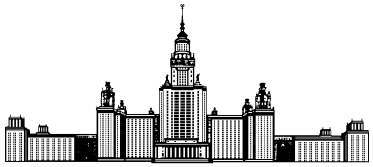
Московский государственный университет имени М. В. Ломоносова



Факультет вычислительной математики и кибернетики

кафедра математических методов прогнозирования

ДИПЛОМНАЯ РАБОТА СТУДЕНТА 517 ГРУППЫ

«Релаксационный подход в задаче структурного обучения по слаборазмеченным данным»

Выполнил:

студент 5 курса 517 группы

Кондрашкин Дмитрий Андреевич

Научный руководитель:

к.ф-м.н., доцент

Ветров Дмитрий Петрович

Москва, 2014

Содержание

1 Введение

Задача семантической сегментации заключается в отнесении каждого пикселя изображения к определенному классу, например: «автомобиль», «дерево», «дорога», «самолет» и т. д.

Будем моделировать семантическую сегментацию совместной разметкой пикселей: по признакам объекта $\mathbf{x} \in \mathcal{X}$ необходимо получить вектор меток $\mathbf{y} \in \mathcal{Y}$. В случае семантической сегментации $\mathcal{Y} = \{1, \ldots, K\}^V = \mathcal{K}^V$, где K — число классов, V — число пикселей в данном изображении, а в качестве признакового описания объекта $\mathbf{x}$ могут выступать как признаки пикселей (например цвет, текстура), так и признаки групп пикселей (например длина общей границы). Следует отметить, что метки соседних пикселей могут сильно коррелировать, поэтому для получения приемлемой точности недостаточно классифицировать каждый пиксель по отдельности.

Задача построения отображения $f\colon \mathcal{X} \to \mathcal{Y}$ по обучающей выборке $\{(\mathbf{x}^j, \mathbf{y}^j)\}_{j=1}^J$ называется *структурным обучением* (англ. *structural learning*), а применение такого отображения f называется *структурным предсказанием* (англ. *structural prediction*).

Для обучения достаточно точной модели семантической сегментации необходимо подготовить обучающую выборку большого объема, где в качестве ответа $\mathbf{y}$ используется пиксельная разметка изображения. Получение такой разметки производится вручную и является очень трудоемкой задачей.

Можно расширить обучающую выборку, добавив изображения со *слабой аннотацией*, в качестве которой выступает некоторая статистика от полной разметки. Применительно к задаче семантической сегментации можно рассмотреть несколько типов слабых аннотаций:

1. Множество классов, присутствующих на изображении (англ. *image-level labels*),
2. ограничивающие рамки объектов (англ. *bounding boxes*),
3. семена объектов (англ. *object seeds*).

В данной работе в качестве слабой аннотации рассматривается множество классов, присутствующих на изображении.

Для решения задачи структурного обучения с использованием слабой аннотации был предложен структурный метод опорных векторов с латентными переменны-

(a) Оригинальное изображение. (b) Полная пиксельная разметка.

(c) Список меток классов.

Рис. 1: Варианты аннотации обучающей выборки.

ми [18] (см. описание в разделе 2.7). В рамках этого метода возникает задача оптимизации некоторого функционала, имеющая высокую вычислительную сложность. Поэтому предлагаются разные аппроксимации функционала для приближенного решения задачи. Недостаток аппроксимации, предложенной в работе [18], заключается в том, что она не является строго обоснованной. **Целью** данной работы является исследование различных аппроксимаций этого функционала. Для этого были решены следующие задачи:

1. Построена корректная верхняя оценка минимизируемого функционала,
2. разработан и реализован метод оптимизации построенной оценки,
3. проведено экспериментальное сравнение метода со стандартным подходом.

Следующие секции устроены следующим образом: в секции 2 рассмотрены вспомогательные понятия, далее в секции 3 получена оптимизационная задача и выписан алгоритм для ее решения, в секции 4 приведены результаты экспериментов, в конце следует заключение.

2 Вспомогательные понятия

2.1 Марковское случайное поле

Пусть задан граф $G = (\mathcal{V}, \mathcal{E})$, $|\mathcal{V}| = V$, каждой вершине которого $v \in \mathcal{V}$ соответствует переменная $y_v \in \mathcal{K}$. Такой объект называется *Марковским случайным полем* (англ. *Markov random field, MRF*), подробнее см. [1]. Пусть C — клика в графе G, а $\mathbf{y}_C$ — набор переменных, соответствующих этой клике. Введем *функцию энергии* $E(\mathbf{y})$:

$$E(\mathbf{y}) = \sum_C \varphi_C(\mathbf{y}_C). \tag{1}$$

Здесь функции $\varphi_C(\mathbf{y}_C)$ могут зависеть от признаков объектов, обозначаемых $\mathbf{x}_C$, а также от параметров $\mathbf{w}$, т. е. правильнее было бы написать $E(\mathbf{y}, \mathbf{x}, \mathbf{w})$. Для удобства в данном разделе опускаем зависимость от $\mathbf{x}$ и $\mathbf{w}$, считая эти параметры фиксированными. Нас будет интересовать значения переменных $\mathbf{y}^*$, минимизирующие функцию энергии:

$$\mathbf{y}^* = \operatorname*{argmin}_{\bar{\mathbf{y}} \in \mathcal{Y}} E(\bar{\mathbf{y}}). \tag{2}$$

Процесс нахождения значений переменных, минимизирующих функцию энергии, называется *выводом* (англ. *inference*).

Зачастую при записи энергии выделяют потенциалы первого порядка (также называемые *унарными потенциалами*):

$$E(\mathbf{y}) = \sum_{v \in \mathcal{V}} \varphi_v(y_v) + \sum_{C:\, |C| \geqslant 2} \varphi_C(\mathbf{y}_C). \tag{3}$$

Выделим важный класс *парносепарабельных энергий*, состоящих только из унарных и парных потенциалов:

$$E(\mathbf{y}) = \sum_{v \in \mathcal{V}} \varphi_v(y_v) + \sum_{(u,v) \in \mathcal{E}} \varphi_{uv}(y_u, y_v). \tag{4}$$

Существуют эффективные приближенные методы для минимизации энергий такого вида:

1. Методы, которые находят неточный минимум исходной задачи, например, метод α-расширения [3], основанный на поиске минимального разреза в графе.

2. Методы, которые находят точный минимум некоторой релаксации, например метод TRW [10] (англ. *tree-reweighted message passing*), основанный на LP-релаксации функции энергии.

Далее, если не сказано иное, будем рассматривать парносепарабельные энергии вида 4.

2.2 Избыточное представление

В дальнейшем нам понадобится *избыточное представление* (англ. *overcomplete representation*) переменных $\mathbf{y} = (y_1, \dots, y_V)$. Каждую K-значную переменную y_v заменим на набор бинарных переменных $y_{v1}, \dots, y_{vK}$, из которых только одна может принимать единичное значение: $y_{vp} = 1 \Leftrightarrow y_v = p$. Аналогично для пары переменных (y_u, y_v), соединенных ребром, введем набор бинарных переменных $y_{uv11}, \dots, y_{uvKK}$, такой что $y_{uvpq} = 1 \Leftrightarrow y_u = p, y_v = q$. Обозначив $\theta_{vp} = \varphi_v(p), \theta_{uvpq} = \varphi_{uv}(p, q)$, перепишем энергию в виде линейной функции бинарных переменных:

$$E(\mathbf{y}) = \sum_{v \in \mathcal{V}} \sum_{p=1}^{K} \theta_{vp} y_{vp} + \sum_{(u,v) \in \mathcal{E}} \sum_{p,q=1}^{K} \theta_{uvpq} y_{uvpq}. \tag{5}$$

Выпишем ограничения на переменные $\mathbf{y}$:

$$\sum_{p=1}^{K} y_{vp} = 1, \quad \sum_{p=1}^{K} y_{uvpq} = y_{vq}, \quad \sum_{q=1}^{K} y_{uvpq} = y_{up}, \quad y_{vp}, y_{uvpq} \in \{0, 1\},$$
$$\forall v, u \in \mathcal{V}, \forall p, q \in \{1, \dots, K\}. \tag{6}$$

Полученный набор переменных y_{vp}, y_{uvpq} с ограничениями (6) называется избыточным представлением $\mathbf{y}$.

2.3 MRF для семантической сегментации изображений

Марковское случайное поле для семантической сегментации изображений задается на графе $G = (\mathcal{V}, \mathcal{E})$, где вершины $\mathcal{V}$ соответствуют пикселям (суперпикселям); две вершины соединены ребром, если соответствующие им пиксели (суперпиксели) имеют общую границу. Для каждой вершины $v \in \mathcal{V}$ задан унарный потенциал $\varphi_v(\cdot)$, для каждого ребра $(u, v) \in \mathcal{E}$ задан парный потенциал $\varphi_{uv}(\cdot, \cdot)$. На рис. 2 границы

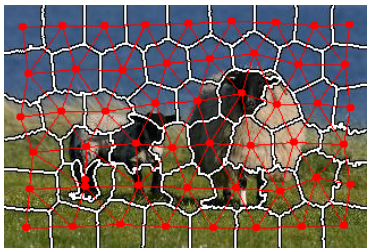


Рис. 2: MRF для семантической сегментации изображений.

суперпикселей выделены белыми линиями, переменные — красными точками, связи — красными линиями.

