацию. Каждая переменная y_v, соответствующая суперпикселю v, принимает значение одной из меток категорий из множества $\mathcal{K} = \{1, \dots, K\}$. Пространство $\mathcal{X}$ содержит всевозможные признаки изображения $\mathbf{x}$, а пространство $\mathcal{Y}$ — всевозможные разметки $\mathbf{y}$ (на практике облака точек могут иметь разное число суперпикселей и разное число их соседних пар, однако в нотации этот факт игнорируется для простоты; обобщение на общий случай тривиально).

Снова рассмотрим логлинейную параметризацию (1.48) марковской сети. В ней сегментация ищется как MAP-оценка:

$$\mathbf{y}_\text{MAP} = \operatorname*{argmax}_{\mathbf{y}\in\mathcal{Y}} \mathbf{w}^\intercal \boldsymbol{\psi}(\mathbf{y}; \mathbf{x}) = \operatorname*{argmax}_{\mathbf{y}\in\mathcal{Y}} \sum_{v\in\mathcal{V}} \boldsymbol{\psi}_v(y_v; \mathbf{x})^\intercal \mathbf{w}^\text{v} + \sum_{(v,u)\in\mathcal{E}} \boldsymbol{\psi}_{vu}(y_v, y_u; \mathbf{x})^\intercal \mathbf{w}^\text{e}. \tag{3.1}$$

В предыдущей главе использовалась ассоциативная марковская сеть, то есть значение парного потенциала в формуле (2.22) было всегда неотрицательным. Для этого парные потенциалы для пары одинаковых меток приравнивались к нулю и вводились искусственные ограничения на признаки и параметры: $\mathbf{w}^\text{e} \geq \mathbf{0}$, $\mathbf{x}^\text{e} \geq \mathbf{0}$. В настоящей главе мы используем другую параметризацию парных потенциалов:

$$\boldsymbol{\psi}_{vu}(y_v, y_u; \mathbf{x})^\intercal \mathbf{w}^\text{e} = \sum_{k\in\mathcal{K}} \sum_{l\in\mathcal{K}} [\![y_v = k]\!][\![y_u = l]\!](\mathbf{x}_{vu}^\intercal \mathbf{w}_{kl}^\text{e}) = \mathbf{x}_{vu}^\intercal \mathbf{w}_{y_v y_u}^\text{e}. \tag{3.2}$$

В ней каждой паре меток сопоставлен свой вектор параметров, который скалярно умножается на вектор признаков ребра при данном назначении пары меток (это скалярное произведение может быть переопределено через ядра, как показано в разделе 3.3). При такой параметризации парные потенциалы перестают удовлетворять свойствам метрики, поэтому нельзя использовать максимизацию функционала (3.1) на основе разрезов на графах, в том числе для вывода, дополненного функцией потерь при обучении структурного SVM (раздел 1.2.4). Вместо этого используется алгоритм передачи сообщений на деревьях с перевзвешиванием (англ. *tree-reweighted message passing, TRW*) [43], один из вариантов двойственного разложения марковской сети на поддеревья. Он возвращает оценку снизу на значение функционала (3.1), что может приводить к раннему останову оптимизации функционала структур-

ного SVM. Такая аппроксимация называется *оптимизацией на расширенном множестве* (англ. *undergenerating*) [76]. Эксперименты показывают, что такая аппроксимация позволяет обучить качественный функционал (раздел 3.5).

3.2 Функция потерь для несбалансированных категорий

На практике различные категории могут быть представлены в обучающей выборке в разном объёме. В частности, фоновые категории, такие как ‘земля’ обычно содержат гораздо больше суперпикселей, чем объектные категории, такие как ‘автомобиль’ или ‘столб’. Цель обучения зависит от конечного приложения. В некоторых приложениях бывает нужно правильно распознать максимальное число точек, независимо от их категорий. В других основным объектом интереса являются объектные категории. Например, в задаче паспортизации придорожной инфраструктуры важно качественно выделять столбы. Поэтому эмпирическая функция потерь может явно учитывать штрафы за ошибки на отдельных категориях:

$$\Delta(\bar{\mathbf{y}}, \mathbf{y}) = \sum_{v \in \mathcal{V}} c_v r_{y_v} [\![\bar{y}_v \neq y_v]\!], \tag{3.3}$$

где $\mathbf{y}$ — верная разметка, $\bar{\mathbf{y}}$ — произвольная разметка, c_v — количество точек в суперпикселе, а r_k — штраф за неправильную классификацию точки категории k.

Лемма 3.1. *Пусть $\bar{R}$ — средняя полнота (recall) по категориям на обучающей выборке, состоящей из J объектов. Тогда сумма функций потерь на объектах обучающей выборки пропорциональна величине $(1 - \bar{R})$ при следующем значении параметров:*

$$r_k = \frac{\sum_{j=1}^{J} \sum_{v \in \mathcal{V}^j} c_v^j}{\sum_{j=1}^{J} \sum_{v \in \mathcal{V}^j} c_v^j [\![y_v^j = k]\!]}, \quad \forall k \in \mathcal{K}. \tag{3.4}$$

Доказательство. Преобразуем выражение:

$$\begin{aligned} 1 - \bar{R} = & \frac{1}{K} \sum_{k \in \mathcal{K}} \frac{FN_k}{TP_k + FN_k} = \frac{1}{K} \sum_k \frac{\sum_j \sum_v c_v^j [\![y_v^j = k]\!][\![\bar{y}_v^j \neq k]\!]}{\sum_j \sum_v c_v^j [\![y_v^j = k]\!]} = \\ & \frac{1}{K} \sum_j \sum_v \sum_k c_v^j \frac{1}{\sum_{j'} \sum_{v'} c_{v'}^{j'} [\![y_{v'}^{j'} = k]\!]} [\![y_v^j = k]\!][\![\bar{y}_v^j \neq k]\!] \propto \\ & \sum_j \sum_v \sum_k c_v^j r_k [\![y_v^j = k]\!][\![\bar{y}_v^j \neq k]\!] = \sum_j \sum_v c_v^j r_{y_v^j} [\![\bar{y}_v^j \neq y_v^j]\!] = \sum_j \Delta(\bar{\mathbf{y}}^j, \mathbf{y}^j). \end{aligned} \tag{3.5}$$

□

Согласно условиям леммы, штраф равен обратной частоте точек данной категории в обучающей выборке. Эксперименты в разделе 3.5 показывают, что такая модификация позволяет улучшить даже поточечную точность в случае несбалансированной выборки, если маленькие категории представлены достаточно, чтобы построить их модель.

3.3 Нелинейные ядра

Как и классический SVM, структурный его вариант допускает ядровой переход. Покажем это, сформулировав двойственную формулировку, и затем покажем, как её обобщить, заменив скалярное произведение в Евклидовом пространстве, заменив его на произвольную ядровую функцию.

3.3.1 Двойственная формулировка структурного SVM

Построим двойственную задачу к формулировке структурного SVM с линейными ограничениями (оптимизационная задача 1.6). Обозначим $\mathbf{Y} = \mathbf{y}^1 \oplus \mathbf{y}^2 \oplus \cdots \oplus \mathbf{y}^J$ конкатенацию разметок всех объектов выборки, $\mathbf{X} = \mathbf{x}^1 \oplus \mathbf{x}^2 \oplus \cdots \oplus \mathbf{x}^J$ — конкатенацию их признаков, $\Delta(\bar{\mathbf{Y}}; \mathbf{Y}) = \frac{1}{J}\sum_{j=1}^{J} \Delta(\bar{\mathbf{y}}^j; \mathbf{y}^j)$, и $\boldsymbol{\psi}(\bar{\mathbf{Y}}; \mathbf{X}) = \frac{1}{J}\sum_{j=1}^{J} \boldsymbol{\psi}(\bar{\mathbf{y}}^j; \mathbf{x}^j)$. Для упрощения нотации обозначим $\boldsymbol{\psi}_{\mathbf{Y}} \equiv \boldsymbol{\psi}(\mathbf{Y}; \mathbf{X})$, $\boldsymbol{\psi}_{\bar{\mathbf{Y}}} \equiv \boldsymbol{\psi}(\bar{\mathbf{Y}}; \mathbf{X})$, $\boldsymbol{\psi}_{\tilde{\mathbf{Y}}} \equiv \boldsymbol{\psi}(\tilde{\mathbf{Y}}; \mathbf{X})$, $\Delta_{\bar{\mathbf{Y}}} \equiv \Delta(\bar{\mathbf{Y}}; \mathbf{Y})$. Запишем оптимизационную задачу структурного SVM для объектов $\mathbf{X}, \mathbf{Y}$.

Оптимизационная задача 3.1 (формулировка SSVM с одной фиктивной переменной)**.**

$$\min_{\mathbf{w},\xi} \frac{1}{2}\mathbf{w}^\intercal\mathbf{w} + C\xi, \tag{3.6}$$

$$\text{при условиях} \quad \mathbf{w}^\intercal\boldsymbol{\psi}_{\mathbf{Y}} \geq \mathbf{w}^\intercal\boldsymbol{\psi}_{\bar{\mathbf{Y}}} + \Delta_{\bar{\mathbf{Y}}} - \xi, \ \forall\bar{\mathbf{Y}} \in \mathcal{Y}^J. \tag{3.7}$$

Эта задача называется формулировкой SSVM с одной фиктивной переменной (англ. *1-slack formulation*). Минимум по $\mathbf{w}$ в ней достигается в той же точке, что и в задаче 1.6 [20]. Следуя Йоахимсу и др. [20], запишем функцию Лагранжа, используя множители $\alpha_{\bar{\mathbf{Y}}}$, соответствующие ограничениям (3.7):

$$L(\mathbf{w}, \xi, \boldsymbol{\alpha}) = \frac{1}{2}\mathbf{w}^\intercal\mathbf{w} + C\xi + \sum_{\bar{\mathbf{Y}} \in \mathcal{Y}^J} \alpha_{\bar{\mathbf{Y}}} \left[\mathbf{w}^\intercal\boldsymbol{\psi}_{\bar{\mathbf{Y}}} - \mathbf{w}^\intercal\boldsymbol{\psi}_{\mathbf{Y}} + \Delta_{\bar{\mathbf{Y}}} - \xi\right]. \tag{3.8}$$

Для того чтобы найти его минимум по целевым переменным при $\boldsymbol{\alpha} \geq \mathbf{0}$, приравняем градиент к нулю:

$$\frac{\partial L}{\partial \mathbf{w}} = \mathbf{w} + \sum_{\bar{\mathbf{Y}} \in \mathcal{Y}^J} \alpha_{\bar{\mathbf{Y}}} \left(\boldsymbol{\psi}_{\bar{\mathbf{Y}}} - \boldsymbol{\psi}_{\mathbf{Y}}\right) = \mathbf{0} \quad \Rightarrow \quad \mathbf{w} = \sum_{\bar{\mathbf{Y}} \in \mathcal{Y}^J} \alpha_{\bar{\mathbf{Y}}} \left(\boldsymbol{\psi}_{\bar{\mathbf{Y}}} - \boldsymbol{\psi}_{\mathbf{Y}}\right), \tag{3.9}$$

$$\frac{\partial L}{\partial \xi} = C - \sum_{\bar{\mathbf{Y}} \in \mathcal{Y}^J} \alpha_{\bar{\mathbf{Y}}} = \mathbf{0} \quad \Rightarrow \quad \sum_{\bar{\mathbf{Y}} \in \mathcal{Y}^J} \alpha_{\bar{\mathbf{Y}}} = C. \tag{3.10}$$

Максимизируя функцию Лагранжа при этих ограничениях, получим двойственную задачу. Подставим в неё значение $\mathbf{w}$, полученное в (3.9).

Алгоритм 3.1 Обучение двойственной формулировки SSVM методом секущей плоскости

1: **Вход:** обучающая выборка $(\mathbf{X}, \mathbf{Y})$, гиперпараметры C, ε.
2: **Выход:** параметры $\boldsymbol{\alpha}$.
3: $\mathcal{W} \leftarrow \varnothing$
4: **repeat**
5: $\quad \bar{\mathbf{Y}} \leftarrow \operatorname{argmax}_{\bar{\mathbf{Y}} \in \mathcal{Y}^J} \left\{ \sum_{\bar{\mathbf{Y}} \in \mathcal{W}} \alpha_{\bar{\mathbf{Y}}} \left(\boldsymbol{\psi}_{\mathbf{Y}}^\intercal \boldsymbol{\psi}_{\bar{\mathbf{Y}}} - \boldsymbol{\psi}_{\bar{\mathbf{Y}}}^\intercal \boldsymbol{\psi}_{\bar{\mathbf{Y}}} \right) + \Delta_{\bar{\mathbf{Y}}} \right\}$
6: $\quad$ **if** $\bar{\mathbf{Y}} \notin \mathcal{W}$ **then**
7: $\quad\quad \mathcal{W} \leftarrow \mathcal{W} \cup \{\bar{\mathbf{Y}}\}$
8: $\quad\quad \boldsymbol{\alpha} \leftarrow \underset{\boldsymbol{\alpha} \geq \mathbf{0}}{\operatorname{argmax}} \; -\frac{1}{2} \sum_{\bar{\mathbf{Y}} \in \mathcal{Y}^J} \sum_{\tilde{\mathbf{Y}} \in \mathcal{Y}^J} \alpha_{\bar{\mathbf{Y}}} \alpha_{\tilde{\mathbf{Y}}} H(\bar{\mathbf{Y}}, \tilde{\mathbf{Y}}) + \sum_{\bar{\mathbf{Y}} \in \mathcal{Y}^J} \alpha_{\bar{\mathbf{Y}}} \Delta(\bar{\mathbf{Y}}; \mathbf{Y})$
9: $\quad\quad\quad$ при условиях $\sum_{\bar{\mathbf{Y}} \in \mathcal{Y}^J} \alpha_{\bar{\mathbf{Y}}} = C; \quad \alpha_{\bar{\mathbf{Y}}} = 0, \; \forall \bar{\mathbf{Y}} \in \mathcal{Y}^J \setminus \mathcal{W}$
10: $\quad$ **end if**
11: **until** $\bar{\mathbf{Y}} \in \mathcal{W}$

Оптимизационная задача 3.2 (двойственная формулировка SSVM)**.**

$$\max_{\boldsymbol{\alpha} \geq \mathbf{0}} \; -\frac{1}{2} \sum_{\bar{\mathbf{Y}} \in \mathcal{Y}^J} \sum_{\tilde{\mathbf{Y}} \in \mathcal{Y}^J} \alpha_{\bar{\mathbf{Y}}} \alpha_{\tilde{\mathbf{Y}}} H(\bar{\mathbf{Y}}, \tilde{\mathbf{Y}}) + \sum_{\bar{\mathbf{Y}} \in \mathcal{Y}^J} \alpha_{\bar{\mathbf{Y}}} \Delta(\bar{\mathbf{Y}}; \mathbf{Y}), \tag{3.11}$$

$$\text{при условии} \quad \sum_{\bar{\mathbf{Y}} \in \mathcal{Y}^J} \alpha_{\bar{\mathbf{Y}}} = C, \tag{3.12}$$

где скалярное произведение разностей обобщённых признаков определено как

$$\begin{aligned} H(\bar{\mathbf{Y}}, \tilde{\mathbf{Y}}) &= \left(\boldsymbol{\psi}_{\mathbf{Y}} - \boldsymbol{\psi}_{\bar{\mathbf{Y}}} \right)^\intercal \left(\boldsymbol{\psi}_{\mathbf{Y}} - \boldsymbol{\psi}_{\tilde{\mathbf{Y}}} \right) = \boldsymbol{\psi}_{\mathbf{Y}}^\intercal \boldsymbol{\psi}_{\mathbf{Y}} - \boldsymbol{\psi}_{\mathbf{Y}}^\intercal \boldsymbol{\psi}_{\tilde{\mathbf{Y}}} - \boldsymbol{\psi}_{\bar{\mathbf{Y}}}^\intercal \boldsymbol{\psi}_{\mathbf{Y}} + \boldsymbol{\psi}_{\bar{\mathbf{Y}}}^\intercal \boldsymbol{\psi}_{\tilde{\mathbf{Y}}} \\ &= \boldsymbol{\psi}(\mathbf{Y}; \mathbf{X})^\intercal \boldsymbol{\psi}(\mathbf{Y}; \mathbf{X}) - \boldsymbol{\psi}(\mathbf{Y}; \mathbf{X})^\intercal \boldsymbol{\psi}(\tilde{\mathbf{Y}}; \mathbf{X}) - \boldsymbol{\psi}(\bar{\mathbf{Y}}; \mathbf{X})^\intercal \boldsymbol{\psi}(\mathbf{Y}; \mathbf{X}) + \boldsymbol{\psi}(\bar{\mathbf{Y}}; \mathbf{X})^\intercal \boldsymbol{\psi}(\tilde{\mathbf{Y}}; \mathbf{X}). \end{aligned} \tag{3.13}$$

Получим теперь выражение для вывода разметки тестового объекта $\tilde{\mathbf{x}}$, снова перейдя к двойственным переменным с помощью (3.9):

$$\mathbf{y}_{\text{MAP}} = \underset{\tilde{\mathbf{y}} \in \mathcal{Y}}{\operatorname{argmax}} \; \mathbf{w}^\intercal \boldsymbol{\psi}(\tilde{\mathbf{y}}; \tilde{\mathbf{x}}) = \underset{\tilde{\mathbf{y}} \in \mathcal{Y}}{\operatorname{argmax}} \sum_{\bar{\mathbf{Y}} \in \mathcal{Y}^J} \alpha_{\bar{\mathbf{Y}}} \left(\boldsymbol{\psi}_{\mathbf{Y}}^\intercal \boldsymbol{\psi}(\tilde{\mathbf{y}}; \tilde{\mathbf{x}}) - \boldsymbol{\psi}_{\bar{\mathbf{Y}}}^\intercal \boldsymbol{\psi}(\tilde{\mathbf{y}}; \tilde{\mathbf{x}}) \right). \tag{3.14}$$

В этом выражении используется также обучающая выборка $(\mathbf{X}, \mathbf{Y})$. Из него следует, что для вычисления потенциалов тестовой задачи максимизации необходимо суммировать $|\mathcal{Y}^J|$ слагаемых, однако вектор $\boldsymbol{\alpha}$ оказывается разреженным, следовательно, большинство из слагаемых — нулевые. Рассмотрим алгоритм секущей плоскости 1.1 для оптимизации SSVM. Вместо целевых переменных прямой задачи в нём можно обновлять целевые переменные двойственной. Алгоритм 3.1 демонстрирует такую модификацию. На каждой итерации решается двойственная задача к задаче SSVM на рабочем наборе ограничений (строки 8–9). Поскольку целевая функция выпукла, а ограничения линейны, оптимумы в прямой и двойственной задачах совпадают, а решения могут быть получены друг из друга с помощью (3.9). Поэтому последовательности $\bar{\mathbf{Y}}$, получаемые двумя вариантами алгоритма, совпадают.