2.4 Структурное обучение

Пусть задана обучающая выборка $\{(\mathbf{x}^j, \mathbf{y}^j)\}_{j=1}^J$. Задачей структурного обучения является нахождение таких параметров $\mathbf{w}^*$ функции энергии $E(\mathbf{y}, \mathbf{x}, \mathbf{w})$, чтобы для каждого объекта j минимум энергии достигался на верной разметке $\mathbf{y}^j$. Таким образом, вывод в такой энергии будет давать верную разметку обучающих объектов.

2.5 Параметризация

Будем рассматривать только парносепарабельные энергии. Зададим *линейную параметризацию* энергии:

$$E(\mathbf{y}, \mathbf{x}, \mathbf{w}) = -\sum_{v \in \mathcal{V}} \mathbf{w}^\intercal \psi_v(y_v, \mathbf{x}_v) - \sum_{(u,v) \in \mathcal{E}} \mathbf{w}^\intercal \psi_{uv}(y_u, y_v, \mathbf{x}_{uv}) = -\mathbf{w}^\intercal \Psi(\mathbf{y}, \mathbf{x}). \tag{7}$$

Здесь $\mathbf{x}_v$ — вектор признаков, соответствующий вершине v, $\mathbf{x}_{uv}$ — соответствующий паре вершин (u, v). Вектор $\Psi(\mathbf{y}, \mathbf{x})$ обычно называют вектором *обобщенных признаков* (англ. *joint features*). Следует отметить, что в литературе, посвященной структурному обучению, обычно от энергии переходят к *оценочной* или *дискриминантной* функции $F(\mathbf{y}, \mathbf{x}, \mathbf{w}) = \mathbf{w}^\intercal \Psi(\mathbf{y}, \mathbf{x})$ (англ. *score function*, *discriminant function*), равной энергии с противоположным знаком.

Частным случаем линейной параметризации является *обобщенная модель Поттса* (англ. *generalized Potts model*):

$$-E(\mathbf{y}, \mathbf{x}, \mathbf{w}) = \mathbf{w}^\intercal \Psi(\mathbf{y}, \mathbf{x}) = \sum_{v \in \mathcal{V}} \sum_{k=1}^{K} [y_v = k] (\mathbf{w}_k^{\mathrm{u}})^\intercal \mathbf{x}_v + \sum_{(u,v) \in \mathcal{E}} \sum_{k=1}^{K} [y_u = k][y_v = k] (\mathbf{w}_k^{\mathrm{p}})^\intercal \mathbf{x}_{uv},$$

(8)

где $\mathbf{x}_v \in \mathbb{R}^d$ — вектор признаков, соответствующий вершине $v \in \mathcal{V}$, $\mathbf{x}_{uv} \in \mathbb{R}^e$ — вектор признаков, соответствующий ребру $(u, v) \in \mathcal{E}$, $\mathbf{w}_k^{\mathrm{u}} \in \mathbb{R}^d$, $\mathbf{w}_k^{\mathrm{p}} \in \mathbb{R}^e$, а $\mathbf{w} = (\mathbf{w}_1^{\mathrm{u}}, \ldots, \mathbf{w}_K^{\mathrm{u}}, \mathbf{w}_1^{\mathrm{p}}, \ldots, \mathbf{w}_K^{\mathrm{p}})$ — вектор параметров модели. Для удобства используем нотацию скобок Иверсона $[\cdot]$: результат равен единице, если логическое выражение внутри скобок истинно, нулю — если ложно. Далее будет использоваться именно обобщенная модель Поттса.

2.6 Структурный метод опорных векторов

При решении задачи структурного обучения методом *максимизации отступа* требуется найти такое значение параметра $\mathbf{w}^*$, чтобы величина оценочной функции $F(\mathbf{y}, \mathbf{x}, \mathbf{w})$ правильной разметки для каждого из объектов обучающей выборки с разметкой $\mathbf{y}^j$ была не только наибольшей, но и как можно дальше отстояла от второй по значению точки $\operatorname{argmax}_{\mathbf{y} \in \mathcal{Y} \setminus \{\mathbf{y}^j\}} F(\mathbf{y}, \mathbf{x}, \mathbf{w})$. Еще одной особенностью этого подхода является использование нетривиальной функции потерь при обучении. Более подробно см. [12]. Эти соображения приводят к постановке оптимизационной задачи *структурного метода опорных векторов* (англ. *structural support vector machine*, *SSVM*).

Оптимизационная задача 1.

$$\min_{\mathbf{w}, \boldsymbol{\xi} \geqslant 0} \frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{j=1}^{J} \xi_j, \tag{9}$$

$$s.t.\ \mathbf{w}^\intercal \Psi(\mathbf{y}^j, \mathbf{x}^j) \geqslant \max_{\bar{\mathbf{y}} \in \mathcal{Y}} \{\mathbf{w}^\intercal \Psi(\bar{\mathbf{y}}, \mathbf{x}^j) + \Delta(\bar{\mathbf{y}}, \mathbf{y}^j)\} - \xi_j,\ \forall j \in \{1, \ldots, J\}. \tag{10}$$

Здесь C — структурный параметр, отвечающий за вклад регуляризатора, $\Delta(\bar{\mathbf{y}}, \mathbf{y})$ — функция потерь, такая что $\Delta(\mathbf{y}, \mathbf{y}) = 0$ и $\Delta(\bar{\mathbf{y}}, \mathbf{y}) \geqslant 0, \forall \bar{\mathbf{y}}$. В качестве функции потерь часто используется расстояние Хэмминга: $\Delta(\bar{\mathbf{y}}, \mathbf{y}) = \sum_v [\bar{y}_v \neq y_v]$. При выполнении ограничений (10) значение $\mathbf{w}^\intercal \Psi(\mathbf{y}^j, \mathbf{x}^j)$ на правильной разметке больше, чем на любой другой разметке $\bar{\mathbf{y}}$ (с допуском ξ_j), причем отступ увеличивается при удалении $\bar{\mathbf{y}}$ от $\mathbf{y}^j$. Поэтому такой подход к обучению параметров модели называется *максимизацией отступа* между верной разметкой и второй после нее.

Внутренняя задача оптимизации называется *выводом, дополненным функцией потерь* (англ. *loss-augmented inference*). В случае, когда $\Delta(\bar{\mathbf{y}}, \mathbf{y})$ — расстояние Хэмминга, для выполнения оптимизации требуется модифицировать только унарные потенциалы и применить стандартный алгоритм вывода.

Запишем задачу структурного SVM в виде без ограничений.

Оптимизационная задача 2 (SSVM в виде без ограничений)**.**

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{j=1}^{J}\left(\max_{\bar{\mathbf{y}}\in\mathcal{Y}}\{\mathbf{w}^\intercal\Psi(\bar{\mathbf{y}}, \mathbf{x}^j) + \Delta(\bar{\mathbf{y}}, \mathbf{y}^j)\} - \mathbf{w}^\intercal\Psi(\mathbf{y}^j, \mathbf{x}^j)\right). \tag{11}$$

Разработаны специальные эффективные методы для решения задач такого вида, например метод *отсекающей плоскости* (англ. *cutting-plane*) [7], решающий задачу 1.

Также можно оптимизировать целевую функцию задачи 2, она является выпуклой, но недифференцируемой, поэтому можно применять метод субградиентного спуска.

2.7 Использование слабой аннотации в структурном обучении

В работе Йу и Йохимса [18] впервые был предложен *структурный метод опорных векторов с латентными переменными* (англ. *structural support vector machine with latent variables*), который позволяет обучать параметры энергии по слабоаннотированным данным, латентной переменной в этом случае является полная разметка.

Пусть обучающая выборка помимо J объектов $\{(\mathbf{x}^j, \mathbf{y}^j)\}_{j=1}^{J}$ с полной разметкой, содержит также I слабоаннотированных объектов $\{(\mathbf{x}^i, \mathbf{z}^i)\}_{i=J+1}^{J+I}$. Потребуем, чтобы произвольной слабой аннотации $\mathbf{z}$ соответствовало множество $L(\mathbf{z}) \subseteq \mathcal{Y}$ совместных с ней разметок. (В данной работе $\mathbf{z}$ — это множество классов, присутствующих на изображении). Рассмотрим обобщение метода SSVM, которое позволяет учитывать как объекты с полной разметкой, так и объекты со слабой аннотацией.

Оптимизационная задача 3 (Обобщенный SSVM)**.**

$$\min_{\mathbf{w},\boldsymbol{\xi}\geqslant 0,\boldsymbol{\eta}\geqslant 0} \frac{1}{2}\|\mathbf{w}\|^2 + C\left(\sum_{j=1}^{J}\xi_j + \alpha\sum_{i=1}^{I}\eta_i\right), \tag{12}$$

$$s.t.\ \mathbf{w}^\intercal\Psi(\mathbf{y}^j,\mathbf{x}^j) \geqslant \max_{\bar{\mathbf{y}}\in\mathcal{Y}}\{\mathbf{w}^\intercal\Psi(\bar{\mathbf{y}},\mathbf{x}^j) + \Delta(\bar{\mathbf{y}},\mathbf{y}^j)\} - \xi_j,\ \forall j\in\{1,\ldots,J\}, \tag{13}$$

$$\max_{\mathbf{y}\in L(\mathbf{z}^i)} \mathbf{w}^\intercal\Psi(\mathbf{y},\mathbf{x}^i) \geqslant \max_{\bar{\mathbf{y}}\in\mathcal{Y}}\{\mathbf{w}^\intercal\Psi(\bar{\mathbf{y}},\mathbf{x}^i) + \kappa(\bar{\mathbf{y}},\mathbf{z}^i)\} - \eta_{i-J},\ \forall i\in\{J+1,\ldots,I+J\}. \tag{14}$$

Здесь присутствует структурный параметр C, отвечающий за вклад регуляризатора, а также параметр α, отвечающий за вклад слабоаннотированных данных. В качестве функции потерь $\Delta(\bar{\mathbf{y}},\mathbf{y})$, как и ранее, выступает расстояние Хэмминга. Также появляется *слабая функция потерь* $\kappa(\bar{\mathbf{y}},\mathbf{z})$, которая задает степень несогласованности разметки $\bar{\mathbf{y}}$ со слабой аннотацией $\mathbf{z}$.

Задача 3 эквивалентна задаче безусловной минимизации следующей целевой функции (этот результат можно получить, избавившись от фиктивных переменных $\boldsymbol{\xi}$ и $\boldsymbol{\eta}$):

$$\mathcal{L}(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{j=1}^{J}\left(\max_{\bar{\mathbf{y}}\in\mathcal{Y}}\{\mathbf{w}^\intercal\Psi(\bar{\mathbf{y}},\mathbf{x}^j) + \Delta(\bar{\mathbf{y}},\mathbf{y}^j)\} - \mathbf{w}^\intercal\Psi(\mathbf{y}^j,\mathbf{x}^j)\right)$$
$$+\alpha C\sum_{i=J+1}^{J+I}\max_{\bar{\mathbf{y}}\in\mathcal{Y}}\{\mathbf{w}^\intercal\Psi(\bar{\mathbf{y}},\mathbf{x}^i) + \kappa(\bar{\mathbf{y}},\mathbf{z}^i)\} - \alpha C\sum_{i=J+1}^{J+I}\max_{\mathbf{y}\in L(\mathbf{z}^i)}\mathbf{w}^\intercal\Psi(\mathbf{y},\mathbf{x}^i). \tag{15}$$

Первые три слагаемых в (15) выпуклы по $\mathbf{w}$, а последнее с учетом знака минус — вогнуто. Это следует из того, что максимум конечного числа линейных функций является выпуклой функцией, так же как и сумма выпуклых функций является выпуклой функцией. Таким образом функционал $\mathcal{L}(\mathbf{w})$ является суммой выпуклой и вогнутой функций. Следуя подходу, предложенному в работе [18], для приближенной минимизации этого функционала воспользуемся *выпукло-вогнутой процедурой* [19] (англ. *convex-concave procedure, CCCP*). Идея метода заключается в итеративном построении линейной верхней оценки на вогнутую часть при фиксированных параметрах с предыдущей итерации, и минимизации суммы выпуклой и линеаризованной вогнутой частей. В нашем случае на этапе линеаризации решается задача

$$\mathbf{y}_i^* = \operatorname*{argmax}_{\mathbf{y}\in L(\mathbf{z}^i)}\mathbf{w}^\intercal\Psi(\mathbf{y},\mathbf{x}^i). \tag{16}$$

После подстановки $\mathbf{y}_i^*$ в функционал (15), он становится выпуклым (так как последнее слагаемое в нем становится линейным по $\mathbf{w}$), и может быть оптимизирован, например, методом отсекающей плоскости. Эти шаги повторяются до сходимости. Задача вывода (16) называется *выводом, согласованным с аннотацией* (англ. *annotation-consistent inference*).

2.8 Слабая функция потерь

Следуя [20], введем слабую функцию потерь для задачи семантической сегментации, где в качестве слабой аннотации используется множество классов на изображении:

$$\kappa(\bar{\mathbf{y}}, \mathbf{z}) = \sum_{k \notin \mathbf{z}} \sum_{v \in \mathcal{V}} [y_v = k] + \sum_{k \in \mathbf{z}} c_k \prod_{v \in \mathcal{V}} [y_v \neq k] =$$
$$\sum_{k \notin \mathbf{z}} \sum_{v \in \mathcal{V}} [y_v = k] - \sum_{k \in \mathbf{z}} c_k [\exists v \in \mathcal{V} \colon y_v = k] + const, \tag{17}$$

где c_k — некоторые неотрицательные константы, отвечающие за силу штрафа за отсутствие метки k на изображении. Выпишем теперь задачу *вывода, дополненного слабой функцией потерь*:

$$\operatorname*{argmax}_{\bar{\mathbf{y}} \in \mathcal{Y}} \left\{ \mathbf{w}^\intercal \Psi(\bar{\mathbf{y}}, \mathbf{x}) + \sum_{k \notin \mathbf{z}} \sum_{v \in \mathcal{V}} [y_v = k] - \sum_{k \in \mathbf{z}} c_k [\exists v \in \mathcal{V} \colon y_v = k] \right\}. \tag{18}$$

Это стандартная задача вывода со штрафами за метки (англ. *label costs*), она может быть эффективно решена с помощью модификации алгоритма α-расширения, предложенной в работе [5].

3 Релаксационный подход в задаче структурного обучения

В работе Финли и Йохимса [6] было показано, что использование релаксационного подхода для вывода, дополненного функцией потерь, позволяет повысить точность обученной модели. В работе рассматривалась задача обучения по полностью размеченным данным. В данной работе исследуется применение релаксационного подхода для аппроксимации функционала задачи обобщенного SSVM. Релаксационный

подход применяется для решения задачи вывода, дополненного функцией потерь, и задачи вывода, дополненного слабой функцией потерь. В разделе будет построена верхняя оценка на функционал (15), затем будет рассмотрен алгоритм для ее оптимизации. При минимизации верхней оценки мы можем быть уверены в том, что оптимальное значение целевой функции окажется не более некоторой величины. При других аппроксимациях этого гарантировать нельзя. Этот факт является основной мотивацией для построения верхней оценки. Релаксационный подход — подход к оптимизации, при котором множество допустимых значений оптимизируемых переменных расширяется.

3.1 Верхняя оценка

В функционале (15) есть три оптимизационных задачи:

1. Вывод, дополненный функцией потерь $\Delta(\bar{\mathbf{y}}, \mathbf{y})$:

$$\max_{\bar{\mathbf{y}}} \left\{ \mathbf{w}^\intercal \Psi(\bar{\mathbf{y}}, \mathbf{x}) + \Delta(\bar{\mathbf{y}}, \mathbf{y}) \right\}. \tag{19}$$

2. Вывод, дополненный слабой функцией потерь $\kappa(\bar{\mathbf{y}}, \mathbf{z})$:

$$\max_{\bar{\mathbf{y}}} \left\{ \mathbf{w}^\intercal \Psi(\bar{\mathbf{y}}, \mathbf{x}) + \kappa(\bar{\mathbf{y}}, \mathbf{z}) \right\}. \tag{20}$$

3. Вывод, согласованный с аннотацией:

$$\max_{\mathbf{y}\colon \mathbf{y} \in L(\mathbf{z})} \mathbf{w}^\intercal \Psi(\mathbf{y}, \mathbf{x}). \tag{21}$$

Для решения задачи (21) с учетом знака «минус» в функционале (15) следует использовать метод, который строит нижнюю оценку. В качестве такого метода в данной работе был выбран метод α-расширения [3]. Для решения задач (19) и (20) необходимо использовать методы, оптимизирующие верхнюю оценку. В следующих подразделах подробнее рассмотрим две оставшихся задачи.

3.1.1 Вывод, дополненный функцией потерь

Следуя подходу, предложенному в работе [9], для построения верхней оценки в задаче (19) воспользуемся методом *двойственного разложения* (англ. *dual*

decomposition). Рассмотрим задачу (19):

$$\max_{\bar{\mathbf{y}}} \{\mathbf{w}^\intercal \Psi(\bar{\mathbf{y}}, \mathbf{x}) + \Delta(\bar{\mathbf{y}}, \mathbf{y})\} =$$

$$\max_{\bar{\mathbf{y}}} \left\{ \sum_{v \in \mathcal{V}} (\mathbf{w}^\intercal \psi_v(\bar{y}_v, \mathbf{x}_v) + [\bar{y}_v \neq y_v]) + \sum_{(u,v) \in \mathcal{E}} \mathbf{w}^\intercal \psi_{uv}(\bar{y}_u, \bar{y}_v, \mathbf{x}_{uv}) \right\} =$$

$$\max_{\bar{\mathbf{y}}} \left\{ \sum_{v \in \mathcal{V}} \varphi_v(\bar{y}_v) + \sum_{(u,v) \in \mathcal{E}} \varphi_{uv}(\bar{y}_u, \bar{y}_v) \right\}, \tag{22}$$

где:

$$\varphi_v(\bar{y}_v) = \mathbf{w}^\intercal \psi_v(\bar{y}_v, \mathbf{x}_v) + [\bar{y}_v \neq y_v],$$
$$\varphi_{uv}(\bar{y}_u, \bar{y}_v) = \mathbf{w}^\intercal \psi_{uv}(\bar{y}_u, \bar{y}_v, \mathbf{x}_{uv}).$$

Разобьем исходный граф $G = (\mathcal{V}, \mathcal{E})$ на подграфы-деревья $G_\tau = (\mathcal{V}_\tau, \mathcal{E}_\tau)_{\tau=1}^T$ так, чтобы были выполнены условия:

$$\cup_{\tau=1}^T \mathcal{V}_\tau = \mathcal{V},$$
$$\cup_{\tau=1}^T \mathcal{E}_\tau = \mathcal{E},$$
$$\mathcal{E}_i \cap \mathcal{E}_j = \varnothing, \forall i, j \in \{1, \ldots, T\}.$$