На каждой итерации алгоритма 3.1 не более одной компоненты вектора $\boldsymbol{\alpha}$ может стать ненулевой. Поэтому количество ненулевых компонент в финальном решении ограничено сверху числом итераций, которое при фиксированной точности полиномиально зависит от длины выборки [20]. Другими словами, согласно условиям дополняющей нежёсткости в теореме Каруша–Куна–Таккера, ненулевыми переменными могут быть только те, которые соответствуют активным ограничениям в прямой задаче (неактивные ограничения выполняются с нестрогими неравенствами). При достижении сходимости алгоритма 1.1 активные ограничения входят в рабочий набор $\mathcal{W}$. Их размер ограничен многочленом от числа компонент в разметках, что существенно меньше экспоненциального числа $|\mathcal{Y}^J|$. Таким образом, решение получается существенно разреженным. Разметки, которым соответствуют ненулевые $\alpha_{\bar{\mathbf{y}}^*}$, называются опорными векторами. Они соответствуют наиболее неправдоподобным разметкам обучающих объектов. Из решающего правила (3.14) видно, что MAP-оценка стремится быть близкой по обобщённым признакам к верной разметке обучающей выборки $\mathbf{y}^*$, но далёкой от опорных векторов.

3.3.2 Ядровой переход

Можно заметить, что ни в формулировке целевой функции (3.11), ни в решающем правиле (3.14) не фигурирует вектор параметров $\mathbf{w}$. Все зависимости между признаками выражаются через скалярные произведения обобщённых признаков. Будем называть такое произведение *ядровой функцией для выборок*:

$$\begin{aligned} Q(\mathbf{X}_1, \mathbf{Y}_1, \mathbf{X}_2, \mathbf{Y}_2) = \boldsymbol{\psi}(\mathbf{Y}_1; \mathbf{X}_1)^\intercal \boldsymbol{\psi}(\mathbf{Y}_2; \mathbf{X}_2) = \frac{1}{J^2} \Bigg(\sum_{j=1}^{J} \boldsymbol{\psi}(\mathbf{y}_1^j; \mathbf{x}_1^j) \Bigg)^\intercal \Bigg(\sum_{i=1}^{J} \boldsymbol{\psi}(\mathbf{y}_2^i; \mathbf{x}_2^i) \Bigg) \\ = \frac{1}{J^2} \sum_{j=1}^{J} \sum_{i=1}^{J} \boldsymbol{\psi}(\mathbf{y}_1^j; \mathbf{x}_1^j)^\intercal \boldsymbol{\psi}(\mathbf{y}_2^i; \mathbf{x}_2^i) = \frac{1}{J^2} \sum_{j=1}^{J} \sum_{i=1}^{J} q(\mathbf{y}_1^j, \mathbf{x}_1^j, \mathbf{y}_2^i, \mathbf{x}_2^i). \end{aligned} \tag{3.15}$$

Функцию $q(\mathbf{y}_1, \mathbf{x}_1, \mathbf{y}_2, \mathbf{x}_2)$ назовём *ядровой функцией для объектов*. Переопределив функцию

$$H(\bar{\mathbf{Y}}, \tilde{\mathbf{Y}}) = \; Q(\mathbf{X}, \mathbf{Y}, \mathbf{X}, \mathbf{Y}) - Q(\mathbf{X}, \mathbf{Y}, \mathbf{X}, \tilde{\mathbf{Y}}) - Q(\mathbf{X}, \bar{\mathbf{Y}}, \mathbf{X}, \mathbf{Y}) + Q(\mathbf{X}, \bar{\mathbf{Y}}, \mathbf{X}, \tilde{\mathbf{Y}}) \tag{3.16}$$

и решающее правило

$$\mathbf{Y}_{\text{MAP}} = \operatorname*{argmax}_{\tilde{\mathbf{Y}} \in \mathcal{Y}} \sum_{\bar{\mathbf{Y}} \in \mathcal{Y}^J} \alpha_{\bar{\mathbf{Y}}} \Big[Q(\tilde{\mathbf{X}}, \tilde{\mathbf{Y}}, \mathbf{X}, \mathbf{Y}) - Q(\tilde{\mathbf{X}}, \tilde{\mathbf{Y}}, \mathbf{X}, \bar{\mathbf{Y}}) \Big], \tag{3.17}$$

получим формулировку задач обучения и вывода, содержащую только ядровые функции, но не обобщённые признаки в явном виде. Скалярное произведение в ядровой функции (3.15) можно заменить на другую функцию. При этом, как и в линейном случае, должен существовать эффективный алгоритм для вывода (3.17), дополненного функцией потерь. Это гарантируется в том случае, если ядро разделяется на факторы соответствующей марковской сети. В случае парно-сепарабельной марковской сети ядровая функция должна быть представима

в виде суммы унарных и парных потенциалов относительно компонент вектора — второго аргумента. Приведём пример такой функции, имеющей практическое значение.

Определение 3.2. *Гауссовской ядровой функцией* (англ. *gaussian radial basis function, RBF*) будем называть ядровую функцию для объектов следующего вида:

$$q(\mathbf{y}', \mathbf{x}', \mathbf{y}'', \mathbf{x}'') = \sum_{v' \in \mathcal{V}'} \sum_{v'' \in \mathcal{V}''} \exp(-\gamma \|\mathbf{x}'_{v'} - \mathbf{x}''_{v''}\|^2) [\![y'_{v'} = y''_{v''}]\!] + \sum_{\substack{(v',u') \\ \in \mathcal{E}'}} \sum_{\substack{(v'',u'') \\ \in \mathcal{E}''}} \exp(-\gamma \|\mathbf{x}'_{v'u'} - \mathbf{x}''_{v''u''}\|^2) [\![y'_{v'} = y''_{v''}]\!] [\![y'_{u'} = y''_{u''}]\!]. \tag{3.18}$$

Здесь под $\mathcal{V}'$ и $\mathcal{V}''$ понимаются множества вершин марковских сетей, образующих $\mathbf{x}'$ и $\mathbf{x}''$, соответственно, а под $\mathcal{E}'$ и $\mathcal{E}''$ — множества их рёбер. Параметр γ отражает ширину ядра, мы полагаем его равным 1.

Гауссовская ядровая функция измеряет расстояние между разметками с учётом близости признаков. Рассматриваются все пары унарных и все пары парных потенциалов, и чем ближе соответствующие признаки, тем больший вклад в расстояние дают неодинаковые значения переменных в соответствующих компонентах разметок.

Согласно результатам экспериментов в разделе 3.5, использование гауссовского ядра позволяет повысить точность сегментации за счёт ослабления требования на линейность зависимости от признаков в решающем правиле.

3.4 Обзор литературы

Ангелов и др. [5] впервые предложили использовать марковские сети для сегментации облаков точек. Они использовали *ассоциативные* марковские сети, поощряющие одинаковые метки категорий для соседних точек. Хотя метод просто добавляет пространственную регуляризацию к индивидуальным классификаторам, его результат оказался значительно лучше за счёт повышения робастности. Достаточно простая формулировка функционала энергии допускает эффективную минимизацию с помощью разрезов на графах (раздел 1.2.4). При обучении настройка параметров выполняется с помощью стандартных методов квадратичного программирования, что ограничивает размер задачи.

При использовании ассоциативной модели на практике нет нужды использовать подробные признаки парных потенциалов. Модель недостаточно ёмкая, чтобы извлечь пользу из признаков парных потенциалов, которые только поощряют равенство меток инцидентных им точек. Как следствие, ранние работы [5, 77] используют один константный признак парных потенциалов. Другими словами, парные потенциалы задают априорные знания о совместной встречаемости конкретных меток в соседних точках. Позже Муноз и др. [32] использовали анизотропную ассоциативную модель, в которой парные потенциалы зависят от парных признаков, таких как модуль и направление вектора, соединяющего соответствующие точки. Позже они предложили моделировать факторы высоких порядков, но при этом не отказались от

требования ассоциативности [6]. В этой модели используются потенциалы Поттса (и парные, и высоких порядков): они могут быть положительными, только если все (или большинство) переменных в факторе принимают одну и ту же метку, иначе потенциал равен нулю. Такой вид потенциалов позволяет проводить эффективный вывод MAP-оценки (раздел 1.2.4).

Требование ассоциативности ограничивает гибкость модели. Ранее нами было показано, что учёт неассоциативных зависимостей, таких как «деревья и здания склонны находиться выше земли», позволяет повысить точность сегментации [28]. Также как и в этой главе использовался общий вид парных потенциалов, однако применялся более простой, эвристический метод обучения на основе наивного Байесовского классификатора, в котором правдоподобие оценивалось непараметрически, независимо для унарных и парных факторов, при этом на этапе обучения не моделировались корреляции между факторами и переменными. Частный случай этой модели используют Познер и др. [78]: их модель неассоциативная, но используется только константный парный признак, как и в ранних работах [5, 77], что соответствует использованию только априорного распределения в наивном Байесовском классификаторе.

В целом, обучение неассоциативных моделей более требовательно к обучающей выборке: в ней должны быть хорошо представлены межклассовые связи, а не только внутриклассовые. Ещё одной трудностью при их использовании является нерегулярность энергии, не допускающая эффективный вывод с помощью разрезов на графах (раздел 1.2.4), поэтому используют приближённые методы MAP-вывода (раздел 1.2.3). Хотя точность приближённых методов может быть достаточна на этапе принятия решения, неточный вывод может негативно сказаться при обучении. Финли и Йоахимс изучили проблему использования неточных методов вывода при структурном обучении [76]. Франк и Савчинский [33] использовали на практике неточный вывод при структурном обучении в задачах компьютерного зрения. Их эксперименты показали, что неассоциативная модель имеет немного более низкую точность, чем ассоциативная, что объясняется неточностью процедуры вывода, дополненного функцией потерь. Однако в этом эксперименте также использовался только константный парный признак, что могло помешать неассоциативной модели проявить свою гибкость.

В разделе 3.3 представлен метод обучения потенциалов марковской сети, нелинейно зависящих от признаков с использованием ядровых функций. Похожую идею использовали Трибель и др. [77], которые совместили ассоциативную марковскую сеть с метрическим классификатором на основе k ближайших соседей. Отличие предложенного метода в том, что опорные векторы выбираются не только из объектов обучающей выборки, а могут генерироваться из множества всевозможных разметок (при этом неправильно размеченные опорные векторы входят в решающие правило с отрицательными коэффициентами). Поскольку предложенный здесь метод разреженный, в нём может выбраться более компактное представление из опорных векторов. Муноз и др. [6] предложили другой метод для восстановления нелинейной зависимости — *функциональный градиентный бустинг* (англ. *functional gradient boosting, FGB*), совместно настраивающий потенциальные функции как нелинейные функции их параметров. Метод подробнее описан в разделе 1.3.3.

Марковские сети — не единственный способ учёта семантического пространственного контекста, который используется при сегментации облаков точек. Некоторые работы используют детектирование объектов с последующей сегментацией и классификацией форм [79,80]. Другие получают сегментацию как побочный продукт детектирования объектов определённого класса методами голосования в обобщённом пространстве Хафа [81,82]. Ряд методов используют последовательную классификацию. Один из них, *пространственная машина вывода*, описан в главе 4 данной диссертации. Сьон и др. [34] предложили идею *эшелонированного трёхмерного парсинга* (англ. *stacked 3D parsing*), который использует семантический контекст для разметки облака точек на различных уровнях подробности (от грубого до тонкого) и запускает последовательную классификацию для согласования разметок.

3.5 Эксперименты

В данном разделе проводится экспериментальная оценка предложенного метода и сравнение его с аналогами на двух наборах данных, полученных, соответственно, аэросъёмкой, и сканированием с движущегося автомобиля. Основная цель экспериментов — показать преимущество неассоциативных моделей в задаче семантической сегментации облаков точек. В качестве слабого базового метода используется ансамбль рандомизированных деревьев, применяемый к суперпикселям независимо. Также показано, что на этих наборах данных предложенный метод с нелинейными ядрами превосходит по качеству другие нелинейные методы, а именно функциональный градиентный бустинг для обучения ассоциативных марковских сетей [6] и наивное Байесовское обучение потенциалов неассоциативной марковской сети [28].

На наборе данных *Аэро* проводятся две серии экспериментов. В первой унарные потенциалы не используются совсем — она демонстрирует способность предложенного метода моделировать зависимости разметки от признаков парных потенциалов. Во второй серии экспериментов моделируется прикладное использование метода: унарные потенциалы назначаются как минус логарифмы вероятностного выхода ансамбля рандомизированных деревьев и фиксируются, а парные — настраиваются с помощью структурного SVM. Рассматриваются два типа функций потерь: расстояние Хэмминга и сбалансированная по категориям функция потерь, описанная в разделе 3.2. Также приводится результат для линейной неассоциативной модели. Предлагаемый метод также протестирован на сложном наборе данных *Авто*, проведён анализ его применимости.[1]

3.5.1 Детали реализации

Марковская сеть строится над суперпикселями облака точек. Это ускоряет выполнение алгоритма, кроме того, пространственная удалённость делает признаки парных потенциалов более информативными (если же строить марковскую сеть над индивидуальными точками, парные связи между ближайшими соседями будут иметь неинформативное направление из-за погрешности измерений, а разница характеристик, таких как восстановленные нормали к

[1] http://graphics.cs.msu.ru/en/science/research/3dpoint/classification

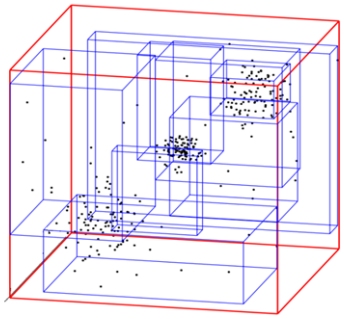


Рисунок 3.2: Визуализация структуры P-дерева с одним корнем и 8 листьями. Охватывающий параллелепипед для корневой вершины показан красным, для листьев — синим. В общем случае используются также промежуточные уровни иерархии.

поверхности, будет нулевой, за исключением шума). Мы используем вариант P-дерева [83] для построения пространственного индекса (структуры данных, позволяющей эффективно находить точки в заданной области), а также используем его листья для определения суперпикселей.[2] P-дерево представляет собой иерархию вложенных параллелепипедов, каждый из которых вмещает всех своих потомков, на нижнем уровне — точки облака. При построении P-дерева стремятся найти наиболее компактные параллелепипеды (рис. 3.2), поэтому его структуру можно использовать в качестве пересегментации. Согласно используемым настройкам, листья P-дерева содержат не более 64 точек каждый, также из процедуры построения следует пространственная близость точек, попавших в общий лист; в результате суперпиксель в среднем содержит около 50 точек. Для каждого суперпикселя находится приближённый медоид (точка множества, сумма расстояний до которой от остальных точек минимальна) как наиболее близкая точка суперпикселя к его центру масс. Для дальнейшего анализа медоид представляет весь суперпиксель: граф соседства строится с учётом расстояний между медоидами (каждый медоид соединяется с 5 ближайшими соседями), признаки вычисляются относительно медоидов и их пар (при этом для вычисления признаков используются все точки облака). Такое разбиение на суперпиксели оказывается критичным для сегментации облаков точек, как обсуждается в разделе 3.5.4.

Для обучения ансамбля рандомизированных деревьев, с помощью которого назначаются унарные потенциалы, вычисляются следующие признаки суперпикселей в фиксированной окрестности их медоидов:

- спектральные признаки и признаки направления [32];
- один из вариантов вариант спин-изображения;
- распределение точек по высоте [28].

Для двух суперпикселей, имеющих соседние медоиды $\mathbf{p}$ и $\mathbf{q}$ и нормали $\mathbf{n}_p$ и $\mathbf{n}_q$, соответственно, используются следующие признаки парных потенциалов:

- косинус угла между нормалями: $\mathbf{n}_p^{\mathrm{T}}\mathbf{n}_q/(\|\mathbf{n}_p\|\|\mathbf{n}_q\|)$;

[2] Авторская реализация опубликована в GML LidarK library: http://graphics.cs.msu.ru/en/science/research/3dpoint/lidark

- разница в высоте точек над землёй (т. е. в значениях проекций на ось z), нормированная на расстояние между ними: $(p_z - q_z)/\|\mathbf{p} - \mathbf{q}\|$.

Мы используем авторскую реализацию субградиентного обучения [32] и функционального градиентного бустинга [6][3]. В этих методах мы также используем функцию потерь для несбалансированных категорий из раздела 3.2, поскольку в противном случае результат этих моделей получается смещённым. Запускается неэкспоненцированный вариант FGB в течение $T = 100$ итераций с убывающим размером шага $\alpha_t = 1/\sqrt{t}$. Также как и авторы, мы подбираем параметр регуляризации на валидационной выборке. Для реализации наивного Байесовского обучения потенциалов также используются авторские параметры [28]: каждое из распределений приближается гистограммой из 10 ячеек. В предлагаемом методе параметр регуляризации подобран на валидационной выборке.

3.5.2 Наборы данных

Аэро. Этот набор получен лазерной аэросъёмкой (рис. 3.3a). Используются отдельные сканы для обучения и для тестирования, каждый содержит около 100 000 точек. Облака точек вручную размечены на три метки категории: ‘земля’, ‘здание’, ‘дерево’. Здания недопредставлены в наборе (около 1/12 всех точек), остальные точки принадлежат земле и растительности в равной пропорции.