Каждый такой подграф наследует парные потенциалы, но имеет свои собственные унарные потенциалы $\boldsymbol{\varphi}^\tau = \{\varphi_v^\tau(k)\}_{v \in \mathcal{V}_\tau, k \in \mathcal{K}}$. Потребуем, чтобы они были согласованными:

$$\sum_{\tau \in \mathcal{I}_v} \varphi_v^\tau(k) = \varphi_v(k), \forall v \in \mathcal{V}, k \in \mathcal{K}, \tag{23}$$

где $\mathcal{I}_v = \{\tau | v \in \mathcal{V}_\tau\}$ — множество индексов подграфов, содержащих вершину v. Введем новые переменные $\Lambda = \{\boldsymbol{\lambda}^\tau\}_{\tau=1}^T$, $\boldsymbol{\lambda}^\tau \in \mathbb{R}^{|\mathcal{V}_\tau| \times K}$:

$$\varphi_v^\tau(k) = \lambda_v^\tau(k) + \frac{\varphi_v(k)}{|\mathcal{I}_v|}, \forall v \in \mathcal{V}_\tau, k \in \mathcal{K}. \tag{24}$$

Тогда ограничения (23) на переменные $\{\boldsymbol{\varphi}^\tau\}_{\tau=1}^T$ перейдут в ограничения на переменные Λ:

$$\sum_{\tau \in \mathcal{I}_v} \lambda_v^\tau(k) = 0, \forall v \in \mathcal{V}, k \in \mathcal{K}. \tag{25}$$

Обозначим это множество ограничений через $\mathcal{C}$.

Воспользуемся избыточным представлением (см. раздел 2.2):

$$E(\mathbf{y}, \boldsymbol{\theta}) = \sum_{v \in \mathcal{V}} \sum_{p=1}^{K} \theta_{vp} y_{vp} + \sum_{(u,v) \in \mathcal{E}} \sum_{p,q=1}^{K} \theta_{uvpq} y_{uvpq}, \tag{26}$$

$$E^{\tau}(\mathbf{y}, \boldsymbol{\theta}, \boldsymbol{\lambda}^{\tau}) = \sum_{v \in \mathcal{V}_{\tau}} \sum_{p=1}^{K} \left(\frac{\theta_{vp}}{|\mathcal{I}_v|} + \lambda_{vp}^{\tau} \right) y_{vp} + \sum_{(u,v) \in \mathcal{E}_{\tau}} \sum_{p,q=1}^{K} \theta_{uvpq} y_{uvpq}. \tag{27}$$

Множество допустимых значений $\mathbf{y}$ (6) обозначим через $\mathcal{B}$. Произведем *релаксацию* путем замены ограничений $y_{vp}, y_{uvpq} \in \{0, 1\}$ на ограничения $y_{vp}, y_{uvpq} \in [0, 1]$, релаксированное множество допустимых значений обозначим через $\mathcal{R}$. Для любых значений Λ, удовлетворяющих ограничениям $\mathcal{C}$ (25), имеет место равенство:

$$E(\mathbf{y}, \boldsymbol{\theta}) = \sum_{\tau=1}^{T} E^{\tau}(\mathbf{y}, \boldsymbol{\theta}, \boldsymbol{\lambda}^{\tau}). \tag{28}$$

С учетом этого равенства справедлива следующая цепочка неравенств:

$$\begin{aligned} \max_{\mathbf{y} \in \mathcal{B}} E(\mathbf{y}, \boldsymbol{\theta}) &\leqslant \max_{\mathbf{y} \in \mathcal{R}} E(\mathbf{y}, \boldsymbol{\theta}) = \max_{\mathbf{y} \in \mathcal{R}} \sum_{\tau=1}^{T} E^{\tau}(\mathbf{y}, \boldsymbol{\theta}, \boldsymbol{\lambda}^{\tau}) \\ &\leqslant \sum_{\tau=1}^{T} \max_{\mathbf{y} \in \mathcal{R}} E^{\tau}(\mathbf{y}, \boldsymbol{\theta}, \boldsymbol{\lambda}^{\tau}) = \sum_{\tau=1}^{T} \max_{\mathbf{y} \in \mathcal{B}} E^{\tau}(\mathbf{y}, \boldsymbol{\theta}, \boldsymbol{\lambda}^{\tau}). \end{aligned} \tag{29}$$

Первое неравенство справедливо, так как мы расширили допустимое множество значений переменных. Во втором неравенстве воспользовались тем, что максимум суммы меньше либо равен сумме максимумов. Последнее равенство следует из того, что максимум в релаксированной задаче на древовидном графе достигается в целочисленной точке Получили верхнюю оценку решения исходной задачи, которая зависит от свободных переменных Λ. Можно ее уточнить, взяв минимум по свободным переменным. Заметим, что $\max_{\mathbf{y} \in \mathcal{B}} E^{\tau}(\mathbf{y}, \boldsymbol{\theta}, \boldsymbol{\lambda}^{\tau})$ является максимумом конечного числа линейных по $\boldsymbol{\lambda}^{\tau}$ функций, так называемой верхней огибающей семейства линейных функций. Но такая функция выпукла. Следовательно функция $\sum_{\tau=1}^{T} \max_{\mathbf{y} \in \mathcal{B}} E^{\tau}(\mathbf{y}, \boldsymbol{\theta}, \boldsymbol{\lambda}^{\tau})$ является выпуклой по Λ функцией, заданной на выпуклом множестве $\mathcal{C}$.

Выпишем полученную верхнюю оценку:

$$\max_{\mathbf{y} \in \mathcal{B}} E(\mathbf{y}, \boldsymbol{\theta}) \leqslant \min_{\Lambda \in \mathcal{C}} \sum_{\tau=1}^{T} \max_{\mathbf{y} \in \mathcal{B}} E^{\tau}(\mathbf{y}, \boldsymbol{\theta}, \boldsymbol{\lambda}^{\tau}).$$

3.1.2 Вывод, дополненный слабой функцией потерь

Теперь рассмотрим задачу (20):

$$\max_{\bar{\mathbf{y}}} \{\mathbf{w}^\intercal \Psi(\bar{\mathbf{y}}, \mathbf{x}) + \kappa(\bar{\mathbf{y}}, \mathbf{z})\}$$
$$\leqslant \min_{\boldsymbol{\mu}} \left\{ \max_{\bar{\mathbf{y}}} \left\{ \mathbf{w}^\intercal \Psi(\bar{\mathbf{y}}, \mathbf{x}) - \sum_{v \in \mathcal{V}} \mu_v(\bar{y}_v) \right\} + \max_{\bar{\mathbf{y}}} \left\{ \kappa(\bar{\mathbf{y}}, \mathbf{z}) + \sum_{v \in \mathcal{V}} \mu_v(\bar{y}_v) \right\} \right\}. \tag{30}$$

Здесь опять воспользовались двойственным разложением, $\boldsymbol{\mu} \in \mathbb{R}^{|\mathcal{V}| \times K}$ — двойственные переменные. Заметим, что слагаемые $\mu_v(\bar{y}_v)$ относятся к унарным потенциалам. Для оценки сверху первого слагаемого в правой части выражения (30) можем воспользоваться двойственным разложением, как это делали для задачи (19). Более подробно рассмотрим построение верхней оценкой для второго слагаемого

$$\max_{\bar{\mathbf{y}}} \left\{ \kappa(\bar{\mathbf{y}}, \mathbf{z}) + \sum_{v \in \mathcal{V}} \mu_v(\bar{y}_v) \right\} =$$
$$\max_{\bar{\mathbf{y}}} \left\{ \sum_{k \notin \mathbf{z}} \sum_{v \in \mathcal{V}} [\bar{y}_v = k] - \sum_{k \in \mathbf{z}} c_k [\exists v \in \mathcal{V} \colon \bar{y}_v = k] + \sum_{v \in \mathcal{V}} \mu_v(\bar{y}_v) \right\} =$$
$$\max_{\bar{\mathbf{y}}} \left\{ \sum_{v \in \mathcal{V}} \left(\sum_{k \notin \mathbf{z}} [\bar{y}_v = k] + \mu_v(\bar{y}_v) \right) - \sum_{k \in \mathbf{z}} c_k [\exists v \in \mathcal{V} \colon \bar{y}_v = k] \right\}. \tag{31}$$

Здесь воспользовались видом (2.8) функции $\kappa(\bar{\mathbf{y}}, \mathbf{z})$. Получили, что в максимизируемом выражении присутствуют потенциалы порядков 1 и $|\mathcal{V}|$.

Потенциал порядка $|\mathcal{V}|$ штрафует наличие меток из множества $\mathbf{z}$. Можно перебрать все возможные подмножества множества $\mathbf{z}$, и для каждого подмножества решить задачу максимизации, в которой будут присутствовать потенциалы только первого порядка. Поэтому задача (31) эквивалентна следующей:

$$\max_{\mathbf{l} \in \mathcal{P}(\mathbf{z})} \left\{ \max_{\bar{\mathbf{y}} \in \mathcal{Y} \setminus L(\mathbf{l})} \sum_{k \notin \mathbf{z}} \sum_{v \in \mathcal{V}} ([\bar{y}_v = k] + \mu_v(\bar{y}_v)) - \sum_{k \in (\mathcal{K} \setminus \mathbf{l}) \cup \mathbf{z}} c_k \right\} =$$
$$\max_{\mathbf{l} \in \mathcal{P}(\mathbf{z})} \left\{ \sum_{v \in \mathcal{V}} \max_{\bar{y}_v \in \mathcal{K} \setminus \mathbf{l}} \left(\sum_{k \notin \mathbf{z}} [\bar{y}_v = k] + \mu_v(\bar{y}_v) \right) - \sum_{k \in (\mathcal{K} \setminus \mathbf{l}) \cup \mathbf{z}} c_k \right\}, \tag{32}$$

где $\mathcal{P}(\mathbf{z})$ — множество всех подмножеств множества $\mathbf{z}$, а внутренняя максимизация ведется по множеству $\mathcal{Y} \setminus L(\mathbf{l})$, задающему все разметки, в которых отсутствуют метки из множества $\mathbf{l}$. Заметим, что внешняя максимизация осуществляется при помощи полного перебора множества $\mathcal{P}(\mathbf{z})$. Как правило, множество $\mathbf{z}$ не очень большое

(на изображении обычно представлены объекты не более чем 10 классов), поэтому перебор можно осуществить за разумное время. Внутренняя же подзадача распадается на простые задачи максимизации по одномерным переменным, принимающим не более K значений. На каждой итерации перебора второе слагаемое фиксируется, и решается $|\mathcal{V}|$ одномерных задач оптимизации. В конце перебора выбирается лучшее из $2^{|\mathbf{z}|}$ решений.

3.1.3 Верхняя оценка

Введем обозначения:

$$E_j^\tau(\bar{\mathbf{y}}^\tau, \mathbf{x}^j, \mathbf{y}^j, \mathbf{w}, \boldsymbol{\lambda}^{j,\tau}) = \sum_{v \in \mathcal{V}_\tau^j} \left(\frac{1}{|\mathcal{I}_v^j|} (\mathbf{w}^\intercal \psi(\bar{y}_v^\tau, \mathbf{x}_v^j) + [\bar{y}_v^\tau = y_v^j]) + \lambda_v^{j,\tau}(\bar{y}_v^\tau) \right)$$
$$+ \sum_{(u,v) \in \mathcal{E}_\tau^j} \mathbf{w}^\intercal \psi(\bar{y}_u^\tau, \bar{y}_v^\tau, \mathbf{x}_{uv}^j), \tag{33}$$

$$E_i^\tau(\bar{\mathbf{y}}^\tau, \mathbf{x}^i, \mathbf{w}, \boldsymbol{\lambda}^{i,\tau}, \boldsymbol{\mu}^i) = \sum_{v \in \mathcal{V}_\tau^i} \left(\frac{1}{|\mathcal{I}_v^i|} (\mathbf{w}^\intercal \psi(\bar{y}_v^\tau, \mathbf{x}_v^i) - \mu_v^i(\bar{y}_v^\tau)) + \lambda_v^{i,\tau}(\bar{y}_v^\tau) \right)$$
$$+ \sum_{(u,v) \in \mathcal{E}_\tau^i} \mathbf{w}^\intercal \psi(\bar{y}_u^\tau, \bar{y}_v^\tau, \mathbf{x}_{uv}^i), \tag{34}$$

$$F_i(\bar{\mathbf{y}}, \mathbf{z}^i, \boldsymbol{\mu}^i) = \sum_{v \in \mathcal{V}^i} \left(\sum_{k \notin \mathbf{z}^i} [\bar{y}_v = k] + \mu_v^i(\bar{y}_v) \right) - \sum_{k \in \mathbf{z}^i} c_k [\exists v \in \mathcal{V}^i \colon \bar{y}_v = k]. \tag{35}$$

Используя результаты предыдущих подразделов, выпишем верхнюю оценку $\mathcal{L}_u$ функционала $\mathcal{L}$ (15):

$$\mathcal{L}_u(\mathbf{w}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = \frac{1}{2}\|\mathbf{w}\|^2$$
$$+C \sum_{j=1}^{J} \left(\sum_{\tau=1}^{T_j} \max_{\bar{\mathbf{y}}^\tau} E_j^\tau(\bar{\mathbf{y}}^\tau, \mathbf{x}^j, \mathbf{y}^j, \mathbf{w}, \boldsymbol{\lambda}^{j,\tau}) - \mathbf{w}^\intercal \Psi(\mathbf{y}^j, \mathbf{x}^j) \right)$$
$$+\alpha C \sum_{i=J+1}^{J+I} \left(\sum_{\tau=1}^{T_i} \max_{\bar{\mathbf{y}}^\tau} E_i^\tau(\bar{\mathbf{y}}^\tau, \mathbf{x}^i, \mathbf{w}, \boldsymbol{\lambda}^{i,\tau}, \boldsymbol{\mu}^i) + \max_{\bar{\mathbf{y}}} F_i(\bar{\mathbf{y}}, \mathbf{z}^i, \boldsymbol{\mu}^i) \right)$$
$$-\alpha C \sum_{i=J+1}^{J+I} \max_{\mathbf{y} \in L(\mathbf{z}^i)} \mathbf{w}^\intercal \Psi(\mathbf{y}, \mathbf{x}^i). \tag{36}$$

Здесь

$$\boldsymbol{\mu} = \{\boldsymbol{\mu}^i\}_{i=J+1}^{J+I},$$
$$\boldsymbol{\lambda} = \{\boldsymbol{\lambda}^{j,\tau}\}_{j=1,\tau=1}^{J,T_j} \cup \{\boldsymbol{\lambda}^{i,\tau}\}_{i=J+1,\tau=1}^{J+I,T_i}.$$

При этом переменные $\boldsymbol{\lambda}^i, \boldsymbol{\lambda}^j$ должны удовлетворять ограничениям (25), запишем это как $\boldsymbol{\lambda} \in \mathcal{C}$. Таким образом доказали следующее

Утверждение 1. Функция $\mathcal{L}_u(\mathbf{w}, \boldsymbol{\lambda}, \boldsymbol{\mu})$ является верхней оценкой на функцию $\mathcal{L}(\mathbf{w})$ для любых $\mathbf{w}, \boldsymbol{\mu}$ и для любых $\boldsymbol{\lambda} \in \mathcal{C}$, и справедливо следующее неравенство:

$$\mathcal{L}(\mathbf{w}) \leqslant \min_{\boldsymbol{\lambda} \in \mathcal{C}, \boldsymbol{\mu}} \mathcal{L}_u(\mathbf{w}, \boldsymbol{\lambda}, \boldsymbol{\mu}), \forall \mathbf{w}. \tag{37}$$

Слагаемые $\max_{\bar{\mathbf{y}}^\tau} E_j^\tau(\bar{\mathbf{y}}^\tau, \mathbf{x}^j, \mathbf{y}^j, \mathbf{w}, \boldsymbol{\lambda}^{j,\tau})$ являются выпуклыми по переменным $\mathbf{w}$ и $\boldsymbol{\lambda}^{j,\tau}$ как максимумы конечного числа линейных функций (так называемая верхняя огибающая семейства линейных функций). Слагаемые $\max_{\bar{\mathbf{y}}^\tau} E_i^\tau(\bar{\mathbf{y}}^\tau, \mathbf{x}^i, \mathbf{w}, \boldsymbol{\lambda}^{i,\tau}, \boldsymbol{\mu}^i)$ аналогично являются выпуклыми по переменным $\mathbf{w}, \boldsymbol{\lambda}^{i,\tau}, \boldsymbol{\mu}^i$. Однако все они не являются дифференцируемыми (например, в точках излома верхней огибающей). То же самое можно сказать и про слагаемые $\max_{\bar{\mathbf{y}}} F_i(\bar{\mathbf{y}}, \mathbf{z}^i, \boldsymbol{\mu}^i)$. Слагаемое $\frac{1}{2}\|\mathbf{w}\|^2$ также является выпуклым. Слагаемые $\max_{\mathbf{y} \in L(\mathbf{z}^i)} \mathbf{w}^\intercal \Psi(\mathbf{y}, \mathbf{x}^i)$ являются выпуклыми по $\mathbf{w}$, однако они входят в функционал со знаком «минус». При фиксированных решениях задач

$$\max_{\mathbf{y} \in L(\mathbf{z}^i)} \mathbf{w}^\intercal \Psi(\mathbf{y}, \mathbf{x}^i)$$

последняя сумма в функционале (36) линейна по $\mathbf{w}$, следовательно выпукла и вогнута по $\mathbf{w}$, тогда функционал (36) является выпуклым по переменным $\mathbf{w}, \boldsymbol{\lambda}, \boldsymbol{\mu}$, как сумма выпуклых функций. В общем же случае получили, что функционал $\mathcal{L}_u(\mathbf{w}, \boldsymbol{\lambda}, \boldsymbol{\mu})$ является суммой выпуклого и вогнутого слагаемых.

3.2 Метод оптимизации

Рассмотрим алгоритм минимизации функционала $\mathcal{L}_u(\mathbf{w}, \boldsymbol{\lambda}, \boldsymbol{\mu})$ при ограничениях (25) на переменные $\{\boldsymbol{\lambda}\}$. Так как функционал не является дифференцируемым, будем использовать субградиентный спуск по переменным $\mathbf{w}$ и $\boldsymbol{\mu}$ и метод проекции субградиента для оптимизации по переменным $\boldsymbol{\lambda}$. Сначала приведем формулы для обновления оптимизируемых переменных, затем выпишем итоговый алгоритм.