Авто. Данные, снятые автомобилем, содержат около 0.4 миллиона точек в обучающей выборке и 1 миллион в тестовой (рис. 3.1). Используются четыре категории: ‘земля’, ‘транспорт’, ‘дерево’, ‘столб’ (последняя категория включает в себя и ламповые столбы, и столбы для дорожных знаков). В наборе только 0.2 % точек относятся к столбам, 5 % — к транспортным средствам, 12 % — к автомобилям, остальные точки принадлежат категории ‘земля’.

3.5.3 Результаты

Результат экспериментов на наборе *Аэро* приведены в таблице 3.1. Поскольку этот набор несбалансирован по категориям, приводятся точность и полнота (2.40) по всем категориям в отдельности. Также приводится среднее геометрическое полноты по категориям, которое трактует все категории одинаково важными, независимо от их размера [84]. Как и ожидалось, использование унарных потенциалов улучшает качество, что особенно заметно при обучении ассоциативной марковской сети функциональным градиентным бустингом. Это можно объяснить недостаточной ёмкостью ассоциативной модели. При этом, добавление унарных потенциалов не приводит к идеальному результату — оба варианта обучения неассоциативной марковской сети улучшают результат ассоциативной. Визуальные результаты сегментации представлены на рис. 3.3.

Предложенный метод приводит к разреженному решению: были определены всего 10 опорных векторов (хотя любая потенциальная разметка порождает возможный опорный вектор). При этом гауссово ядро (3.18) содержит сумму по всем факторам, так что применение

[3] http://www.cs.cmu.edu/~dmunoz/projects/m3n.html

Таблица 3.1: Точность и полнота для каждой из категорий и геометрическое среднее полноты по категориям на наборе *Аэро*. Приведены результаты независимой классификации ансамблем рандомизированных деревьев (UNARY), ассоциативной модели, обученных функциональным градиентным бустингом (FUNC), а также неассоциативных моделей, использующей наивный Байесовский классификатор (BAYES) и обученную предложенным методом (SVM). Постфикс «-PW» добавляется к моделям, не использующим унарные потенциалы. В последних двух строках приведены результаты упрощённых моделей: линейного структурного SVM (SVM-LIN) и нелинейного структурного SVM с невзвешенной Хэмминговой функцией потерь (SVM-HAM).

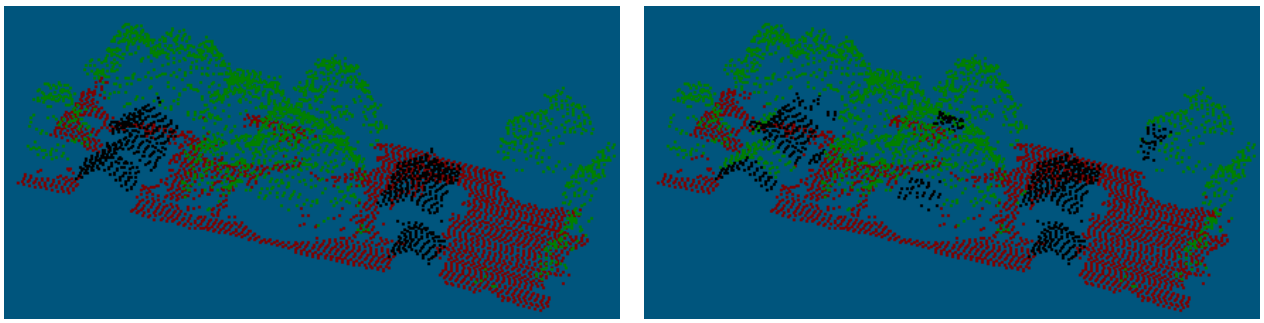
Метод	земля		здание		дерево		ср. геом.
	точность	полнота	точность	полнота	точность	полнота	полнота
UNARY	0.992	0.952	0.576	0.688	0.890	0.892	0.836
BAYES-PW	0.985	0.979	0.493	0.698	0.898	0.809	0.821
FUNC-PW	0.911	0.975	0.578	0.545	0.923	0.850	0.767
SVM-PW	0.981	0.977	0.602	0.803	0.924	0.849	0.874
BAYES	0.983	0.978	0.496	0.779	0.917	0.789	0.844
FUNC	0.975	0.981	0.758	0.645	0.913	0.940	0.841
SVM	0.975	0.979	0.574	**0.923**	**0.960**	0.805	**0.900**
SVM-LIN	**0.994**	**0.987**	**0.641**	0.693	0.907	0.896	0.850
SVM-HAM	0.952	0.985	0.612	0.181	0.813	**0.922**	0.548

Таблица 3.2: F-мера для результатов субградиентной оптимизации структурного SVM (SUB, [32]), функционального градиентного бустинга (FUNC), и предложенного метода (SVM-LIN, SVM) на наборе данных *Авто*

Метод	земля	транспорт	дерево	столб
SUB	0.974	0.302	0.497	0.138
FUNC	0.979	0.821	**0.934**	**0.397**
SVM-LIN	0.934	0.792	0.789	0.203
SVM	**0.980**	**0.868**	0.928	0.000

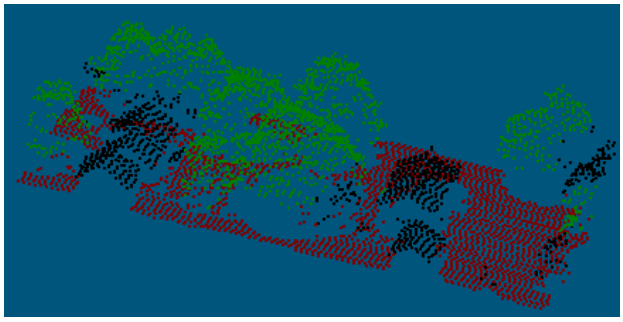
даже одного опорного вектора происходит значительно медленнее, чем в линейной модели, где ядра суммируются к набору параметров потенциалов. К сожалению, в этом эксперименте линейная модель лишь немного улучшает качество независимой классификации. Нижний ряд таблицы 3.1 показывает, что критично использовать взвешенное расстояние Хэмминга в качестве функции потерь (раздел 3.2) — обученная при отсутствии взвешивания модель склонна игнорировать мелкие классы, в данном случае, ‘здание’.

Результаты наиболее успешных методов (нелинейного структурного SVM и FGB, а также их линейных аналогов) на наборе *Авто* приведены в таблице 3.2. Приводится f-мера (среднее гармоническое между точностью и полнотой (2.40)) для каждой из категорий. Структурный SVM и FGB показывают аналогичные результаты на категориях ‘земля’ и ‘дерево’. Первый лучше классифицирует ‘транспорт’, но совершенно не находит ‘столбы’, которых было очень мало в обучающей выборке. Таким образом, предложенный метод плохо применим к данным, содержащим много категорий, а также когда некоторые категории сильно недопредставлены.

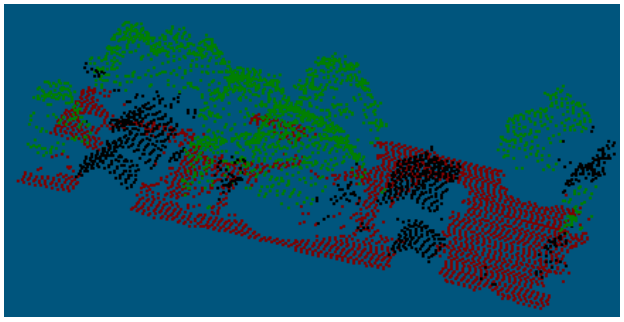


(a) GTRUTH

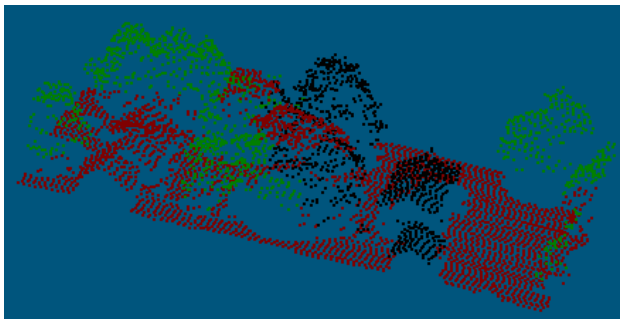
(b) UNARY



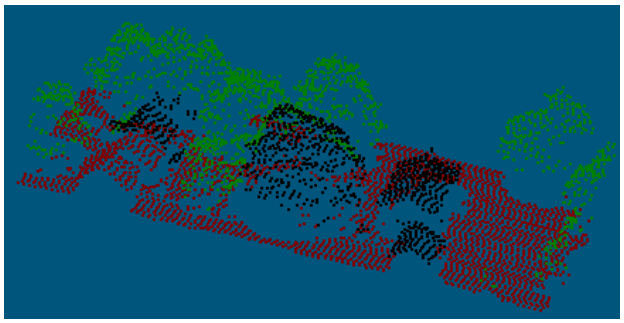
(c) BAYES-PW



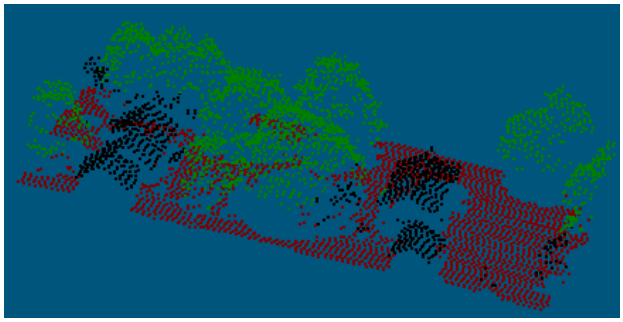
(d) BAYES



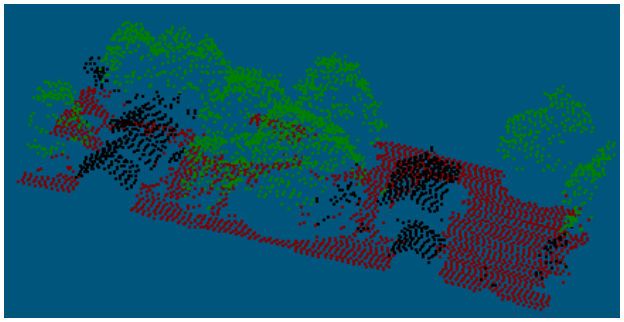
(e) FUNC-PW



(f) FUNC



(g) SVM-PW



(h) SVM

Рисунок 3.3: Результаты на части тестового облака точек из набора *Аэро*, на котором предложенный метод показывает высокую точность. Красный цвет соответствует категории *земля*, чёрный — *здание*, зелёный — *растительность*. (a) Верная (ручная) разметка. (b) Ансамбль рандомизированных деревьев, или только унарные потенциалы. (c)–(d) Наивный Байес, без унарных потенциалов и с ними. (e)–(f) Функциональный градиентный бустинг. (g)–(h) Метод секущей плоскости

3.5.4 Обсуждение

Эксперимент на наборе *Аэро* показывает, что неассоциативные марковские сети сегментируют облако точек точнее, чем ассоциативные, особенно при отсутствии унарных потенци-

алов. Даже простое наивное Байесовское обучение парных потенциалов показывает лучший результат, чем функциональный градиентный бустинг, стеснённый требованием ассоциативности. Ассоциативность может служить в качестве регуляризации, и таким образом лучше обучаться на небольших выборках с недостающей статистикой парных потенциалов. В этом эксперименте для обучения использовался один небольшой скан (он содержит около 100 тысяч точек, или 2 тысячи суперпикселей), при этом удалось обучить модель парных потенциалов с зависимостями между любыми парами из 3 категорий. Таким образом, неассоциативная модель может эффективно настраиваться даже на небольшой выборке, если число категорий небольшое, и среди них нет слишком недопредставленных.

В используемой модели парные потенциалы связывают достаточно удалённые точки из-за использования суперпикселей. Из-за этого доля гетерогенных рёбер (таких, что инцидентные им вершины помечены различно) больше, чем в модели, построенной над отдельными точками, так как суперпиксели обычно объединяют точки, которые должны относиться к одной категории. Более того, при использовании суперпикселей признаки парных потенциалов более информативны: для двух соседних точек облака, полученного достаточно плотным сканированием, такие признаки, как ориентация и длина соединяющего их отрезка, бесполезны из-за шума, возникающего при сканировании. Ангелов и др. [5] не используют сэмплирование исходного скана. Согласно их экспериментам, использование признаков помимо константного не увеличивает точность сегментации (хотя это может быть вызвано использованием достаточно грубой ассоциативной модели).

Для вывода финальной разметки на этапе предсказания и для вывода, дополненного функцией потерь, на этапе обучения используется алгоритм TRW-S. Несмотря на то, что он находит лишь приближённый максимум MAP-оценки, на практике обучается довольно точная модель. Использование приближённого вывода означает, что используемый вариант метода секущей плоскости выполняет *оптимизацию на расширенном множестве* (англ. *undergenerating*) [76], то есть, на каждой итерации находится наиболее нарушаемое ограничение среди подмножества множества линейных ограничений (1.64), таким образом, в рабочий набор добавляется действительное ограничение задачи, но, возможно, не самое нарушаемое. Альтернативой является выполнение *оптимизации на суженном множестве* (англ. *overgenerating*), в которой наоборот — множество доступных ограничений расширяется, но на таком расширенном множестве возможно искать точный оптимум достаточно эффективно. Этого можно добиться использованием LP-релаксации (раздел 1.2.1) или оптимизации двойственной задачи (раздел 1.2.3) в задачах вывода, дополненного функцией потерь. Таким образом, ограничения будут соответствовать не целочисленным разметкам, а также и дробным. При этом сохраняются теоретические свойства метода секущей плоскости [76].

Неточность недопорождающего подхода заключается в том, что на последней итерации может быть получено не самое нарушаемое ограничение, и оно будет удовлетворяться, таким образом, оптимизация остановится раньше реальной сходимости. Эта ошибка может быть ограничена сверху интервалом двойственности в TRW-S, который на практике обычно близок к нулю на последних итерациях метода секущей плоскости. Таким образом, в приведённых

экспериментах приближённый вывод не мог сильно повлиять на точность модели, однако его использование замедляет оптимизацию из-за большего числа генерируемых ограничений.

3.6 Выводы

В настоящей главе описана структура неассоциативной марковской сети и соответствующие алгоритмы для вывода разметки и обучения потенциалов. Приведена новая функция потерь, а также формулировка гауссовского ядра для неявного нелинейного преобразования признакового пространства. Эксперименты по семантической сегментации на двух наборах данных, представляющих собой облака точек в трёхмерном пространстве, показали, что все эти три модификации модели ведут к улучшению результата сегментации.

Глава 4

Использование пространственного контекста при последовательной классификации

В главе 1 описана методология решения задач разметки с помощью графических вероятностных моделей. Для решения задачи производится вывод наиболее вероятной разметки в соответствии с некоторым вероятностным распределением. Само же распределение, как правило, моделируется с помощью методов статистического машинного обучения, т. е. минимизируется некоторая функция эмпирического риска. Недостатком этого подхода является большая вычислительная сложность оптимизационных задач как вывода, так и минимизации риска: на практике их приходится решать приближённо, в связи с чем теоретические гарантии теряются. Кроме того, графические модели, используемые на практике, моделируют только локальные зависимости в данных: добавление дальнодействующих зависимостей сильно усложняет модель, так что реализовать достаточно эффективный вывод удаётся только для потенциалов специального вида [85].

В этой главе рассматривается альтернативный подход к решению задач разметки на основе *последовательной классификации*. В отличие от графических моделей, решающее правило сводится к последовательному применению классификаторов, которые обучаются жадным образом. Классификаторы каждого следующего слоя зависят от выходов классификаторов предыдущего слоя, таким образом, разметка постепенно уточняется с учётом обновлённой предварительной разметки. Достоинством данного метода является концептуальная простота, как следствие, более высокая скорость вывода, чем в графических моделях. Гибкость метода позволяет моделировать произвольные зависимости между метками, не только локальные, как в графических моделях.

Идея использования последовательной классификации для семантической сегментации не является новой. В алгоритме «автоконтекст» [25] решение о разметке принимается последовательным применением нескольких линейных классификаторов, обучаемых логистической регрессией. Авторы показали, что при добавлении нового слоя ошибка на обучающей выборке уменьшается, следовательно, алгоритм сходится по ошибке на обучении. Росс и др. [26]

предложили *машину вывода* — модификацию метода последовательной классификации, являющуюся обобщением алгоритма распространения доверия (раздел 1.2.2) для вывода в графических моделях. Мы будем пользоваться этим обобщением. Более полный обзор связанных методов приведён ниже в разделе 4.4.

В этой главе описано обобщение машины вывода [26], учитывающее пространственный контекст. Конкретно, предлагаемый метод отличается в следующем:

- функции пересчёта меток вычисляются в два этапа: сначала применяется классификатор, допускающий вероятностный выход, который классифицирует локальные признаки, дополненные гипотезой о разметке с предыдущей итерации; затем применяется линейное преобразование для агрегирования выходов классификаторов в гипотезы о метках этой итерации;

- переопределено понятие фактора, генерирующего сообщение в вершину: в этой главе фактор — это упорядоченная пара множеств вершин, называемых *передатчиком* и *приёмником*. Передатчик (группа переменных) влияет на текущее назначение метки приёмника (одна переменная) — этот процесс назовём передачей сообщения;

- в предлагаемой модели используются априорные знания о видах контекстуальных зависимостей с помощью *типов факторов*. Для каждого типа факторов используется своя функция передачи сообщений, что приводит к упрощению зависимостей, моделируемых этими отдельными функциями. Типы факторов могут задаваться таким образом, чтобы моделировать дальнодействующие зависимости (учитывающие контекст в сцене), которые могут быть анизотропными по своей природе, в отличие от короткодействующих зависимостей. Далее показано, как задавать типы факторов для учёта пространственного контекста при семантической сегментации трёхмерных облаков точек.

Метод экспериментально проверен на базе трёхмерных облаков точек, полученных Коппулой и др. [86] путём склеивания карт глубины, выдаваемых датчиком Kinect. Результаты показывают, что предлагаемый метод работает быстрее и качественнее, чем метод, основанный на использовании графических моделей [86].