3.2.1 Переменные w

Выпишем субградиент по $\mathbf{w}$:

$$d\mathbf{w} = \mathbf{w} + C\sum_{j=1}^{J}\left(\sum_{\tau=1}^{T_j}\left(\sum_{v\in\mathcal{V}_\tau^j}\frac{1}{|\mathcal{I}_v^j|}\psi(\hat{y}_v^{j,\tau},\mathbf{x}_v^j) + \sum_{(u,v)\in\mathcal{E}_\tau^j}\psi(\hat{y}_u^{j,\tau},\hat{y}_v^{j,\tau},\mathbf{x}_{uv}^j)\right) - \Psi(\mathbf{y}^j,\mathbf{x}^j)\right)$$
$$+\alpha C\sum_{i=J+1}^{J+I}\sum_{\tau=1}^{T_i}\left(\sum_{v\in\mathcal{V}_\tau^i}\frac{1}{|\mathcal{I}_v^i|}\psi(\hat{y}_v^{i,\tau},\mathbf{x}_v^i) + \sum_{(u,v)\in\mathcal{E}_\tau^i}\psi(\hat{y}_u^{i,\tau},\hat{y}_v^{i,\tau},\mathbf{x}_{uv}^i)\right) - \alpha C\sum_{i=J+1}^{J+I}\Psi(\hat{\mathbf{y}}^i,\mathbf{x}^i), \tag{38}$$

где $\hat{y}_v^{j,\tau}$ — решения задач:

$$\max_{\bar{\mathbf{y}}^\tau} E_j^\tau(\bar{\mathbf{y}}^\tau,\mathbf{x}^j,\mathbf{y}^j,\mathbf{w},\boldsymbol{\lambda}^{j,\tau}), \forall j\in\{1,\ldots,J\}, \forall\tau\in\{1,\ldots,T_j\}, \tag{39}$$

$\hat{y}_v^{i,\tau}$ — решения задач:

$$\max_{\bar{\mathbf{y}}^\tau} E_i^\tau(\bar{\mathbf{y}}^\tau,\mathbf{x}^i,\mathbf{w},\boldsymbol{\lambda}^{i,\tau},\boldsymbol{\mu}^i), \forall i\in\{J+1,\ldots,J+I\}, \forall\tau\in\{1,\ldots,T_i\}, \tag{40}$$

$\hat{\mathbf{y}}^i$ — решения задач:

$$\max_{\mathbf{y}\in L(\mathbf{z}^i)} \mathbf{w}^\intercal\Psi(\mathbf{y},\mathbf{x}^i), \forall i\in\{J+1,\ldots,J+I\}. \tag{41}$$

Выпишем шаг по субградиенту:

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t\cdot d\mathbf{w}. \tag{42}$$

Задачи (39) и (40) эффективно и точно решаются с помощью метода передачи сообщений (подробнее см. [1]). Задача (41), как уже было сказано выше, решается с помощью метода α-расширения [3].

3.2.2 Переменные $\boldsymbol{\mu}$

Субградиент по переменным $\boldsymbol{\mu}$:

$$d\mu_v^i(k) = -\alpha C\sum_{\tau=1}^{T_i}\frac{1}{|\mathcal{I}_v^i|}[\hat{y}_v^{i,\tau}=k] + \alpha C[\hat{y}_v^i = k], \tag{43}$$

где $\hat{y}^{i,\tau}$ — решения задач (40), $\hat{y}_v^i$ — решения задач:

$$\max_{\bar{\mathbf{y}}} F_i(\bar{\mathbf{y}},\mathbf{z}^i,\boldsymbol{\mu}^i). \tag{44}$$

Выпишем шаг по субградиенту:

$$\mu_{t+1,v}^i(k) = \mu_{t,v}^i(k) - \eta_t\cdot d\mu_v^i(k). \tag{45}$$

Задача (44) решается методом, рассмотренным в разделе 3.1.2.

3.2.3 Переменные $\boldsymbol{\lambda}$

Субградиент по переменным $\boldsymbol{\lambda}$:

$$d\lambda_v^{j,\tau}(k) = [\hat{y}_v^{j,\tau} = k], \tag{46}$$

$$d\lambda_v^{i,\tau}(k) = [\hat{y}_v^{i,\tau} = k], \tag{47}$$

где $\hat{y}_v^{j,\tau}$ — решения задач (39), $\hat{y}_v^{i,\tau}$ — решения задач (40). Эти переменные должны удовлетворять следующим ограничениям:

$$\sum_{\tau \in \mathcal{I}_v^j} \lambda_v^{j,\tau}(k) = 0, \tag{48}$$

$$\sum_{\tau \in \mathcal{I}_v^i} \lambda_v^{i,\tau}(k) = 0. \tag{49}$$

Поэтому после обновления $\boldsymbol{\lambda}_{t+1}^{j,\tau} = \boldsymbol{\lambda}_t^{j,\tau} - \eta_t \cdot d\boldsymbol{\lambda}^{j,\tau}$ необходимо спроецировать полученные переменные на множество, определяемое ограничениями (48). Такая проекция эквивалентна вычитанию среднего $\left(\sum_{\tau \in \mathcal{I}_v^j} \lambda_v^{j,\tau}(k)\right) / |\mathcal{I}_v^j|$ из $\lambda_v^{j,\tau}(k)$ (аналогично для переменных $\boldsymbol{\lambda}^{i,\tau}$ и ограничений (49)). Учитывая это, выпишем формулы обновления переменных $\boldsymbol{\lambda}$, включающие в себя субградиентный шаг и проекцию:

$$\lambda_{t+1,v}^{j,\tau}(k) = \lambda_{t,v}^{j,\tau}(k) - \eta_t \left([\hat{y}_v^{j,\tau} = k] - \frac{1}{|\mathcal{I}_v^j|} \sum_{\tau \in \mathcal{I}_v^j} [\hat{y}_v^{j,\tau} = k] \right), \tag{50}$$

$$\lambda_{t+1,v}^{i,\tau}(k) = \lambda_{t,v}^{i,\tau}(k) - \eta_t \left([\hat{y}_v^{i,\tau} = k] - \frac{1}{|\mathcal{I}_v^i|} \sum_{\tau \in \mathcal{I}_v^i} [\hat{y}_v^{i,\tau} = k] \right). \tag{51}$$

Шаг η_t выбирается по правилу:

$$\eta_t = \frac{\gamma}{t},$$

где γ — некоторая константа.

3.2.4 Алгоритм

Общая схема минимизации функционала 36 приведена в алгоритме 1. Дадим некоторые пояснения. На вход, помимо обучающей выборки, подается параметр ε для проверки сходимости, γ — параметр шага, N — максимальное число итераций. Декомпозиция на деревья производится с помощью алгоритма монотонных цепочек,

Исходные параметры: $\{(\mathbf{x}^j, \mathbf{y}^j)\}_{j=1}^{J}, \{(\mathbf{x}^i, \mathbf{z}^i)\}_{i=J+1}^{J+I}, \varepsilon, N, \gamma$

Проинициализировать нулем оптимизируемые переменные $\boldsymbol{\lambda}, \boldsymbol{\mu}, \mathbf{w}$;

Задать декомпозицию на деревья $G_\tau^i, G_\tau^j, \forall j \in \{1, \ldots, J\}, i \in \{J+1, \ldots, J+I\}$;

$t = 0$;

до тех пор, пока $t < N$ **выполнять**

Получить решения задач (39)–(41), (44);

Посчитать переменные $\mathbf{w}_{t+1}, \boldsymbol{\mu}_{t+1}, \boldsymbol{\lambda}_{t+1}$ по формулам (42), (45), (50), (51);

Проверить сходимость;

$\eta_t = \frac{\gamma}{t}$;

$t = t + 1$;

конец цикла

Алгоритм 1: Псевдокод основного алгоритма.

который описан в работе [8]. Проверка сходимости определяется по изменению значения верхней оценки: алгоритм завершает работу, если

$$|\mathcal{L}_u(\mathbf{w}_{t+1}, \boldsymbol{\lambda}_{t+1}, \boldsymbol{\mu}_{t+1}) - \mathcal{L}_u(\mathbf{w}_t, \boldsymbol{\lambda}_t, \boldsymbol{\mu}_t)| < \varepsilon.$$

4 Эксперименты

4.1 Описание данных

Для экспериментов использовался синтетический набор данных и набор данных MSRC-23[1].

4.1.1 Синтетический набор данных

Был сгенерирован синтетический набор изображений размера 20×20. Каждый пиксель принадлежит одному из 10 классов, у него есть 10 признаков, каждый признак — зашумленный индикатор принадлежности соответствующему классу. Соседние пиксели тяготеют к одинаковым меткам классов. Каждое изображение содержит

[1] http://research.microsoft.com/en-us/projects/objectclassrecognition/

пиксели не более 3-х различных классов, поэтому слабая аннотация в виде множества классов на изображении задает довольно сильное ограничение на допустимые разметки.

Для генерации использовалось сэмплирование Гиббса из Марковского случайного поля на четырех-связной решетке:

$$P(\mathbf{x}, \mathbf{y}) = \frac{1}{Z} \prod_{v \in \mathcal{V}} \phi(\mathbf{x}_v, y_v) \prod_{(v,u) \in \mathcal{E}} \phi(y_v, y_u),$$

где $\phi(\mathbf{x}_v, y_v) = P(x_v^k = \xi \mid y_v = k), \xi \sim \text{Beta}(a, 1)$ для релевантного признака, $\phi(\mathbf{x}_v, y_v) = P(x_v^k = \eta \mid y_v \neq k), \eta \sim \text{Uniform}(0, 1)$ — для остальных и сглаживающий потенциал $\phi(y_v, y_u) = \exp(-T[y_v \neq y_u])$. Эти же величины являются признаками объектов. Использовались параметры $a = 2$ и $T = 4$. В обучающей выборке используется 100 объектов, в тестовой — 400. Было сгенерировано 18 таких выборок для того, чтобы можно было оценить статистическую значимость результата.