4.1 Машина вывода

В данном разделе описывается машина вывода — метод структурного обучения, предложенный Россом и др. [26]. В отличие от стандартных методов структурного обучения, он не подбирает параметры графической модели, которая потом используется для вывода, а обучает непосредственно параметры алгоритма вывода. Метод мотивирован алгоритмом передачи сообщений на графе с циклами (раздел 1.2.2), который итеративно обновляет сообщения в зависимости от их предыдущих значений. Чтобы получить явную рекуррентную формулу для

пересчёта сообщения от переменной v в фактор f на итерации n, подставим (1.14) в (1.15):

$$\mu^n_{v\to f}(y_v) = \prod_{f':\ v\in\mathcal{C}_{f'}, f'\neq f} \left[\sum_{\mathbf{y}'_{f'}: y'_v = y} \Phi_{f'}(\mathbf{y}'_{f'}; \mathbf{x}_{f'}, \mathbf{w}) \prod_{v'\in\mathcal{C}_{f'}\setminus\{v\}} \mu^{n-1}_{v'\to f'}(y'_{v'}) \right]. \tag{4.1}$$

Первое произведение берётся по всем факторам f', в множество переменных которых входит v, за исключением фактора-адресата сообщения f. Каждый из множителей представляет собой сумму по всем разметкам переменных фактора f', согласованным с меткой y_v, а элементы суммы являются произведениями потенциала фактора f' на произведение сообщений, входящих в f', от переменных, принадлежащих этому фактору, за исключением переменной v.

Зависимость результата передачи сообщений от настраиваемых параметров $\mathbf{w}$ выражается через потенциальные функции $\Phi_{f'}$. Метод машины вывода предлагает отказаться от потенциальных функций, вместо этого настраивать зависимость значений сообщений от их предыдущих значений в более общем виде:

$$\boldsymbol{\mu}^n_{v\to f} = \mathbf{g}_n\Bigg(\bigoplus_{\substack{f':\ v\in\mathcal{C}_{f'},\\ f'\neq f}} \mathbf{x}_{f'}, \bigoplus_{\substack{v', f':\ v'\in\mathcal{C}_{f'},\\ v\in\mathcal{C}_{f'}, f'\neq f}} \boldsymbol{\mu}^{n-1}_{v'\to f'} \Bigg). \tag{4.2}$$

Здесь $\bigoplus$ — операция, зависящая от предметной области, например, усреднение или конкатенация векторов признаков, а $\mathbf{g}_n$ — некоторая функция-предиктор сообщения, подбираемая на этапе обучения (см. ниже). Обозначим её первый аргумент $\tilde{\mathbf{x}}_{v,f}$, а второй аргумент — $\tilde{\boldsymbol{\mu}}^{n-1}_{v,f}$, тогда (4.2) можно переписать как $\boldsymbol{\mu}^n_{v\to f} = \mathbf{g}_n\left(\tilde{\mathbf{x}}_{v,f}, \tilde{\boldsymbol{\mu}}^{n-1}_{v,f}\right)$, а оба этих аргумента вместе будем называть *расширенными признаками* предиктора. При этом первая часть вектора расширенных признаков $\tilde{\mathbf{x}}_{v,f}$ постоянна, а вторая часть $\tilde{\boldsymbol{\mu}}^{n-1}_{v,f}$ — пересчитывается на каждой итерации.

Как и в методе распространения убеждений, маргинальные распределения на последней итерации N оцениваются аналогично сообщениям из вершины, только используются признаки и сообщения во *все* соседние факторы:

$$\mathbf{b}^N_v = \mathbf{g}_N\Bigg(\bigoplus_{\substack{f':\\ v\in\mathcal{C}_{f'}}} \mathbf{x}_{f'}, \bigoplus_{\substack{v',\ f':\\ v'\in\mathcal{C}_{f'},\ v\in\mathcal{C}_{f'}}} \boldsymbol{\mu}^{N-1}_{v'\to f'} \Bigg). \tag{4.3}$$

Как и вывод, обучение происходит итерационно (см. алгоритм 4.1). На итерации n функция g_n настраивается в виде некоторого вероятностного классификатора с помощью алгоритма машинного обучения, например, логистической регрессии или ансамбля рандомизированных решающих деревьев (строка 13). Для каждой из пар $(v, f) : v \in \mathcal{C}_f$ обучающей выборки в вектор расширенных признаков включаются признаки $\mathbf{x}_{f'}$ и сообщения $\boldsymbol{\mu}^{n-1}_{v'\to f'}$ (если $n \neq 1$), где f' — все факторы, включающие v, за исключением f (кроме последней итерации), а v' — их переменные. В качестве целевых переменных $\boldsymbol{\mu}^n_{v\to f}$ ($\mathbf{b}^N_v$ на последней итерации) берутся ответы обучающей выборки y_v (точнее, их переопределённые представления $\boldsymbol{\Upsilon}_v$: $\Upsilon_{v,k} = [\![y_v = k]\!], \forall k \in \mathcal{K}$). Поскольку на итерации n используется выход классификаторов с итерации $n-1$, модель может получиться смещённой. Чтобы этого избежать, на

Алгоритм 4.1 Обучение машины вывода

1: **Вход:** размеченная выборка $(\mathbf{x}, \mathbf{y})$, множество факторов обучающей выборки $\mathcal{F}$, разделённое на части $\mathbf{f}$, число итераций вывода N.
2: **Выход:** набор функций-предикторов сообщений $\{g_n(\cdot)\}_{n\in\{1,\dots,N\}}$
3: инициализировать $\boldsymbol{\mu}^0_{v\to f} = \frac{1}{K},\ \forall v \in \mathcal{V}, \forall f : v \in \mathcal{C}_f$
4: **for** $n = 1$ to $N-1$ **do**
5: **for all** $\mathbf{f} \in \mathcal{F}$ **do**
6: обучить вспомогательный предиктор $\mathbf{g}_{tmp}(\cdot)$ так чтобы $\boldsymbol{\Upsilon}_v \approx \mathbf{g}_{tmp}\left(\tilde{\mathbf{x}}_{v,f}, \tilde{\boldsymbol{\mu}}^{n-1}_{v,f}\right)$
на выборке, соответствующей парам $\left\{(v,f) : f \in \bigcup_{\mathbf{f}'\in\mathcal{F}\setminus\{\mathbf{f}\}} \mathbf{f}', v \in \mathcal{C}_f\right\}$
7: **for all** $f \in \mathbf{f}$ **do**
8: **for all** $v \in \mathcal{C}_f$ **do**
9: $\boldsymbol{\mu}^n_{v\to f} \leftarrow \mathbf{g}_{tmp}\left(\tilde{\mathbf{x}}_{v,f}, \tilde{\boldsymbol{\mu}}^{n-1}_{v,f}\right)$ # несмещённые оценки ответов на итерации n
10: **end for**
11: **end for**
12: **end for**
13: обучить окончательный предиктор $\mathbf{g}_n(\cdot)$ так чтобы $\boldsymbol{\Upsilon}_v \approx \mathbf{g}_n\left(\tilde{\mathbf{x}}_{v,f}, \tilde{\boldsymbol{\mu}}^{n-1}_{v,f}\right)$
на выборке, соответствующей парам $\left\{(v,f) : f \in \bigcup_{\mathbf{f}'\in\mathcal{F}} \mathbf{f}', v \in \mathcal{C}_f\right\}$
14: **end for**
15: обучить предиктор маргиналов $\mathbf{g}_N(\cdot)$, так чтобы $\boldsymbol{\Upsilon}_v \approx \mathbf{g}_N\left(\bigoplus_{\substack{f':\\ v\in\mathcal{C}_{f'}}} \mathbf{x}_{f'},\ \bigoplus_{\substack{v',\ f':\\ v'\in\mathcal{C}_{f'},\ v\in\mathcal{C}_{f'}}} \boldsymbol{\mu}^{N-1}_{v'\to f'}\right)$
на выборке, соответствующей парам $\left\{(v,f) : f \in \bigcup_{\mathbf{f}'\in\mathcal{F}} \mathbf{f}', v \in \mathcal{C}_f\right\}$

каждой итерации вычисляются несмещённые оценки сообщений по отложенной части выборки (строки 5–12). Для этого множество всех факторов делится на части $\mathbf{f}$, и несмещённые оценки значений сообщений для конкретной из частей получаются с помощью вспомогательного классификатора, обученного по объединению факторов всех остальных частей обучающей выборки (строка 6).

4.2 Пространственная машина вывода

В этом разделе описана *пространственная* модификация машины вывода, которая позволяет учитывать априорные знания о структуре задачи. Основным инструментом для этого является *д-фактор*, который может относиться к одному из *типов факторов*.

4.2.1 Описание модели и вывода в ней

Определение 4.1. *Д-фактором* называется пара $p = (d_f, \mathcal{S}_f)$, состоящая из приёмника — переменной $d_f \in \mathcal{V}$ и передатчика — множества переменных $\mathcal{S}_f \subset \mathcal{V}$.

Д-факторы в явном виде определяют, какие переменные и признаки используются при прогнозировании значения каждой из переменных d_f, вместо неявного определения множества-передатчика на основе структуры графической модели, как делается в машине вывода.

Определение 4.2. *Функция-предиктор* сообщения $\mathbf{g}^n_{t(f)}(\cdot)$ на n-й итерации для типа факторов $t(f)$ (см. ниже) имеет следующий вид:

$$\boldsymbol{\mu}^n_{\mathcal{S}_f \to d_f} = \mathbf{g}^n_{t(f)}\big(\mathbf{b}^{n-1}_{d_f}, \mathbf{x}_{d_f}, \mathbf{x}_f, \mathbf{x}_{\mathcal{S}_f}, \frac{1}{|\mathcal{S}_f|}\sum_{v \in \mathcal{S}_f} \mathbf{b}^{n-1}_v\big). \tag{4.4}$$

Она подбирается в семействе ансамблей решающих деревьев с помощью алгоритма random forest [87], однако может использоваться любой другой классификатор, допускающий вероятностный выход. В качестве аргументов предиктора используются признаки д-фактора $\mathbf{x}_f$ и усреднённые убеждения о метках в множестве-передатчике $\mathbf{b}^{n-1}_v,\ v \in \mathcal{S}_f$, а также, в отличие от классической машины вывода, используются признаки приёмника $\mathbf{x}_{d_f}$ и убеждения о его метке с предыдущей итерации $\mathbf{b}^{n-1}_{d_f}$. Первые позволяют получить качественную классификацию по локальным признакам уже на первой итерации, а последние позволяют получить тождественную функцию (по отношению к убеждениям приёмника) на последних итерациях, когда остальные параметры малоинформативны. Кроме того, аргумент может включать некоторые признаки передатчика $\mathbf{x}_{\mathcal{S}_f}$ (например, если передатчик характеризуется регионом пространства, это могут быть признаки облака точек в данном регионе), но, как показывают эксперименты, они малоинформативны. Конкатенацию всех аргументов функции-предиктора будем снова называть *расширенными признаками*.

Переменной v могут соответствовать несколько д-факторов, имеющих эту переменную приёмником, в этом случае вероятностные выходы предикторов необходимо агрегировать. На n-й итерации нормированный вектор убеждений относительно метки y_v определяется как взвешенное произведение сообщений д-факторов из $\mathcal{S}_f$ в v:

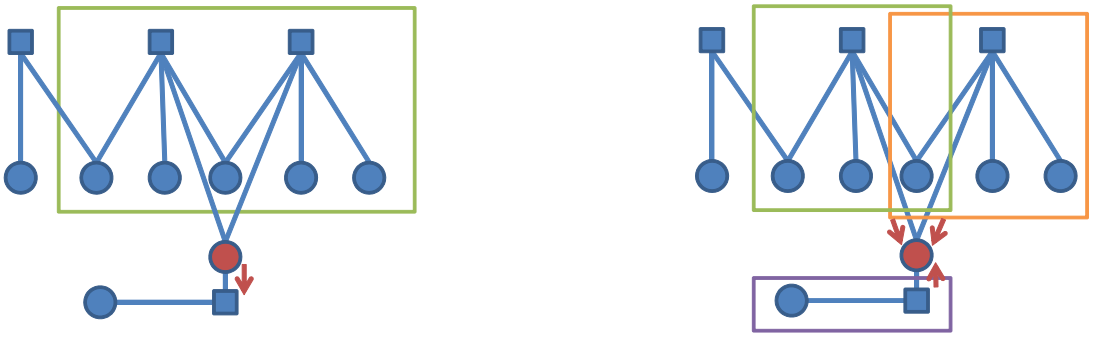
$$b^n_v(y) \propto \prod_{f: d_f = v} \Big(\mu^n_{\mathcal{S}_f \to v}(y)\Big)^{\alpha^n_{t(f)}}, \quad \forall y \in \{1, \ldots, K\}, \tag{4.5}$$

где $\alpha^n_{t(f)}$ — параметр, соответствующий вкладу типа факторов $t(f)$ (см. ниже).

Определение 4.3. *Типом фактора* $t(f) \in \mathcal{T}$ называется признак, заданный для каждого из д-факторов и определяющий конкретную функцию-предиктор сообщений $\mathbf{g}^n_{t(f)}(\cdot)$ и коэффициент $\alpha^n_{t(f)}$, которые используются для предсказания убеждения на итерации n.

Например, некоторые д-факторы моделируют ближние зависимости, которые, как правило, служат для сглаживания разметки, а другие моделируют пространственные зависимости разнообразного характера, анизотропные по своей природе. При использовании типов факторов моделирование зависимостей не возлагается на один предиктор; вместо этого для каждого типа факторов настраивается своя функция-предиктор, что делает их проще и уменьшает эффект переобучения.

Если предположить, что отдельные сообщения в (4.5) независимы, то по закону произведения вероятностей убеждения о метках переменной можно получить, просто перемножив соответствующие сообщения, однако используется более гибкая модель. Коэффициенты $\alpha^n_{t(f)}$



(a) Классическая машина вывода [26]

(b) Предложенный метод

Рисунок 4.1: Различные способы применить последовательную классификацию. Переменные показаны кругами, факторы — квадратами. Чтобы вычислить сообщение, которое переменная (красный круг) пересылает в фактор, классическая машина вывода [26] (a) использует все сообщения с предыдущей итерации, которые были посланы из всех переменных, имеющих общий фактор с данной, кроме неё самой. Сообщения из этой вершины в два других фактора, инцидентных ей, вычисляются аналогично. В пространственной машине вывода (b), все три инцидентных д-фактора (передатчиками которых являются множества переменных в рамках соответствующего цвета) «посылают сообщения» в приёмник (красный круг), которые агрегируются, чтобы получить убеждение о метке данной переменной. В каждом из случаев, рамка соответствует обучаемой функции вычисления сообщения. Видно, что в (b) множества аргументов функций меньше по размеру.

подбираются так, чтобы исключить вклад малоинформативных типов факторов, таким образом накладывают регуляризацию на модель (подробнее о настройке параметров в разделе 4.2.3).

Последний аргумент функции (4.4) — усреднённые убеждения о метках переменных в передатчике с предыдущей итерации. Такое усреднение ведёт к потере информации о пространственном расположении объектов, соответствующих переменным в передатчике. Предполагается, что передатчик состоит из «пространственно близких» переменных (например, соответствующих пикселям из некоторого прямоугольника на изображении). Эти множества должны быть достаточно большими, чтобы избежать переобучения (много маленьких передатчиков позволят настроиться на шум, в то время как усреднение в передатчике повышает робастность), при этом достаточно маленькими, чтобы не потерять важные зависимости.

В отличие от (4.2), функции-предикторы сообщений $\boldsymbol{\mu}^n_{\mathcal{S}_f \to v}$ в предлагаемом методе зависят от сообщений с предыдущей итерации $\boldsymbol{\mu}^{n-1}_{\mathcal{S}_{f'} \to v'}$ не напрямую. Вместо этого, аргументом являются их взвешенные произведения — убеждения о метках $\mathbf{b}^{n-1}_{v'}$. Кроме этого, «охват» аргументов обучаемой функции в предлагаемом методе меньше: они принимают информацию о переменных (убеждения, признаки), вовлечённых в один фактор, в то время как в методе Росса и др. [26] конкатенируются сообщения из всех переменных, которые имеют общий фактор с переменной v, за исключением целевого фактора f. Рис. 4.1 иллюстрирует разницу. Предлагаемый метод комбинирует полученные «локализованные» сообщения в явном виде, согласно (4.5), таким образом получая параметры предиктора сообщений следующей итерации или финальные оценки маргинальных распределений меток.

4.2.2 Пространственные и структурные д-факторы

В данном разделе мы будем предполагать, что модель определяется для задач распознавания в некоторой двумерной или трёхмерной визуальной сцене, состоящей из *элементов* — пикселей, вокселей, точек или суперпикселей, соответствующих переменным в задаче разметки. Элемент v характеризуется координатами $\mathbf{p}_v = (x, y)$ в двумерном пространстве или $\mathbf{p}_v = (x, y, z)$ в трёхмерном. Мы определим два семейства типов факторов и опишем область их применимости.

Определение 4.4. *Пространственные* д-факторы — семейство типов факторов, моделирующих пространственное взаиморасположение элементов сцены. Тип факторов t однозначно задаётся регионом координатного пространства $\mathcal{P}_t$. Для элемента пространства, соответствующего переменной v с координатами $\mathbf{p}_v$, порождается д-фактор $(v, \mathcal{S})$, где в $\mathcal{S}$ входят переменные, соответствующие всем элементам из региона $\mathcal{P}_t + \mathbf{p}_v = \{\mathbf{p}_t + \mathbf{p}_v \mid \mathbf{p}_t \in \mathcal{P}_t\}$.

Пространственный тип факторов может быть параметризован парой отступа и радиуса $(\delta\mathbf{p}, r)$: регион будет задаваться так, что в него попадут все элементы на расстоянии не более r по некоторой метрике (например, порождённой L_1 или L_2 мерой) от точки, полученной смещением элемента на $\delta\mathbf{p}$, а именно $\mathcal{P}_t = \{\mathbf{p} \mid \|\mathbf{p} - \delta\mathbf{p}\| \leq r\}$. Такая формулировка позволяет с помощью пространственных типов факторов моделировать произвольные дальнодействующие контекстуальные зависимости между переменными.

Определение 4.5. *Структурные* д-факторы — тип факторов, моделирующих локальные зависимости. Для элемента сцены, соответствующего переменной v с координатами $\mathbf{p}_v$, порождаются д-факторы $(v, \{u\})$ для каждой из переменных u, таких что $\mathbf{p}_u$ принадлежит некоторой окрестности $\mathbf{p}_v$ (либо r-окрестности по какой-либо мере, либо входит в k ближайших соседей).

Структурные д-факторы мотивированы типичными парными связями в графических моделях: при их использовании можно сохранить структуру локальных взаимодействий, тем самым в некотором смысле они позволяют обобщить графическую модель в рамках последовательной классификации. В отличие от пространственных типов факторов, структурный тип факторов не соответствует конкретному геометрическому смещению. Всем структурным д-факторам соответствуют одинаковые предикторы $\mathbf{g}_t^n$ и весовые коэффициенты α_t^n.

Пространственный тип факторов может характеризоваться несколькими парами $(\delta\mathbf{p}, r)$, если предполагается, что природа зависимостей для этих смещений одинакова, следовательно, они могут моделироваться одинаковыми функциями-предикторами. Например, д-факторы, отвечающие смещениям влево и вправо на рис. 4.2, отнесены к одному типу факторов, так как зависимости в реальных сценах обычно инвариантны к отражению относительно вертикальной оси. Для порождения системы пространственных факторов использована следующая схема: определяется шаблон, состоящий из нескольких параметризаций д-факторов и соответствующих им типов: $\left[\left((\delta\mathbf{p}_i, r_i), t_i\right)\right]_{i=1}^{I}$, где $t_i \in \mathcal{T}$ — один из возможных типов факторов. Далее, этот шаблон применяется ко всем элементам сцены (пикселям, точкам или супер-

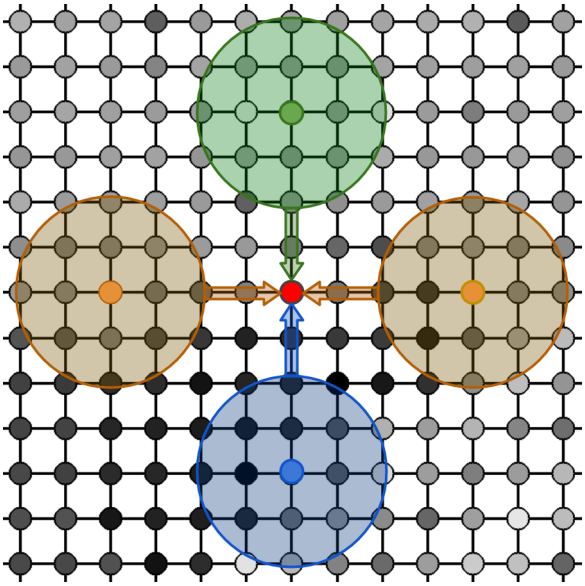


Рисунок 4.2: Иллюстрация определения структурных и пространственных д-факторов для фрагмента изображения с рис. 1.2a. Переменные модели соответствуют пикселям изображения. Чёрные линии обозначают структурные взаимосвязи между переменными. Также показаны четыре пространственных д-фактора трёх типов, приёмником которых является переменная, показанная красным кругом. Пусть координатные оси направлены вправо и вверх. Регион-передатчик д-фактора типа ‘Вверх’ ($\delta\mathbf{p} = (0, +4), r = 2$) показан зелёным, регион-передатчик д-фактора типа ‘Вниз’ ($\delta\mathbf{p} = (0, -4), r = 2$) — синим, д-факторов типа ‘Вправо/влево’ ($\delta\mathbf{p} = (\pm 4, 0)$, $r = 2$) — оранжевым.

пикселям), чтобы породить конкретные д-факторы. Таким образом, в сцене с V элементами будет $V \cdot I$ д-факторов (с поправкой на граничные эффекты).

Теоретически, подобные дальнодействующие зависимости можно включить и в структуру марковской сети, однако при этом возникают технические проблемы: вывод в модели становится вычислительно трудным, что особенно проблематично при настройке параметров, где его нужно вызывать итеративно (подробнее об обучении марковских сетей см. раздел 1.3). Что более важно, в парно-сепарабельной модели учитывается зависимость лишь от одного элемента, без пространственного усреднения, что делает метод чувствительным к высокочастотному шуму. Если же моделировать это усреднение с помощью графической модели, в ней возникают факторы высоких порядков, при этом на каждый пространственный д-фактор f, передатчик которого содержит $|S_f|$ вершин, необходимо создать $(|S_f|+1)$ фактор графической модели порядка $(|S_f| + 1)$. Учитывая характерные размеры передатчиков факторов (десятки переменных), можно сделать вывод о невозможности использования такой модели в реальных задачах.

Описанная схема предполагает, что переменной-приёмнику соответствует одна точка пространства. Однако на практике элементами модели часто являются суперпиксели. В таком случае можно считать каждый суперпиксель «материальной точкой», приближая его положение центром масс, либо объединять регионы-передатчики для всех точек, соответствующих приёмнику (на рис. 4.3c показан пример региона-передатчика для суперпикселя в трёхмерном пространстве). Первый способ вычислительно эффективнее, однако в наших экспериментах

он показывал немного меньшую точность. Объединение регионов требует дополнительных вычислений, однако они могут проводиться на этапе предобработки, так как границы регионов не зависят от конкретных меток, так что в некоторых приложениях этими дополнительными затратами можно пренебречь.

4.2.3 Обучение модели

Обучение в предложенной модели состоит из восстановления функций (4.4) и, при необходимости, настройки коэффициентов $\boldsymbol{\alpha}$ в (4.5). При этом необходимо избежать систематического смещения результатов полученной модели, а также переобучения.

Среди аргументов функций-предикторов сообщений присутствуют убеждения с предыдущей итерации, поэтому для их обучения необходимо получить несмещённую оценку убеждений, предсказываемых на этапе вывода. Это означает, что нельзя использовать один и тот же набор данных для оценки убеждений на предыдущей итерации и для настройки параметров предиктора. В этом случае прогноз получился бы смещённым, так как точность предиктора на обучающей выборке в общем случае выше, чем на контрольной. Поскольку получаемые убеждения были бы ближе к верным меткам, обученные по ним предикторы были бы смещёнными в сторону сохранения поданного им на вход убеждения. Вместо этого на каждой итерации для получения несмещённой оценки убеждений используется кросс-валидация по k частям обучающей выборки. Для каждой из частей, вспомогательные предикторы (4.4) для каждого из типов факторов обучаются на всех остальных частях выборки, затем применяются к данной части для оценки убеждений. Для обучения основных предикторов на данной итерации, используется вся доступная выборка (в том числе полученные оценки убеждений для неё). Алгоритм 4.2 более строго описывает процесс обучения: строки 5–12 служат для обучения вспомогательных предикторов данной итерации и получения с их помощью несмещённых оценок сообщений, а строки 13–15 служат для обучения основных предикторов, являющихся выходными данными алгоритма. Несмещённые оценки сообщений могут быть использованы для настройки весов $\boldsymbol{\alpha}^n$ (строка 16), а также необходимы для оценки убеждений, являющихся входными данными следующей итерации (строки 17–21).

Использование большого числа типов факторов может привести к переобучению предикторов, и, как следствие, к ухудшению качества предсказания модели на генеральной совокупности. Для предотвращения этого используются веса α_t^n (4.5), которые регулируют вклад различных факторов в убеждения в зависимости от их типов. Параметры $\boldsymbol{\alpha}^n$ настраиваются на каждой итерации n с помощью максимизации регуляризованной суммы убеждений, взятых на корректных ответах из обучающей выборки y_v:

$$\boldsymbol{\alpha}^n = \underset{\boldsymbol{\alpha}\geq 0}{\operatorname{argmax}} \sum_{v\in\mathcal{V}} \left(\frac{\prod_{f:d_f=v} (\mu^n_{\mathcal{S}_f\to v}(y_v))^{\alpha_{t(f)}}}{\sum_{k\in\mathcal{K}} \prod_{f:d_f=v} (\mu^n_{\mathcal{S}_f\to v}(k))^{\alpha_{t(f)}}} + C \sum_{t\in\mathcal{T}} \alpha_t \right). \tag{4.6}$$

Использование L_1-регуляризации на веса позволяет подавить влияние неинформативных типов факторов, в результате чего получается разреженный вектор весов. В частности, если в

Алгоритм 4.2 Обучение пространственной машины вывода

Вход: размеченная выборка $(\mathbf{x}, \mathbf{y})$, множество д-факторов обучающей выборки $\mathcal{F}$, разделённое на части $\mathbf{f}$, множество типов факторов $\mathcal{T}$, число итераций вывода N.
Выход: набор пар функций-предикторов сообщений и весов $\{(g_{n,t}(\cdot), \alpha_t^n)\}_{t\in\mathcal{T},\ n\in\{1,\dots,N\}}$
инициализировать $\mathbf{b}_v^0 = \frac{1}{K},\ \forall v \in \mathcal{V}$
for $n = 1$ to N **do**
  for all $\mathbf{f} \in \mathcal{F}$ **do**
    for all $t \in \mathcal{T}$ **do**
      обучить предиктор $\mathbf{g}_t^{tmp}(\cdot)$ так, чтобы $\boldsymbol{\Upsilon}_{d_f} \approx \mathbf{g}_t^{tmp}(\langle \textit{расширенные признаки } f \rangle)$
        на выборке д-факторов $\left\{ f \in \bigcup_{\mathbf{f}'\in\mathcal{F}\setminus\{\mathbf{f}\}} \mathbf{f}' \;\middle|\; t(f) = t \right\}$
    end for
    for all $f \in \mathbf{f}$ **do**
      $\boldsymbol{\mu}^{tmp}_{\mathcal{S}_f \to d_f} \leftarrow \mathbf{g}^{tmp}_{t(f)}\left(\mathbf{b}^{n-1}_{d_f}, \mathbf{x}_{d_f}, \mathbf{x}_f, \mathbf{x}_{\mathcal{S}_f}, \frac{1}{|\mathcal{S}_f|}\sum_{v\in\mathcal{S}_f} \mathbf{b}_v^{n-1}\right)$
    end for
  end for
  for all $t \in \mathcal{T}$ **do**
    обучить предиктор $\mathbf{g}_t^{n}(\cdot)$ так, чтобы $\boldsymbol{\Upsilon}_{d_f} \approx \mathbf{g}_t^{n}(\langle \textit{расширенные признаки } f \rangle)$
      на выборке д-факторов $\left\{ f \in \bigcup_{\mathbf{f}'\in\mathcal{F}} \mathbf{f}' \;\middle|\; t(f) = t \right\}$
  end for
  задать веса типов факторов $\boldsymbol{\alpha}^n$, например $\boldsymbol{\alpha}^n = \mathbf{1}$ или максимизируя (4.6)
  if $n < N$ **then**
    for all $v \in \mathcal{V}$ **do**
      вычислить убеждения $\mathbf{b}_v^n$ по сообщениям $\boldsymbol{\mu}^{tmp}_{\mathcal{S}_f\to v}$ согласно (4.5)
    end for
  end if
end for

данных отсутствуют дальнодействующие зависимости, веса соответствующих типов факторов будут нулевыми благодаря такой регуляризации (см. раздел 4.5). C — параметр, задающий силу регуляризации. Максимизация может выполняться, например, с помощью квазиньютоновского метода.

4.3 Детали реализации

Этот раздел демонстрирует, как описанная модель может использоваться в задаче сегментации трёхмерных облаков точек (рис. 4.4a). Для экспериментов использовались данные, полученные сшивкой сканов комнатных сцен, полученных датчиком Kinect Коппулой и др. [86]. Предполагается, что описанная в этой главе реализация будет применяться к данным похожей природы, с аналогичным процессом предобработки.

4.3.1 Структура модели

Коппула и др. [86] используют пересегментацию облака точек, основанную на выделении плоских сегментов. Это позволяет добиться того, что суперпиксели будут гомогенными, то есть будут точки внутри каждого суперпикселя будут относиться к одной категории. Также,

Таблица 4.1: Типы факторов, используемые в модели для сегментации трёхмерных облаков точек. Строки содержат названия типов факторов их обозначения, а также относительные координаты регионов передатчика.

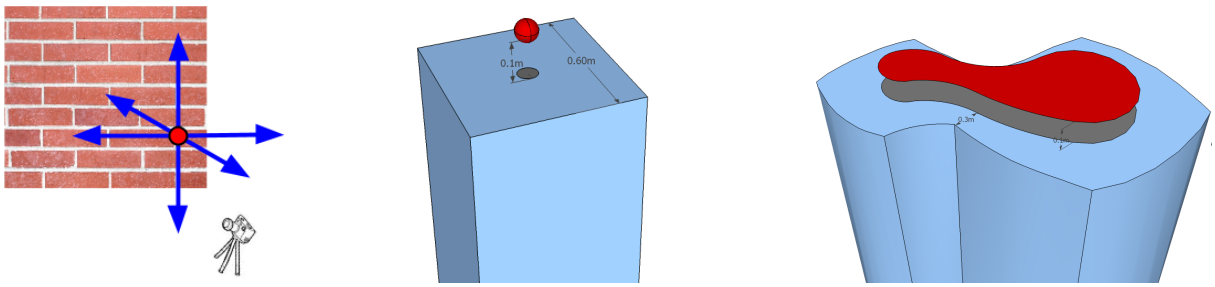
#	название	обозн.	регион $((x_0, y_0, z_0), (x_1, y_1, z_1))$, м
0	Структурный	S	не применимо
1	Локальный	Lo	$((-0.1, -0.1, -0.1), (0.1, 0.1, 0.1))$
2	К-вниз	Td	$((0.1, -0.3, -\infty), (\infty, 0.3, -0.1))$
3	От-вверх	Fu	$((-\infty, -0.3, 0.1), (-0.1, 0.3, \infty))$
4	Вниз	D	$((-0.3, -0.3, -\infty), (0.3, 0.3, -0.1))$
5	Лево	Lr	$((-0.3, -1.0, -0.3), (0.3, -0.1, 0.3))$
6	Право	Rr	$((-0.3, 0.1, -0.3), (0.3, 1.0, 0.3))$
7	От	F	$((-\infty, -0.3, -0.3), (-0.1, 0.3, 0.3))$
8	К	T	$((0.1, -0.3, -0.3), (\infty, 0.3, 0.3))$
9	Вверх	U	$((-0.3, -0.3, 0.1), (0.3, 0.3, \infty))$

учитывая соображения эффективности, было принято решение использовать суперпиксели в качестве элементарных объектов модели. Предложенная модель также сохраняет структуру взаимодействий, использованную Коппулой и др. [86]: любая пара суперпикселей v и u, кратчайшее расстояние между точками которых менее 0.6 м, порождает два *структурных* д-фактора: $(v, \{u\})$ и $(u, \{v\})$. Все структурные д-факторы в предлагаемой модели относятся к одному типу.

Чтобы определить параметризацию *пространственных* д-факторов, необходимо ввести набор координатных систем, ассоциированных с каждым из приёмников д-факторов. Сначала мы опишем, как такая координатная система определяется для отдельной точки, а потом обобщим определение на случай суперпикселей. Поскольку облака точек изображают комнатные сцены, можно определить такую систему координат так, чтобы не осталось степеней свободы. Вертикальное направление определено однозначно. Для каждой точки также известна позиция камеры, зафиксировавшей эту точку. Большинство объектов сцены находятся достаточно близко к стенам, так что появляется ещё одно направление: в сторону камере, горизонтально перпендикулярно ближайшей стене (рис. 4.3a). Третье направление определяется перпендикулярно первым двум (используется правая тройка координат).

Описанный алгоритм требует знания положения стен в сцене. Мы используем эвристический алгоритм поиска стен (робастное вписывание вертикальных плоскостей), который на практике способен найти почти все стены. Однако это не означает, что найденные точки в итоге будут отнесены к категории стен; результат детектирования используется только для определения ортогонального направления для установления систем координат. Если данные имеют другую природу, и нельзя установить неинвариантную систему координат, необходимо использовать другую параметризацию типов факторов: например, передатчик должен включать в себя не все точки внутри сферы, а все точки внутри тора — геометрическое место всех шаров при вариации азимута.

Таблица 4.1 содержит информацию о параметризации типов факторов, используемых в предлагаемой модели. В целях эффективности при определении передатчиков использует-



(a) Система координат (b) Регион-передатчик для точки (c) Регион-передатчик для суперпикселя

Рисунок 4.3: Определение пространственных типов факторов в трёхмерном пространстве. (a) Для каждой точки вводится система координат, ко-ориентированная с ближайшей стеной. (b)–(c) Регионы, которые используются для определения переменных передатчика, когда переменная-приёмник соответствует (b) индивидуальной точке или (c) суперпикселю для типа факторов ‘Вниз’. Красная сфера обозначает точку, а красный сегмент плоскости обозначает суперпиксель, соответствующие переменной-приёмнику. Статистики, формирующие векторы признаков передатчика ($\mathbf{x}_{\mathcal{S}_f}$ и $\langle \mathbf{b}_v^{n-1} \rangle$), определяются по точкам, попавшим в голубой регион.

ся расстояние по манхэттенской метрике, то есть регионы ограничены параллелепипедами, возможно открытыми (кроме того, задаётся индивидуальное расстояние для каждого из направлений). Параллелепипед в трёхмерном пространстве задаётся шестью числами, которые можно сгруппировать в координаты противолежащих вершин, где координатная система связана с точкой, как описано выше. Эти координаты указаны в строках таблицы 4.1 в метрах вместе с информацией о названиях типов факторов. Например, тип ‘Вниз’ (строка 4 таблицы) предполагает, что относящиеся к нему пространственные д-факторы включают в передатчик все точки ниже приёмника с отступом в 10 см в коридоре 60 см $\times$ 60 см, другими словами, координаты x и y варьируются в диапазоне $[-0.3, 0.3]$, а координата z — в диапазоне $(-\infty, -0.1]$ (рис. 4.3b).