4.1.2 Набор данных MSRC-23

Набор данных MSRC-23 содержит 276 изображений в обучающей выборке и 256 — в тестовой. В этом наборе представлены объекты 23 классов. Для экспериментов требовалось выделить обучающую подвыборку объектов с сильной разметкой. Мы стремились выбрать подвыборки, у которых распределение меток классов похоже на распределение у всей обучающей выборки.

Опишем используемые признаки. Было произведено разбиение на суперпиксели с помощью алгоритма *gPb* [4]. Используются следующие унарные признаки: гистограмма SIFT [13], полученная с помощью библиотеки VLFeat [16] с использованием словаря размера 512 и жесткой кластеризацией, гистограмма RGB со словарем размера 128 и жесткой кластеризацией. Была произведена L_2 нормировка признаков, затем они были отображены в пространство более высокой размерности, в котором скалярное произведение аппроксимирует χ^2-ядро в исходном пространстве [17] (таким образом размерность увеличилась в три раза). Парные потенциалы задаются на суперпикселях, имеющих общую границу. Используются следующие парные признаки: $\exp(-c_{uv}/10)$, $\exp(-c_{uv}/40)$, $\exp(-c_{uv}/100)$, 1. Здесь c_{uv} — это сила границы между суперпикселями u и v, полученная с помощью алгоритма *gPb*.

4.2 Оценка качества

Для оценки качества использовалось расстояние Хэмминга:

$$\Delta(\mathbf{y}, \bar{\mathbf{y}}) = \sum_{v \in \mathcal{V}} b_v [y_v \neq \bar{y}_v], \tag{52}$$

где $\sum_{v \in \mathcal{V}} b_v = 1$, таким образом $\Delta(\mathbf{y}, \bar{\mathbf{y}}) \in [0, 1]$. В простейшем случае $b_v = 1/|\mathcal{V}|$ для всех v. *Точностью* (англ. *accuracy*) назовем следующую величину:

$$S = \frac{1}{N} \sum_{i=1}^{N} (1 - \Delta(\mathbf{y}^i, \bar{\mathbf{y}}^i)), \tag{53}$$

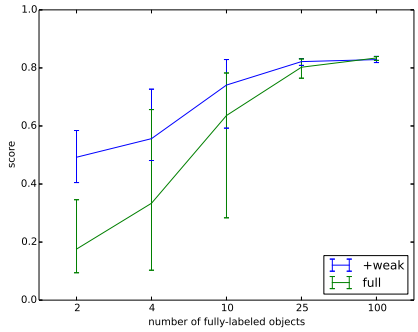
где N — размер тестовой выборки. Заметим, что $S \in [0, 1]$, и чем больше значение S, тем лучше. Значение $S = 1$ соответствует идеальной классификации.

4.3 Стандартный подход

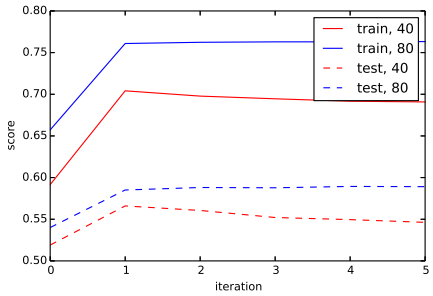
В этом разделе рассмотрим стандартный подход к решению задачи 3, основанный на выпукло-вогнутой процедуре, при этом в качестве методов оптимизации используется алгоритм α-расширения для вывода, дополненного функцией потерь, и алгоритм α-расширения со штрафами за метки для вывода, дополненного слабой функцией потерь. Для решения задачи структурного метода опорных векторов используется алгоритм отсекающей плоскости [7]. Исследовалась зависимость точности от числа полностью размеченных объектов в обучающей выборке, а также изменение точности в зависимости от итераций метода оптимизации. См. рис. 3.

На рис. 3a зеленой линией отмечена точность при использовании только полностью размеченных объектов, синей линией — с учетом слабоаннотированных объектов (точность измерялась на тестовой выборке). С ростом числа полностью размеченных объектов точность увеличивается, и, начиная с некоторого числа объектов, точность стабилизируется (в данном случае 25).

На рис. 3b штриховой линией показаны значения точности на тестовой выборке, непрерывной линией — на обучающей. Опять наблюдается увеличение точности при увеличении числа полностью размеченных объектов. Отметим, что наиболее значимое увеличение точности происходит после первой итерации выпукло-вогнутой процедуры, далее точность почти не изменяется.



(a) Зависимость точности от числа полностью размеченных объектов в обучающей выборке, синтетические данные, $C = 10, \alpha = 0.1$.



(b) Зависимость точности от номера итерации выпукло-вогнутой процедуры на наборе данных MSRC-23 при разном числе полностью размеченных объектов, $C = 100, \alpha = 0.1$.

Рис. 3: Зависимость точности от числа полностью размеченных объектов в обучающей выборке.

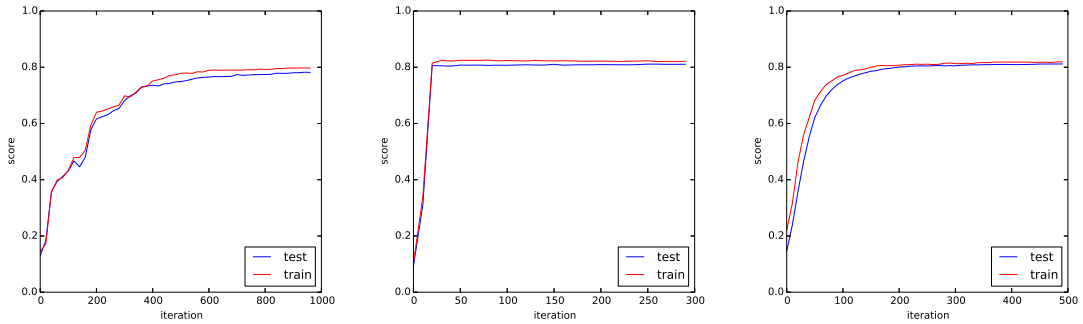
4.4 Релаксационный подход

4.4.1 Полностью размеченные данные

Исследуем релаксационный подход на выборке, состоящей только из полностью размеченных объектов. Используем модификацию алгоритма 1, в которой отсутствуют вычисления, связанные со слабоаннотированными объектами $\{(\mathbf{x}^i, \mathbf{z}^i)\}_{i=J+1}^{J+I}$.

Релаксационный метод сравнивается с двумя другими стандартными подходами: оптимизация функционала из задачи 2 субградиентным спуском, решение этой же задачи, методом Франка-Вольфа [2]. На рис. 4 показана точность методов на тестовой и обучающей выборках в зависимости от номера итерации на синтетическом наборе данных. Метод, основанный на релаксационном подходе, дает сравнимую со стандартными методами точность, однако сходится несколько медленнее.

На рис. 5 приведены графики точности в зависимости от номера итерации на наборе данных MSRC-23. Сравниваются метод Франка-Вольфа и метод, основанный на релаксационном подходе. На этом наборе данных последний метод показывает меньшую точность.



(a) Метод, основанный на релаксационном подходе.

(b) Субградиентный спуск.

(c) Метод Франка-Вольфа.

Рис. 4: Сравнение методов на синтетических данных, 100 полностью размеченных объектов, $C = 10$.

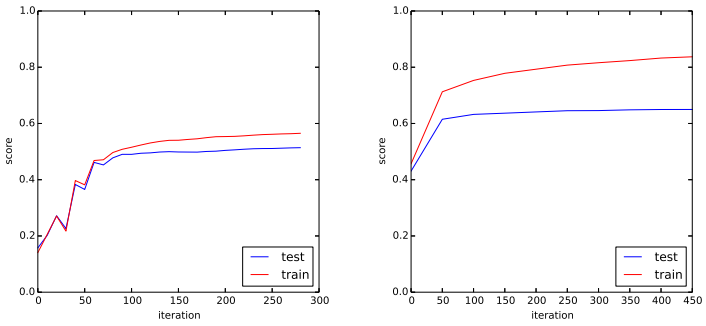
4.4.2 Использование слабой аннотации

Теперь рассмотрим работу релаксационного подхода при обучении с использованием слабоаннотированных данных.

Исследуется работа метода при использовании разных алгоритмов для решения оптимизационных задач (20) и (40).

1. На рис. 6a приводится результат работы алгоритма, в котором для решения задачи вывода, дополненного слабой функцией потерь (20), используется метод α-расширения со штрафами за метки. В этом случае не получается корректной верхней оценки.Этот метод показывает наилучшую точность. Это связано с тем, что метод α-расширения со штрафами за метки, хоть и не гарантирует получение верхней оценки, зачастую выдает решения очень близкие к оптимальным.

2. На рис. 6b приводится результат работы алгоритма, использующего двойственное разложение по переменным $\boldsymbol{\mu}$ в задаче (20); для решения задачи, возникающей в первом слагаемом (40) используется метод α-расширения, а задача (44) решается предложенным в разделе 3.1.2 методом. В этом случае также не получается корректной верхней оценки. Этот метод проигрывает по точности предыдущему около 5%.