Поскольку элементарным объектом в модели является суперпиксель, нужно определить параметризацию пространственных д-факторов, приёмником в которых является переменная, соответствующая суперпикселю. В этом случае регион-передатчик определяется как объединение всех регионов, которые были бы передатчиками при порождении д-фактора данного типа на каждой из точек суперпикселя в качестве приёмника. Например, для суперпикселя, отвечающего столешнице, д-фактор типа ‘Вниз’ включит в свой передатчик все точки ниже неё с зазором в 10 см по высоте, расширенную на 30 см по ширине и глубине (рис. 4.3c).

4.3.2 Обучение предикторов сообщений и их признаки

В качестве функций-предикторов сообщений (4.4) для всех итераций и всех типов факторов используются ансамбли из 100 решающих деревьев. Для их обучения используется алгоритм random forest [87]. Для определения функции разбиения в узле дерева тестируются d разбиений по случайно выбранному признаку, из них выбирается лучшее в соответствии

Таблица 4.2: Унарные и парные признаки, используемые Коппулой и др. [86]. В описании спектральных признаков λ_{vi} означает i-е по величине собственное значение матрицы ковариаций точек суперпикселя v, $i \in \{1, 2, 3\}$.

Унарные признаки суперпикселя v	разм-ть
Цветовые признаки, в т. ч.:	**48**
Гистограмма значений каналов по цветовому пространству HSV	14
Среднее значение каналов HSV	3
Среднее значение ячеек гистограммы ориентированных градиентов посчитанных по изображению суперпикселя	31
Признаки локальной формы и геометрии, в т. ч.:	**8**
Линейность $(\lambda_{v1} - \lambda_{v2})$, планарность $(\lambda_{i2} - \lambda_{i3})$, Разброс λ_{i1}	3
Вертикальная компонента нормали n_{vz}	1
Вертикальная позиция центра масс c_{vz}	1
Вертикальный и горизонтальный размах огибающего параллелепипеда	2
Расстояние до границы сцены	1
Парные признаки для $(u, \{v\})$	разм-ть
Цветовые признаки, в т. ч.:	**3**
Разница средних значений по каналам HSV	3
Признаки локальной формы и геометрии, в т. ч.:	**2**
Копланарность и выпуклость	2
Признаки геометрического контекста, в т. ч.:	**6**
Горизонтальная проекция расстояния между центрами масс	1
Вертикальная проекция расстояния между центрами масс $(c_{uz} - c_{vz})$	1
Скалярное произведение нормалей $\mathbf{n}_u \cdot \mathbf{n}_v$	1
Разница в отклонениях нормалей от вертикали $(\cos^{-1} n_{uz} - \cos^{-1} n_{vz})$	1
Кратчайшее расстояние между суперпикселями	1
Относительное расположение по отношению к позиции камеры (перед/позади)	1

с коэффициентом Джини. Значение d выбирается как квадратный корень из общего числа признаков.

Для проведения экспериментов используются те же признаки, что и у Коппулы и др. [86]. Унарные признаки описывают локальный вид суперпикселя, например, его планарность, ориентацию и гистограмму градиентов цветов. Парные признаки описывают взаимодействие между соседними суперпикселями, например, это может быть угол между нормалями или проекция расстояния между центрами масс на вертикальную ось. Таблица 4.2 содержит список используемых признаков суперпикселей и структурных факторов. Функции-предикторы сообщений (4.4) используют в качестве аргументов локальные признаки и убеждения с предыдущей итерации. Для пространственных типов факторов к ним добавляются средние убеждения передатчика (таким образом, зависимость от признаков фактора $\mathbf{x}_f$ и признаков передатчика $\mathbf{x}_{\mathcal{S}_f}$ фиктивная), всего размерность входа — $56 + 2K$, где K — число категорий в задаче. Для структурного типа факторов зависимость от признаков фактора $\mathbf{x}_f$ существенная, в качестве них используются парные признаки, так что размерность входа — $56 + 11 + 2K$.

4.4 Обзор литературы

Одним из первых использований последовательной классификации был теггер Брилла [12], служащий для разметки частей речи в предложении. После того как части речи каждого из слов определены с помощью локальной классификации, теггер применяет к этой первичной разметке последовательность нелокальных *корректировок*. Например, следующая корректировка оказывается эффективной для разметки частей речи в английских предложениях: «Если слово ‘to’ отмечено как *частица инфинитива*, и за ней следует слово, отмеченное как *артикль*, изменить метку последнего слова на *предлог*». Если корректировка не применяется к фразе (предпосылка не верна), разметка остаётся без изменений. Таким образом, метки часто остаются такими же, как на предыдущих итерациях. Аналогично, предложенный метод использует убеждения с предыдущей итерации в качестве одного из аргументов функции-предиктора сообщений, что позволяет возвращать тождественную функцию, не изменяющую разметку — это бывает полезно на поздних итерациях. На этапе обучения системы последовательность корректировок может быть определена жадным образом: на каждой итерации из пула выбирается та, которая сильнее всего уменьшает ошибку на обучающей выборке.

Эта идея также использовалась в компьютерном зрении. Алгоритм *«автоконтекст»* (англ. *auto-context*) [25] последовательно применяет настроенные классификаторы для уточнения разметки. Среди аргументов классификтора используется разметка с предыдущей итерации. Не все элементы разметки используются в качестве аргументов. Пользователь задаёт систему соседства: набор смещений (окрестность) относительно данного пикселя. Они являются аналогом предлагаемых пространственных типов факторов. В отличие от описанного выше метода, «автоконтекст» конкатенирует метки из окрестности, и использует один линейный классификатор. При его обучении на каждой итерации в качестве целевых переменных используется верная разметка обучающей выборки.

Позже авторы предложили использовать одну и ту же функцию на всех итерациях, таким образом, сформулировали задачу обучения как поиск сжимающего отображения, сходящегося к верной разметке обучающей выборки [35]. Они провели теоретический анализ и сформулировали условия, при которых логистическая регрессия является сжимающим отображением, а также предложили метод обучения произвольной модели, гарантирующий сходимость к неподвижной точке.

«Semantic texton forest» (*STF*) [3] — ещё одна модель, позволяющая учитывать контекстуальные зависимости между метками в явном виде с помощью двух стадий последовательной классификации. STF используется для категоризации и сегментации изображений. На первой стадии по локальным признакам пикселей оцениваются так называемые *семантические текстоны* и априорные убеждения о метках регионов. На второй стадии пиксели классифицируются с учётом выхода первой стадии, агрегированного по прямоугольным регионам изображения. Априорные убеждения аналогичны убеждениям, который предлагаемый метод получает на первой итерации, а прямоугольные регионы изображения аналогичны передатчикам пространственных д-факторов. На самом деле, в STF можно предложить использовать больше двух итераций.

Модель *«entanglement forest»* [88] обобщает и автоконтекст, и STF. Новой является идея использования контекстуальных зависимостей непосредственно в структуре элементарного классификатора. Модель состоит из набора решающих деревьев. В узлах этих деревьев вычисляются признаки на основе предсказаний, сделанных вершинами на более высоких уровнях в соседних локациях. Аналогичная идея используется в модели *«geodesic forest»* [89]. Дальнодействующие зависимости в ней учитываются с помощью *признаков мягкой связности*, которые могут быть эффективно вычислены с помощью обобщённого преобразования расстояний.

Модели *«вещей и материалов»* (англ. *things and stuff, TAS*) [90], также как и предложенный метод, моделирует дальнодействующие зависимости в сцене, изучая их по данным. В терминах этой статьи, *вещи* — объекты определённой формы, такие как люди или автомобили; а *материалы* — это аморфные регионы, характеризующиеся цветом и текстурой, такие как дорога или трава. Авторы демонстрируют, как находить объекты, используя контекст материалов. Они предполагают, что в сценах существуют значимые пространственные зависимости, такие как «автомобили паркуются примерно в 10 метрах от зданий», которое может транслироваться в термины изображений как «обнаружение i находится в 100 пикселях от региона j». Модель материалов обучается без учителя, так что подобный вид зависимостей можно рассматривать как частично семантический контекст. На этапе обучения генерируется избыточное множество возможных типов зависимостей, затем применяется структурный EM-алгоритм для отбора значимых. В предлагаемом методе подобную функцию выполняет L_1-регуляризация.

Ещё одна связанная модель была предложена Дезаи и др. [91]. Она также служит для обнаружения объектов, но моделирует контекстуальные зависимости только между *вещами*. Также как и в TAS, генерируется избыточный набор обнаружений объектов. Над ними задаётся марковская сеть, переменные которой определяют категорию каждого из обнаружений (или её отсутствие). Унарные потенциалы определяются как отклик детектора. Каждая пара обнаружений порождает ребро в марковской сети. Парные потенциалы моделируют, насколько вероятно пара объектов данных категорий будет находиться в определённой пространственной конфигурации. Эти конфигурации кодируют следующие взаимные расположения объектов: ‘далеко’, ‘близко’, ‘над’, ‘под’, ‘рядом’, ‘поверх’. Например, конфигурация ‘под’ означает, что центр второго объекта находится строго ниже огибающего прямоугольника первого объекта. Это идеологически похоже на то, как определяются пространственные д-факторы в предлагаемом методе (см. раздел 4.3). Параметры парных потенциалов, регулирующие участие каждой из конфигураций, подбирается автоматически с помощью структурного SVM (раздел 1.3.2).

Муноз и др. [92] предложили метод послойной иерархической разметки (англ. *stacked hierarchical labeling*), который затем Хьон и др. [34] применили к сегментации трёхмерных облаков точек. Последовательная классификация выполняется на последовательных уровнях иерархической сегментации изображений, от грубого к тонкому. На каждом уровне выводится распределение меток в каждом из регионов, оно же добавляется к признакам при определении меток на более низком уровне иерархии. Контекстуальные зависимости могут быть учтены

с помощью добавления меток верхнего уровня, собранных в регионе выше и ниже данного суперпикселя — это более простой аналог используемых здесь пространственных д-факторов. Также к признакам добавляются усреднённые по всем суперпикселям изображения распределения меток с верхнего уровня, что позволяет учитывать глобальный контекст. Росс и др. [26] дали интерпретацию последовательной классификации как вывода в произвольной марковской сети, возможно с факторами высоких порядков. Рис. 4.1 объясняет отличие этого метода от используемого нами.

Марковские сети со стандартными алгоритмами вывода могут использоваться для учёта локального контекста, но не дальнодействующих связей — в этом случае вывод стал бы невозможным из-за высокой вычислительной сложности. Например, при сегментации облаков точек каждая точка может соединяться парными связями с k ближайшими соседями, однако k не может быть большим. Один из таких методов предложен в главе 3, там же дан обзор релевантной литературы.

4.5 Результаты экспериментов

4.5.1 Данные и постановка эксперимента

Экспериментальная верификация проведена с использованием набора данных, собранного Коппулой и др. [86]. Он представляет собой зарегистрированные карты глубины и RGB-изображения, полученные датчиком Kinect. Для съёмки использовались комнаты жилых и офисных помещений, 24 и 28 комнат, соответственно. Для получения облака точек, соответствующего одной сцене, использовались 8–9 сканов. Облака точек были вручную сегментированы на 17 категорий с помощью ручной разметки суперпикселйей. Для разметки офисных сцен использовались следующие категории: ‘стена’, ‘пол’, ‘столешница’, ‘ящик стола’, ‘ножка стола’, ‘спинка стула’, ‘сиденье стула’, ‘зад стула’, ‘перед принтера’, ‘клавиатура’, ‘верх компьютера’, ‘перед компьютера’, ‘торец компьютера’, ‘книга’, ‘бумага’. Для разметки жилых сцен используются: ‘стена’, ‘пол’, ‘верх компьютера’, ‘ящик стола’, ‘ножка стола’, ‘спинка стула’, ‘сиденье стула’, ‘сиденье дивана’, ‘подлокотник дивана’, ‘спинка дивана’, ‘кровать’, ‘торец кровати’, ‘одеяло’, ‘подушка’, ‘полка’, ‘ноутбук’, ‘книга’.

Выполняется скользящий контроль по 4 частям выборки для жилых и офисных сцен по отдельности. Каждая из сцен может принадлежать только одной части. В облаках точек остаются только суперпиксели тех 17 категорий, которые использовались для разметки данных соответствующего типа, фоновые суперпиксели не учитываются. Таким образом, остаётся 690 суперпикселей в офисных сценах и 800 — в жилых. В обоих наборах большинство суперпикселей принадлежат к категории ‘стена’. Структурные связи в нашей модели соответствуют парным факторам, используемым Коппулой и др. [86].

В задачах бинарной классификации традиционными мерами качества являются точность (англ. *precision*) и полнота (англ. *recall*), показывающие, соответственно, долю верно обнаруженных объектов среди объектов целевого класса, и долю верно обнаруженных среди объектов, отнесённых к целевому классу. Их можно обобщить на многоклассовый случай дву-

Таблица 4.3: Результаты экспериментов на *офисных* и *жилых* сценах. Показана оценка скользящего контроля микро- и макро-точности и макро-полноты после 5 итераций обучения. STR: модель, в которой используются только структурные факторы. STR+SPAT: используются структурный и пространственные типы факторов с единичными коэффициентами. STR+SPAT_C: используются структурный и пространственные типы факторов с настраиваемыми коэффициентами, полученными максимизацией регуляризованной целевой функции (4.6), $C = 0.03$.

Метод	Офисные сцены			Жилые сцены		
	микро-	макро-		микро-	макро-	
	т/п	точность	полнота	т/п	точность	полнота
шанс	0.262	0.058	0.058	0.293	0.058	0.058
SVM_CRF [86]	0.840	0.805	0.726	0.722	0.568	0.548
STR	0.889	0.872	0.825	0.777	0.690	0.609
STR+SPAT	0.866	0.811	0.794	0.711	0.578	0.527
STR+SPAT_C	**0.902**	**0.882**	**0.844**	**0.783**	**0.716**	**0.620**

мя способами: с помощью микро- и макроусреднения. Обе меры интересны, так как микро-точность p (также известная как аккуратность, англ. *accuracy*) недооценивает неправильную разметку недостаточно представленных категорий, а макро-точность P и макро-полнота R учитывают все категории одинаково, независимо от их размера:

$$p = \frac{\sum_{k=1}^{K} TP_k}{\sum_{k=1}^{K} TP_k + FP_k} = \frac{\sum_{k=1}^{K} TP_k}{\sum_{k=1}^{K} TP_k + FN_k} = r, \tag{4.7}$$

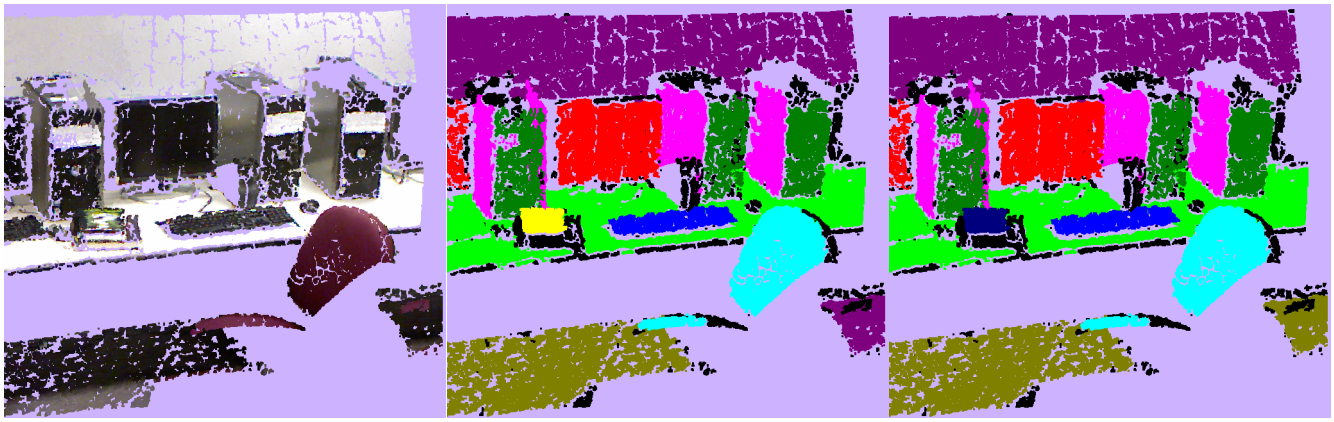
$$P = \frac{1}{K}\sum_{k=1}^{K} \frac{TP_k}{TP_k + FP_k}, \; R = \frac{1}{K}\sum_{k=1}^{K} \frac{TP_k}{TP_k + FN_k}, \tag{4.8}$$

где TP_k, FP_k, TN_k, FN_k — число истинно-положительных, ложноположительных, истинно-отрицательных и ложноотрицательных обнаружений для категории k, соответственно. Результаты собраны в таблице 4.3.

4.5.2 Качество сегментации

Модель, использующая только структурные зависимости (STR) показывает лучший результат, чем марковская сеть [86], хотя она использует ту же самую структуру зависимостей и признаки.

Добавление пространственных д-факторов с единичными весами (STR+SPAT), хотя и имеет теоретико-вероятностное обоснование, ухудшает результат по сравнению с моделью без пространственных д-факторов. Модель с оптимизацией по весам (STR+SPAT_C) теоретически должна работать лучше (по крайней мере, на обучающей выборке), так как является обобщением предыдущих: если все веса равны 1, она вырождается в наивную комбинацию структурных и пространственных д-факторов. Последние могут быть отключены установкой соответствующих весов равными нулю. Чтобы предотвратить переобучение, при настройке



(a) Исходное облако цветных точек

(b) Результат с использованием только структурных факторов

(c) Результат с использованием структурных и пространственных факторов

Рисунок 4.4: Пример сцены, в которой использование пространственных факторов улучшает качество сегментации. Модель, использующая только структурные факторы (b) неправильно классифицирует книгу (слева) и пол (справа), при этом модель, в которой также присутствуют пространственные факторы (c) корректно сегментирует всю сцену. Цветовое кодирование: ***‘стена’***, ***‘пол’***, ***‘столешница’***, ***‘стул’***, ***‘монитор’***, ***‘клавиатура’***, ***‘верх компьютера’***, ***‘перед компьютера’***, ***‘торец компьютера’***, ***‘книга’***.

весов используется регуляризация. На практике при большом коэффициенте регуляризации веса пространственных факторов стремятся к нулю.

На наборе офисных данных добавление пространственных типов факторов влечёт улучшение качества на 1–1.5 процентных пункта. Худшую разницу в производительности на сканах жилых помещений можно объяснить особенностями данных. В офисных данных обычно присутствует одна длинная стена, в то время как в жилых данных много углов. Около угла, направление к стене определяется неустойчиво, так как рядом расходятся две стены, так что «горизонтальные» типы факторов ненадёжны. Несмотря на то что при этом «вертикальные» типы факторов всё ещё значимы, они во многом дублируют структурные д-факторы, соединяющие пары суперпикселей, расстояние между которыми не превосходит 0.6 м. Поскольку высота сцены небольшая, большинство пар суперпикселей, близких по горизонтальной позиции, соединены структурными д-факторами.

При проведении экспериментов использовалось фиксированное значение гиперпараметра $C = 0.03$. При достаточном объёме данных его настройка может улучшить результат. Пространственные типы факторов (таблица 4.1) были заданы вручную, следовательно, субоптимально. Поскольку пространственные типы факторов параметризованы непрерывными переменными, подбор идеальных типов факторов можно осуществить с помощью градиентной оптимизации или направленной случайной выборки. Для этого также желательно иметь много данных, чтобы избежать переобучения.

Рассмотрим пример сегментации скана, изображённого на рис. 4.4b. Модель с только структурными типами факторов классифицирует книгу в левой части сцены как верх компьютера из-за соседства с суперпикселями категорий ‘перед компьютера’ и ‘торец компьюте-

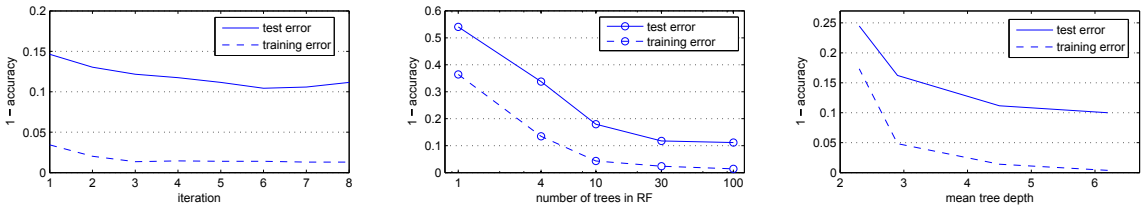


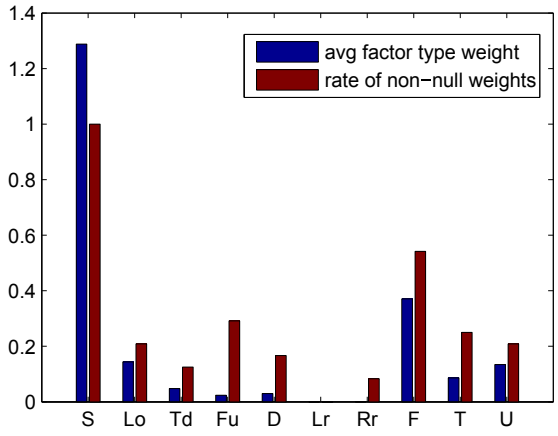
Рисунок 4.5: Слева: эволюция ошибки на тестовой и на обучающей выборках на наборе *офисных* данных при использовании 100 деревьев в ансамбле. Ошибка на обучении уменьшается, при этом ошибка на тесте стабилизируется и затем начинает расти после 5–6 итераций из-за эффекта переобучения. В центре: ошибка после 5 итераций в зависимости от числа деревьев в ансамбле. Справа: Ошибка после 5 итераций в зависимости от средней глубины решающих деревьев в ансамбле.

ра’. Пространственные признаки структурных факторов недостаточно экспрессивны, чтобы запретить обнаружение категории ‘верх компьютера’ везде, кроме как наверху суперпикселей, отнесённых к ‘переду компьютера’ и ‘торцу компьютера’. Пространственные д-факторы учитывают это в явном виде, и книга корректно классифицируется (рис. 4.4c). Поскольку структурные факторы ограничены по длине, они не моделируют зависимость между столешницей и полом на рисунке справа. Увеличение минимальной длины структурной связи более 0.6 м приведёт к запоминанию зависимостей, являющихся выбросами, так как структура зависимостей усложняется с расстоянием, а размер обучающей выборки ограничен [86]. Модель с пространственными типами факторов корректно распознаёт пол.

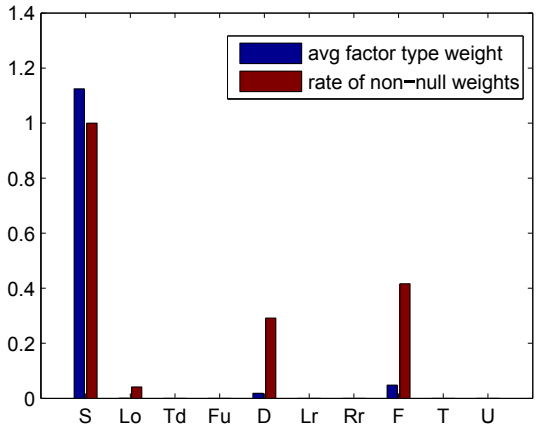
4.5.3 Вычислительная сложность и число итераций

Важным преимуществом последовательной классификации перед графическими моделями является быстрый вывод. Для модели со структурными д-факторами и д-факторами девяти пространственных типов среднее время вывода равно 0.7 секунды на сцену. При этом вывод останавливался после 5 итераций, использовался компьютер с 8-ядерным процессором и достаточным количеством памяти. В это время не включено выполнение предобработки, при которой вычисляются индексы всех переменных, составляющих структурные и пространственные д-факторы. Этот процесс может занять несколько минут для одной сцены, если передатчики пространственных д-факторов вычисляются для суперпикселей (как показано на рис. 4.3c), а не для их центров масс. Для сравнения, MAP-вывод в марковских сетях для этой задачи, реализованный с помощью смешанного целочисленного программирования, занимает 18 минут для одной сцены; решение LP-релаксации с помощью квадратичной псевдобулевой оптимизации быстрее (10 секунд), но точность ниже на 2–3 п. п. [86]. Таким образом, предложенный метод либо в тысячу раз быстрее и на 6 п. п. точнее, либо в десять раз быстрее и на 8 п. п. точнее.

Эксперименты показали, что обычно достаточно 5 итераций последовательной классификации. После этого точность стабилизируется и затем начинает падать из-за эффекта переобу-



(a) Веса для *офисных* данных



(b) Веса для *жилых* данных

Рисунок 4.6: Веса типов факторов, усреднённые по факторам и итерациям, а также доля ненулевых д-факторов каждого типа для *офисных* и *жилых* данных. Веса структурных (S) д-факторов не обращаются в ноль, в то время как пространственные типы факторов ‘лево’ (Lr) и ‘право’ (Rr) практически бесполезны. Это означает, например, что на столах обычно нет устоявшегося порядка предметов.

чения (рис. 4.5). В течение всего процесса обучения точность с пространственными типами факторов всегда выше, чем без них.

4.5.4 Анализ пространственных типов факторов

Из-за L_1-регуляризации в целевой функции (4.6) вектор оптимальных весов $\boldsymbol{\alpha}$ получается разреженным. Если некоторый вес обращается в нуль, это означает, что соответствующий тип факторов неинформативен, так что веса могут дать понять, какое подмножество типов факторов достаточно для моделирования пространственных зависимостей (рис. 4.6). Степень разреженности зависит от номера итерации и коэффициента регуляризации C. Эксперименты показали, что на первой итерации только вес, соответствующий структурным д-факторам, оказался ненулевым — для обоих наборов данных. Можно сделать вывод, что структурные д-факторы задают сильные зависимости, которые могут потом уточняться с помощью пространственных д-факторов. Таким образом, локальные зависимости оказываются сильнее дальнодействующий.

4.6 Выводы

В этой главе описан новый метод для семантической сегментации трёхмерных облаков точек, основанный на методе машины вывода [26]. Метод способен в явном виде учитывать семантический контекст. Он превосходит марковскую сеть, обученную структурным SVM [86] как по качеству, так и по скорости сегментации. Описанный метод может применяться и

для других задач разметки, где существуют дальнодействующие зависимости, например, в сегментации изображений.

Заключение

В данной работе предложены различные методы машинного обучения для задач совместной разметки. Они имеют определённые преимущества перед другими методами и друг перед другом:

1. Метод обучения задач разметки по данным с различными типами аннотации, описанный в главе 2, позволяет сократить человеческие усилия, необходимые при подготовке обучающей выборки. В настоящее время становятся доступными большие объёмы неаннотированных данных, таких как фотографии в сети Интернет, при этом ручная разметка пикселей по категориям достаточно трудоёмка и не может быть получена для большого количества фотографий. Для задачи семантической сегментации изображений проведено исследование других возможностей аннотирования выборки, которые, с одной стороны, будут простыми для ручной генерации, с другой — достаточно информативными для обучения сегментатора. Сформулирован общий принцип для определения целевой функции в задачах структурного обучения разметки по слабоаннотированным данным.

2. Метод обучения неассоциативных марковских сетей, описанный в главе 3, позволяет настроить подробную модель парных потенциалов при наличии достаточного количества размеченных обучающих данных или при небольшом числе категорий. Также описаны модификации метода, позволяющие расширить его применимость.

3. Пространственная машина вывода, описанная в главе 4, позволяет учитывать дальнодействующий семантический контекст. Это особенно полезно в задаче сегментации облаков точек, где локальная информация обычно невыразительна, и поэтому критично производить разметку совместно. Кроме этого, модель обеспечивает относительно быстрый вывод разметки, что может быть полезно в интерактивных приложениях.

Для демонстрации практической применимости разработанных методов были произведены эксперименты на реальных данных, где все они показали превосходство над базовыми методами в своих нишах применимости.

Список рисунков

Список таблиц

Список алгоритмов

Литература

1. Szeliski Richard. Computer vision: algorithms and applications. New York, NY: Springer-Verlag, 2010. URL: http://szeliski.org/Book.

2. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation / Jamie Shotton, John Winn, Carsten Rother [и др.] // European Conference on Computer Vision. 2006. С. 1–14. URL: http://jamie.shotton.org/work/publications/eccv06.pdf.

3. Shotton Jamie, Johnson Matthew, Cipolla Roberto. Semantic texton forests for image categorization and segmentation // IEEE Conference on Computer Vision and Pattern Recognition. 2008. June. URL: http://research.microsoft.com/pubs/117887/cvpr08.pdf.

4. Kohli Pushmeet, Torr Philip H.S. Measuring uncertainty in graph cut solutions // Computer Vision and Image Understanding. 2008. URL: http://eprints.pascal-network.org/archive/00006552/01/kt_cviu08_final.pdf.

5. Discriminative Learning of Markov Random Fields for Segmentation of 3D Scan Data / Dragomir Anguelov, Ben Taskar, Vassil Chatalbashev [и др.] // IEEE Conference on Computer Vision and Pattern Recognition. San Diego, CA: 2005. С. 169–176. URL: http://ai.stanford.edu/ vasco/pubs/cvpr05.pdf.

6. Contextual classification with functional Max-Margin Markov Networks / Daniel Munoz, J. Andrew Bagnell, Nicolas Vandapel [и др.] // IEEE Conference on Computer Vision and Pattern Recognition. Miami, FL: 2009. June. С. 975–982. URL: http://repository.cmu.edu/cgi/viewcontent.cgi?article=1039&context=robotics.

7. Hoiem Derek, Efros Alexei, Hebert Martial. Putting Objects in Perspective // IEEE Conference on Computer Vision and Pattern Recognition. 2006. С. 2137–2144. URL: http://repository.cmu.edu/cgi/viewcontent.cgi?article=1282&context=robotics.

8. Scharstein D, Szeliski R. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms // International Journal of Computer Vision. 2002. Т. 47, № 1. С. 7–42. URL: http://vision.middlebury.edu/stereo/taxonomy-IJCV.pdf.

9. Geman Stuart, Geman Donald. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images // IEEE Transactions on Pattern Analysis and Machine Intelligence. 1984. № 6. С. 721–741. URL: http://www.csee.wvu.edu/ xinl/library/papers/infor/Geman_Geman.pdf.

10. Roth Stefan, Black Michael J. Fields of Experts // International Journal of Computer Vision. 2009. January. Т. 82, № 2. С. 205–229. URL: http://cs.brown.edu/ black/Papers/rothIJCV09.pdf.

11. Discriminative Non-blind Deblurring / Uwe Schmidt, Carsten Rother, Sebastian Nowozin [и др.] // IEEE Conference on Computer Vision and Pattern Recognition. Portland, OR: 2013. URL: http://jancsary.net/wp-uploads/2013/04/schmidt_et_al_cvpr2013.pdf.

12. Brill Eric. A simple rule-based part of speech tagger // Conference on Applied Computational Linguistics. Trento, IT: 1992. С. 112–116. URL: http://ucrel.lancs.ac.uk/acl/H/H92/H92-1022.pdf.

13. Lafferty John, McCallum Andrew, Pereira Fernando C.N. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data // International Conference on Machine Learning. Т. 2001. 2001. С. 282–289. URL: http://repository.upenn.edu/cis_papers/159.

14. Kassel Robert H. A comparison of approaches to on-line handwritten character recognition. Ph.D. thesis: Massachusetts Institute of Technology. 1995. URL: http://dspace.mit.edu/handle/1721.1/11407.

15. Rabiner Lawrence R. A tutorial on hidden Markov models and selected applications in speech recognition // Proceedings of the IEEE. 1989. Т. 77, № 2. С. 257–286. URL: http://books.google.com/books?hl=en&lr=&id=iDHgboYRzmgC&oi=fnd&pg=PA

16. Global discriminative learning for higher-accuracy computational gene prediction. / Axel Bernal, Koby Crammer, Artemis Hatzigeorgiou [и др.] // PLoS Computational Biology. 2007. March. Т. 3, № 3. с. e54. URL: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=1828702&tool=pmcentrez&rendertype=ab

17. Yanover Chen, Schueler-Furman Ora, Weiss Yair. Minimizing and learning energy functions for side-chain prediction // Journal of Computational Biology. 2008. September. Т. 15, № 7. С. 899–911. URL: http://w3.cs.huji.ac.il/ yweiss/recomb07-final.pdf.

18. Bishop Christopher M. Pattern Recognition and Machine Learning / под ред. M Jordan, J Kleinberg, B Schölkopf. Springer, 2006. Т. 4 из *Information science and statistics*. с. 738. URL: http://www.library.wisc.edu/selectedtocs/bg0137.pdf.

19. Taskar Ben, Guestrin Carlos, Koller Daphne. Max-margin Markov networks // NIPS. 2003. URL: http://books.nips.cc/papers/files/nips16/NIPS2003_AA04.pdf.

20. Joachims Thorsten, Finley Thomas, Yu C.N.J. Cutting-plane training of structural SVMs // Machine Learning. 2009. Т. 77, № 1. С. 27–59. URL: http://tfinley.net/research/joachims_etal_09a.pdf.

21. Tarlow Daniel, Zemel Richard S. Structured Output Learning with High Order Loss Functions // International Conference on Artificial Intelligence and Statistics. 2012. URL: http://www.cs.toronto.edu/ dtarlow/tarlow_zemel_aistats12.pdf.

22. Pletscher Patrick, Kohli Pushmeet. Learning low-order models for enforcing high-order statistics // International Conference on Artificial Intelligence and Statistics. 2012. URL: http://research.microsoft.com/en-us/um/people/pkohli/papers/pk_aistats2012.pdf.

23. Max-Margin Parsing / Ben Taskar, Dan Klein, Michael Collins [и др.] // Conference on Empirical Methods on Natural Language Processing. Barcelona, Spain: 2004. URL: http://acl.ldc.upenn.edu/acl2004/emnlp/pdf/Taskar.pdf.

24. Weiss David, Sapp Benjamin, Taskar Ben. Structured Prediction Cascades: Tech. Rep.: : 2012.

25. Tu Zhuowen. Auto-context and its application to high-level vision tasks // IEEE Conference on Computer Vision and Pattern Recognition. Anchorage, AL: 2008. June. URL: http://www.loni.ucla.edu/ ztu/publication/cvpr08_autocontext.pdf.

26. Learning Message-Passing Inference Machines for Structured Prediction / Stephane Ross, Daniel Munoz, Martial Hebert [и др.] // IEEE Conference on Computer Vision and Pattern Recognition. Colorado Springs, CO: 2011. С. 2737–2744. URL: http://www.cs.cmu.edu/ sross1/publications/Ross-CVPR11.pdf.

27. Обучение алгоритма семантической сегментации изображений на выборке с разнообразными типами аннотаций / Роман Шаповалов, Дмитрий Ветров, Антон Осокин [и др.] // Интеллектуальные системы. 2014. Т. 18, № 3.

28. Shapovalov Roman, Velizhev Alexander, Barinova Olga. Non-associative Markov networks for 3D point cloud classification // Photogrammetric Computer Vision and Image Analysis. Paris, France: 2010. URL: http://shapovalov.ro/papers/Shapovalov-et-al-PCV2010.pdf.

29. Семантическая сегментация данных лазерного сканирования / Роман Шаповалов, Александр Велижев, Ольга Баринова [и др.] // Программные продукты и системы. 2012. № 1. С. 47–52.

30. Shapovalov Roman, Velizhev Alexander. Cutting-Plane Training of Non-associative Markov Network for 3D Point Cloud Segmentation // IEEE International Conference on 3D Imaging, Modeling, Processing, Visualisation and Transmittion. Hangzhou, China: 2011. С. 1–8. URL: http://shapovalov.ro/papers/Shapovalov-Velizhev-3dimpvt2011.pdf.

31. Shapovalov Roman, Vetrov Dmitry, Kohli Pushmeet. Spatial Inference Machines // IEEE Conference on Computer Vision and Pattern Recognition. Portland, OR: 2013. URL: http://shapovalov.ro/papers/SIM-Shapovalov-et-al-CVPR2013.pdf.

32. Munoz Daniel, Vandapel Nicolas, Hebert Martial. Directional associative markov network for 3-d point cloud classification // International Symposium on 3D

Data Processing, Visualization and Transmission. Atlanta, GA: 2008. URL: http://www.cc.gatech.edu/conferences/3DPVT08/Program/Papers/paper200.pdf.