(a) Метод, основанный на релаксационном подходе.

(b) Метод Франка-Вольфа.

Рис. 5: Сравнение методов на данных MSRC-23, 276 полностью размеченных объектов, $C = 100$.

3. На рис. 6c приводится результат работы алгоритма, который также использует двойственное разложение по переменным $\boldsymbol{\mu}$ в задаче (20), однако задача (40) решается методом субмодулярной релаксации [15]. В этом случае получается корректная верхняя оценка. Данный метод уступает первому методу по точности, проигрывая около 10%.

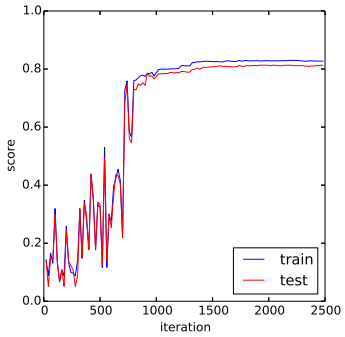
4. На рис. 6d используется алгоритм 1. Метод работает хуже всего. Это может быть связано с большим зазором между верхней оценкой и оптимизируемым функционалом.

4.4.3 Сравнительный эксперимент

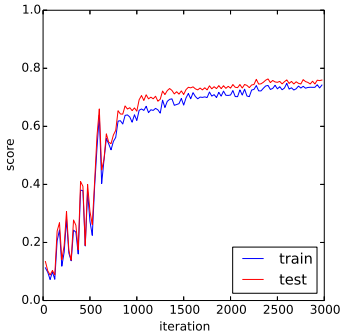
В таблице 1 приведены сравнительные результаты работы двух подходов. Подход, в котором для решения задачи вывода, дополненного функцией потерь, используется метод α-расширения («Подход с α-расширением») превосходит релаксационный подход по точности.

В таблице 2 приведены сравнительные результаты работы алгоритмов на синтетическом наборе данных с учетом слабой аннотации. Сравниваются три подхода:

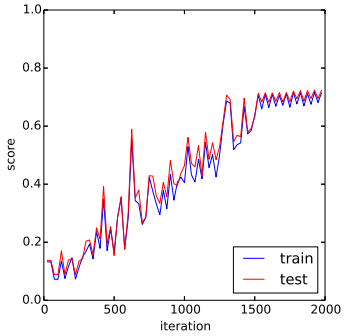
1. Релаксационный подход, в котором задача (40) решается методом субмодулярной релаксации [15],



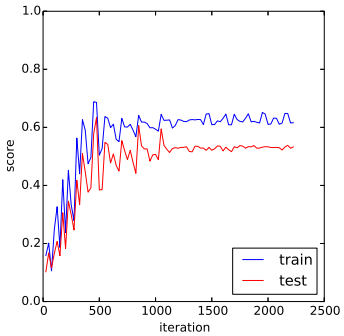
(a) Задача (20) решается методом α-расширения со штрафами за метки.



(b) Задача (40) решается методом α-расширения.



(c) Задача (40) решается методом субмодулярной релаксации.



(d) Алгоритм 1.

Рис. 6: Синтетические данные, 10 полностью размеченных объектов, 90 — слабоаннотированных, $C = 10, \alpha = 0.1$.

2. подход, в котором для решения задачи (19) используется релаксационный подход, а для решения задачи (20) используется алгоритм α-расширения («Релаксационный + α»),

3. подход, в котором для решения подзадач (19), (20) используется алгоритм α-расширения («$\alpha + \alpha$»).

Набор данных		Релаксационный подход	Подход с α-расширением
Синтетический (100)	Обучение	0.7979	0.8192
	Контроль	0.7832	0.8122
MSRC (276)	Обучение	0.5164	0.6518
	Контроль	0.5671	0.8430

Таблица 1: Точность методов на полностью размеченных выборках. В скобках указано число объектов в обучающей выборке.

	Релаксационный	Релаксационный + α	$\alpha + \alpha$
Обучение	0.7144	0.7423	0.8395
Контроль	0.7232	0.7595	0.8254

Таблица 2: Точность методов при обучении с учетом слабой аннотации на синтетическом наборе данных. См. пояснения в тексте.

5 Заключение

В данной работе было исследовано поведение верхней оценки функционала обобщенного структурного метода опорных векторов (15). Для этого были решены следующие задачи:

1. Построена верхняя оценка функционала (15), см. раздел (3.1),
2. разработан метод оптимизации построенной верхней оценки, см. раздел (3.2),
3. проведено экспериментальное сравнение релаксационного метода со стандартным подходом на двух наборах данных, см. раздел (4).

   В результате проведенных экспериментов было выяснено, что метод, основанный на релаксационном подходе, несмотря на свою теоретическую обоснованность, по точности проигрывает методу, где для решения всех задач используется метод α-расширения. Это может быть связано с тем, что, зачастую, метод α-расширения выдает решение близкое к оптимальному, а при использовании двойственного разложения может иметь место большой зазор, и получается слишком неплотная верхняя оценка.

Список литературы

[1] Bishop, C. M. Pattern recognition and machine learning / Christopher M Bishop et al. — springer New York, 2006. — Vol. 1.

[2] Block-coordinate frank-wolfe optimization for structural svms / Simon Lacoste-Julien, Martin Jaggi, Mark Schmidt, Patrick Pletscher // arXiv preprint arXiv:1207.4747. — 2012.

[3] Boykov, Y. Fast approximate energy minimization via graph cuts / Yuri Boykov, Olga Veksler, Ramin Zabih // Pattern Analysis and Machine Intelligence, IEEE Transactions on. — 2001. — Vol. 23, no. 11. — P. 1222–1239.

[4] Contour detection and hierarchical image segmentation / Pablo Arbelaez, Michael Maire, Charless Fowlkes, Jitendra Malik // Pattern Analysis and Machine Intelligence, IEEE Transactions on. — 2011. — Vol. 33, no. 5. — P. 898–916.

[5] Fast approximate energy minimization with label costs / Andrew Delong, Anton Osokin, Hossam N Isack, Yuri Boykov // International journal of computer vision. — 2012. — Vol. 96, no. 1. — P. 1–27.

[6] Finley, T. Training structural svms when exact inference is intractable / Thomas Finley, Thorsten Joachims // Proceedings of the 25th international conference on Machine learning / ACM. — 2008. — P. 304–311.

[7] Joachims, T. Cutting-plane training of structural svms / Thorsten Joachims, Thomas Finley, Chun-Nam John Yu // Machine Learning. — 2009. — Vol. 77, no. 1. — P. 27–59.

[8] Kolmogorov, V. Convergent tree-reweighted message passing for energy minimization / Vladimir Kolmogorov // Pattern Analysis and Machine Intelligence, IEEE Transactions on. — 2006. — Vol. 28, no. 10. — P. 1568–1583.

[9] Komodakis, N. Efficient training for pairwise or higher order crfs via dual decomposition / Nikos Komodakis // Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on / IEEE. — 2011. — P. 1841–1848.

[10] Komodakis, N. Mrf optimization via dual decomposition: Message-passing revisited / Nikos Komodakis, Nikos Paragios, Georgios Tziritas // Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on / IEEE. — 2007. — P. 1–8.

[11] Komodakis, N. Mrf energy minimization and beyond via dual decomposition / Nikos Komodakis, Nikos Paragios, Georgios Tziritas // Pattern Analysis and Machine Intelligence, IEEE Transactions on. — 2011. — Vol. 33, no. 3. — P. 531–552.

[12] Large margin methods for structured and interdependent output variables / Ioannis Tsochantaridis, Thorsten Joachims, Thomas Hofmann, Yasemin Altun // Journal of Machine Learning Research. — 2005. — P. 1453–1484.

[13] Lowe, D. G. Distinctive image features from scale-invariant keypoints / David G Lowe // International journal of computer vision. — 2004. — Vol. 60, no. 2. — P. 91–110.

[14] Müller, A. C. Pystruct-learning structured prediction in python / Andreas C Müller, Sven Behnke // Journal of Machine Learning Research. — 2013. — Vol. 1. — P. 1–1.

[15] Osokin, A. Submodular decomposition framework for inference in associative markov networks with global constraints / Anton Osokin, Dmitry Vetrov, Vladimir Kolmogorov // Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on / IEEE. — 2011. — P. 1889–1896.

[16] Vedaldi, A. Vlfeat: An open and portable library of computer vision algorithms / Andrea Vedaldi, Brian Fulkerson // Proceedings of the international conference on Multimedia / ACM. — 2010. — P. 1469–1472.

[17] Vedaldi, A. Efficient additive kernels via explicit feature maps / Andrea Vedaldi, Andrew Zisserman // Pattern Analysis and Machine Intelligence, IEEE Transactions on. — 2012. — Vol. 34, no. 3. — P. 480–492.

[18] Yu, C.-N. J. Learning structural svms with latent variables / Chun-Nam John Yu, Thorsten Joachims // Proceedings of the 26th Annual International Conference on Machine Learning / ACM. — 2009. — P. 1169–1176.

[19] Yuille, A. L. The concave-convex procedure / Alan L Yuille, Anand Rangarajan // Neural Computation. — 2003. — Vol. 15, no. 4. — P. 915–936.

[20] Ветров, Обучение структурного метода опорных векторов со слабым учителем в задачах сегментации изображений / Д.П. Ветров, Р.В. Шаповалов, А.А. Осокин // Доклады 9-й Международной конференции «Интеллектуализация обработки информации». — 2012.