33. Franc V., Savchynskyy B. Discriminative learning of max-sum classifiers // Journal of Machine Learning Research. 2008. T. 9. C. 67–104. URL: http://jmlr.csail.mit.edu/papers/volume9/franc08a/franc08a.pdf.

34. 3-D Scene Analysis via Sequenced Predictions over Points and Regions / Xuehan Xiong, Daniel Munoz, J. Andrew Bagnell [и др.] // IEEE International Conference on Robotics and Automation. Shanghai, China: 2011. URL: http://www.cs.princeton.edu/courses/archive/spring11/cos598A/pdfs/Xiong11.pdf.

35. Fixed-Point Model For Structured Labeling / Quannan Li, Jingdong Wang, David Wipf [и др.] // International Conference on Machine Learning. Atlanta, GA: 2013. URL: http://research.microsoft.com/pubs/179821/icml_2013_final_dpw.pdf.

36. Murphy Kevin P. Machine learning: a probabilistic perspective. Cambridge, MA; London, UK: The MIT Press, 2012. c. 1067. URL: http://dl.acm.org/citation.cfm?id=2380985.

37. Kohli Pushmeet, Kumar M.P., Torr P.H.S. P3 and Beyond: Solving Energies with Higher Order Cliques // IEEE Conference on Computer Vision and Pattern Recognition. Minneapolis, MN: 2007. URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.119.2624&rep=rep1&type=pdf.

38. Koller Daphne, Friedman Nil. Probabilistic graphical models: principles and techniques. Cambridge, Massachusets: MIT Press, 2009. c. 1231. URL: http://books.google.com/books?hl=en&lr=&id=7dzpHCHzNQ4C&oi=fnd&pg=PR

39. Dagum Paul, Luby Michael. Approximating probabilistic inference in Bayesian belief networks is NP-hard // Artificial Intelligence. 1993. T. 60. C. 141–153. URL: http://commonsenseatheism.com/wp-content/uploads/2011/12/Dagum-Luby-Approximating-probabilistic-inference-in-Bayesian-belief-networks-is-NP-hard.pdf.

40. A Comparative Study of Modern Inference Techniques for Discrete Energy Minimization Problems / Jörg H. Kappes, Bjoern Andres, Fred A. Hamprecht [и др.] // IEEE Conference on Computer Vision and Pattern Recognition. Portland, OR: 2013. URL: http://ipa.iwr.uni-heidelberg.de/ipabib/Papers/Kappes-etal-cvpr-2013-benchmark.pdf.

41. Komodakis Nikos, Paragios Nikos, Tziritas Georgios. MRF Optimization via Dual Decomposition: Message-Passing Revisited // IEEE International Conference on Computer Vision. № 2. 2007. URL: http://www.cs.ualberta.ca/ jag/papersVis2/07ICCV/data/papers/ICCV/053.pdf.

42. Komodakis Nikos, Paragios Nikos. Beyond pairwise energies: Efficient optimization for higher-order MRFs // IEEE Conference on Computer Vision

and Pattern Recognition. Miami, FL: 2009. June. C. 2985–2992. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5206846.

43. Kolmogorov Vladimir. Convergent tree-reweighted message passing for energy minimization // IEEE Transactions on Pattern Analysis and Machine Intelligence. 2006. T. 28, № 10. C. 1568–1583. URL: http://www.cs.ucl.ac.uk/staff/V.Kolmogorov/papers/TRW-S-PAMI.pdf.

44. Globerson Amir, Jaakkola TS. Fixing max-product: Convergent message passing algorithms for MAP LP-relaxations // NIPS. Vancouver, Canada: 2007. URL: http://machinelearning.wustl.edu/mlpapers/paper_files/NIPS2007_940.pdf.

45. Vetrov Dmitry, Osokin Anton, Kolmogorov Vladimir. Submodular Decomposition Framework for Inference in Associative Markov Networks with Global Constraints // IEEE Conference on Computer Vision and Pattern Recognition. Colorado Springs, CO: 2011. URL: http://www.cs.ucl.ac.uk/staff/V.Kolmogorov/papers/OVK_CVPR11_SMD.pdf.

46. Kolmogorov Vladimir, Zabih Ramin. What energy functions can be minimized via graph cuts? // IEEE Transactions on Pattern Analysis and Machine Intelligence. 2004. February. T. 26, № 2. C. 147–159. URL: http://www.ncbi.nlm.nih.gov/pubmed/15376891.

47. Boykov Yuri, Veksler Olga, Zabih Ramin. Fast approximate energy minimization via graph cuts // IEEE Transactions on Pattern Analysis and Machine Intelligence. 2001. T. 23, № 11. C. 1222–1239. URL: http://www.csd.uwo.ca/ yuri/Papers/pami01.pdf.

48. Kohli Pushmeet, Kumar M. Pawan. Energy minimization for linear envelope MRFs // IEEE Conference on Computer Vision and Pattern Recognition. San-Francisco, CA: 2010. C. 1863–1870. URL: http://research.microsoft.com/en-us/um/people/pkohli/papers/kk_cvpr2010.pdf.

49. Gould Stephen. Max-margin Learning for Lower Linear Envelope Potentials in Binary Markov Random Fields // International Conference on Machine Learning. Bellevue, WA: 2011. URL: http://users.cecs.anu.edu.au/ sgould/papers/icml11-linEnvLearning.pdf http://users.cecs.anu.edu.au/ sgould/papers/talk-ICML-2011.pdf.

50. Kohli Pushmeet, Ladicky Lubor, Torr Philip H.S. Robust higher order potentials for enforcing label consistency // International Journal of Computer Vision. 2009. T. 82, № 3. C. 302–324. URL: http://research.microsoft.com/en-us/um/people/pkohli/papers/klt_IJCV09.pdf.

51. Associative hierarchical CRFs for object class image segmentation / L'ubor Ladický, Chris Russell, Pushmeet Kohli [и др.] // IEEE International Conference on Computer Vision. Kyoto, Japan: 2009. URL: http://www.robots.ox.ac.uk/ lubor/iccv09.pdf.

52. Fast Approximate Energy Minimization with Label Costs / Andrew Delong, Anton Osokin, Hossam N. Isack [и др.] // International Journal of Computer Vision. 2012. July. T. 96, № 1. C. 1–27. URL: http://www.csd.uwo.ca/ adelong3/pub/ijcv2011-labelcosts-preprint.pdf.

53. Anstreicher Kurt M., Wolsey Laurence A. Two “well-known” properties of subgradient optimization // Mathematical Programming. 2007. June. Т. 120, № 1. С. 213–220. URL: http://link.springer.com/10.1007/s10107-007-0148-y.

54. Block-Coordinate Frank-Wolfe Optimization for Structural SVMs / Simon Lacoste-Julien, Martin Jaggi, Mark Schmidt [и др.] // International Conference on Machine Learning. 2013. URL: http://arxiv.org/abs/1207.4747 http://www.pletscher.org/papers/lacostejulien2013fwstruct.pdf.

55. Shalev-Shwartz Shai, Singer Yoram, Srebro Nathan. Pegasos: Primal estimated sub-gradient solver for svm // International Conference on Machine Learning. Corvallis, OR: 2007. С. 807–814. URL: http://machinelearning202.pbworks.com/f/stochasticSubGradient-shalev-shwartz.pdf.

56. Efficient backprop / Yann LeCun, Leon Bottou, Genevieve B. Orr [и др.] // Neural Networks: Tricks of the Trade. 1998. URL: http://link.springer.com/chapter/10.1007/3-540-49430-8_2.

57. Payet Nadia, Todorovic Sinisa. (RF)ˆ2 — Random Forest Random Field // NIPS. Т. 1. Vancouver, Canada: 2010. URL: http://machinelearning.wustl.edu/mlpapers/paper_files/NIPS2010_0234.pdf.

58. Boosting Structured Prediction for Imitation Learning for Imitation Learning / Nathan Ratliff, David Bradley, J. Andrew Bagnell [и др.] // NIPS. Vancouver, Canada: 2007. URL: http://repository.cmu.edu/cgi/viewcontent.cgi?article=1053&context=robotics.

59. Decision Tree Fields / Sebastian Nowozin, Carsten Rother, Shai Bagon [и др.] // IEEE International Conference on Computer Vision. Barcelona, ES: 2011. URL: http://www.wisdom.weizmann.ac.il/ bagon/pub/DTF_iccv2011.pdf.

60. Vezhnevets Alexander, Ferrari Vittorio, Buhmann Joachim M. Weakly Supervised Semantic Segmentation with a Multi-Image Model // IEEE International Conference on Computer Vision. Barcelona, ES: 2011. URL: http://www.inf.ethz.ch/personal/vezhneva/Pubs/WeaklySupSemSeg.pdf.

61. Vezhnevets Alexander, Ferrari Vittorio, Buhmann Joachim M. Weakly Supervised Structured Output Learning for Semantic Segmentation // IEEE Conference on Computer Vision and Pattern Recognition. Providence, RI: 2012. URL: http://www.inf.ethz.ch/personal/vezhneva/Pubs/VezhnevetsCVPR2012b.pdf.

62. Structured output learning with indirect supervision / Ming-Wei Chang, Vivek Srikumar, Dan Goldwasser [и др.] // International Conference on Machine Learning. 2010. URL: http://flake.cs.uiuc.edu/ mchang21/publication/CSGR10-slide.pdf http://www.icml2010.org/papers/522.pdf.

63. Learning specific-class segmentation from diverse data / M. Pawan Kumar, Haithem Turki, Dan Preston [и др.] // IEEE International Conference on Computer Vision. 2011. November. С. 1800–1807. URL: http://ai.stanford.edu/ pawan/publications/KTPK-ICCV2011.pdf.

64. Lou Xinghua, Hamprecht Fred A. Structured Learning from Partial Annotations // International Conference on Machine Learning. 2012. URL: http://icml.cc/2012/papers/753.pdf.

65. Yu Chun-Nam John, Joachims Thorsten. Learning structural SVMs with latent variables // International Conference on Machine Learning. Montreal, Canada: 2009. URL: http://www.cs.cornell.edu/ cnyu/papers/icml09_latentssvm.pdf.

66. Yuille A.L., Rangarajan Anand. The concave-convex procedure (CCCP) // NIPS. 2002. URL: http://books.nips.cc/papers/files/nips14/AA66.pdf.

67. Image segmentation with a bounding box prior / Victor Lempitsky, Pushmeet Kohli, Carsten Rother [и др.] // International Conference on Computer Vision. 2009. September. С. 277–284. URL: http://research.microsoft.com/en-us/um/people/pkohli/papers/lkrs_iccv09.pdf.

68. Taskar Ben, Chatalbashev Vassil, Koller Daphne. Learning associative Markov networks // International Conference on Machine Learning. Banff, Alberta, Canada: 2004. С. 102–109. URL: http://www.seas.upenn.edu/ taskar/pubs/mmamn.pdf.

69. Rapid and accurate large-scale coestimation of sequence alignments and phylogenetic trees. / Kevin Liu, Sindhu Raghavan, Serita Nelesen [и др.] // Science (New York, N.Y.). 2009. June. Т. 324, № 5934. С. 1561–4. URL: http://www.ncbi.nlm.nih.gov/pubmed/19541996.

70. Tighe Joseph, Lazebnik Svetlana. SuperParsing: Scalable Nonparametric Image Parsing with Superpixels // European Conference on Computer Vision. Heraklion, Grece: 2010. URL: http://www.cs.unc.edu/ jtighe/Papers/ECCV10/eccv10-jtighe.pdf.

71. Contour detection and hierarchical image segmentation / Pablo Arbeláez, Michael Maire, Charless Fowlkes [и др.] // IEEE Transactions on Pattern Analysis and Machine Intelligence. 2011. May. Т. 33, № 5. С. 898–916. URL: http://www.cs.berkeley.edu/ malik/papers/arbelaezMFM-pami2010.pdf.

72. Lowe David G. Distinctive Image Features from Scale-Invariant Keypoints // International Journal of Computer Vision. 2004. November. Т. 60, № 2. С. 91–110. URL: http://zenithlib.googlecode.com/svn/trunk/papers/cv/ijcv/2004-Distinctive_Image_Features_from_Scale-Invariant_Keypoints.pdf.

73. Vedaldi Andrea, Zisserman Andrew. Efficient Additive Kernels via Explicit Feature Maps // IEEE Conference on Computer Vision and Pattern Recognition. San-Francisco, CA: 2010. July. URL: http://www.robots.ox.ac.uk/ vgg/publications/papers/vedaldi10.pdf.

74. Felzenszwalb Pedro F., Huttenlocher Daniel P. Efficient Graph-Based Image Segmentation // International Journal of Computer Vision. 2004. September. Т. 59, № 2. С. 167–181. URL: http://cvcl.mit.edu/SUNSeminar/Felzenszwalb_IJCV04.pdf.

75. Mencia Eneldo Loza, Fuerkranz Johannes. Efficient Multilabel Classification Algorithms for Large-Scale Problems in the Legal Domain // Semantic Processing of Legal Texts. Berlin, Heidelberg, 2010. Т. 6036. С. 192–215. URL: http://www.ke.tu-darmstadt.de/publications/papers/loza10eurlex.pdf.

76. Finley Thomas, Joachims Thorsten. Training Structural SVMs when Exact Inference is Intractable // International Conference on Machine Learning. New York, NY: 2008. С. 304–311. URL: http://www.joachims.org/publications/finley_joachims_08a.pdf.

77. Instance-based AMN Classification for Improved Object Recognition in 2D and 3D Laser Range Data / R. Triebel, R. Shmidt, O.M. Mozos [и др.] // International Joint Conference on Artificial Intelligence. Hyderabad, India: 2007. С. 2225–2230. URL: http://www.informatik.uni-freiburg.de/ omartine/publications/triebel2007ijcai.pdf.

78. Posner Ingmar, Cummins Mark, Newman Paul. A generative framework for fast urban labeling using spatial and temporal context // Autonomous Robots. 2009. March. Т. 26, № 2-3. С. 153–170. URL: http://www.robots.ox.ac.uk:5000/ mjc/Papers/AutonomousRobots_HIP_MJC_PNM_2009.pdf.

79. Golovinskiy Aleksey, Kim Vladimir G., Funkhouser Thomas. Shape-based Recognition of 3D Point Clouds in Urban Environments // IEEE International Conference on Computer Vision. Kyoto, Japan: 2009. URL: http://www.cs.princeton.edu/gfx/pubs/Golovinskiy_2009_SRO/paper.pdf.

80. Scene Understanding in a Large Dynamic Environment through a Laser-based Sensing / Huijing Zhao, Yiming Liu, Xiaolong Zhu [и др.] // IEEE International Conference on Robotics and Automation. 2010. С. 127–133. URL: http://www.poss.pku.edu.cn/Data/publications/icra10.pdf.

81. Knopp Jan, Prasad Mukta, Van Gool Luc. Scene cut: Class-specific object detection and segmentation in 3D scenes // IEEE International Conference on 3D Digital Imaging, Modeling, Processing, Visualisation and Transmittion. 2011. С. 180–187.

82. Velizhev Alexander, Shapovalov Roman, Schindler Konrad. Implicit shape models for object detection in 3D point clouds // ISPRS Congress. Melbourne, Australia: 2012. URL: http://shapovalov.ro/papers/ISM-Velizhev-et-al-ISPRS2012.pdf.

83. Guttman Antonin. R-trees: A dynamic index structure for spatial searching // ACM SIGMOD International Conference on Management of Data. ACM New York, NY, USA, 1984. С. 47–57. URL: http://www.postgis.org/support/rtree.pdf.

84. Sun Yanmin, Kamel Mohamed S., Wang Yang. Boosting for learning multiple classes with imbalanced class distribution // IEEE International Conference on Data Mining. 2006. С. 592–602. URL: http://people.ee.duke.edu/ lcarin/ImbalancedClassDistribution.pdf.

85. Krähenbühl Philipp, Koltun Vladlen. Efficient inference in fully connected crfs with gaussian edge potentials // NIPS. Granada, ES: 2011. С. 1–9. URL: http://arxiv.org/abs/1210.5644.

86. Semantic Labeling of 3D Point Clouds for Indoor Scenes / Hema Swetha Koppula, Abhishek Anand, Thorsten Joachims [и др.] // NIPS. Granada, ES: 2011. URL: http://pr.cs.cornell.edu/sceneunderstanding/nips_2011.pdf.

87. Breiman Leo. Random forests // Machine Learning. 2001. Т. 45, № 1. С. 5–32. URL: http://www.springerlink.com/index/U0P06167N6173512.pdf.

88. Entangled decision forests and their application for semantic segmentation of CT images / Albert Montillo, Jamie Shotton, John Winn [и др.] // International Conference on Information Processing in Medical Imaging. 2011. URL: http://research.microsoft.com/pubs/146430/Criminisi_IPMI_2011c.pdf.

89. GeoF: Geodesic Forests for Learning Coupled Predictors / Peter Kontschieder, Pushmeet Kohli, Jamie Shotton [и др.] // IEEE Conference on Computer Vision and Pattern Recognition. Portland, OR: 2013. URL: http://research.microsoft.com/pubs/184825/geoForests_final.pdf.

90. Heitz Geremy, Koller Daphne. Learning spatial context: Using stuff to find things // European Conference on Computer Vision. Marseille, France: Springer, 2008. С. 30–43. URL: http://robotics.stanford.edu/ koller/Papers/Heitz%2BKoller:ECCV08.pdf.

91. Desai Chaitanya, Ramanan Deva, Fowlkes Charless. Discriminative models for multi-class object layout // IEEE International Conference on Computer Vision. Tokyo, Japan: Ieee, 2009. September. С. 229–236. URL: http://www.cse.wustl.edu/ mgeorg/readPapers/byVenue/iccv2009/desai2009_iccv_discriminativeMode

92. Munoz Daniel, Bagnell J. Andrew, Hebert Martial. Stacked hierarchical labeling // European Conference on Computer Vision. Heraklion, Grece: 2010. URL: http://www.ri.cmu.edu/pub_files/2010/9/munoz_eccv_10.pdf.