Решение задачи MAP для марковской случайной сети

Новиков А. В. *
МГУ, ВМиК, каф. ММП

25 мая 2013 г.

*Научный руководитель: *Ветров Д. П.*, куратор: *Осокин А.*

Содержание

1 Введение

Марковские случайные поля (MRF) — это популярный подход к решению задач анализа данных. Чаще всего их используют в компьютерном зрении, где область применения простирается от устранения шума и сегментации изображений до стерео–реконструкции, распознавания образов и редактирования изображений. Одна из самых важных задач, возникающих при использовании MRF — максимизация апостериорной вероятности (MAP) [2]. Для большинства приложений задача MAP является NP–трудной.

В данной работе проводится обзор и сравнение существующих state–of–the–art подходов к решению этой задачи.

2 Данные

Чтобы получить воспроизводимые результаты мы использовали данные, опубликованные в работе [10]. Это задачи стерео–реконструкции с MRF типа решетка и парными потенциала Поттса. В них количество переменных равно количеству пикселей на изображениях, а количество меток K соответствует разрешению по оси Z у восстановленной карты глубины. В наших экспериментах участвовали следующие стерео–пары:

Название	Количество переменных	Количество меток
Tsukuba	110592	16
Venus	166222	20
Cones	168750	60

3 Обозначения

Марковская случайная сеть задана графом $G = (V, E)$. Переменные прямой задачи обозначаются $x_a \in \{1, \dots, K\}$ и индексируются вершиной $a \in V$. $X = \{x_a\}_{a \in V}$. Двойственная функция обозначается как $f(\Lambda)$. $\mathcal{L}$ — это множество ограничений на Λ, а g^k — проекция субградиента $f(\Lambda)$ на множество $\mathcal{L}$ (на k–ой итерации).
$P_C(\cdot)$ используется для обозначения Евклидовой проекции на множество C (т. е. $P_C(x_0) = \arg\min_{x \in C} \|x - x_0\|_2$).

4 Постановка задачи

На MRF заданной графом $G = (V, E)$ ставится задача максимизации апостериорной вероятности:

$$P(x \mid z) \propto \prod_{a \in V} \psi(z_a \mid x_a) \prod_{(a,b) \in E} \psi_{ab}(x_a, x_b) \to \max_X$$

От неё переходят к задаче минимизации минус логарифма вероятности, который называют *энергией*:

$$E(x) = \sum_{a \in V} \varphi_a(x_a) + \sum_{(a,b) \in E} \varphi_{ab}(x_a, x_b) \to \min_X \tag{1}$$

Унарный потенциал $\varphi_a(p)$ отражает стоимость присваивания $x_a = p$, а парный потенциал $\varphi_{ab}(p, q)$ — стоимость присваивания $x_a = p, x_b = q$.
В общем случае (1) — NP–трудная задача дискретной оптимизации, но существует частные случаи для которых она решается эффективно [7].

5 Обзор методов

Одним из хорошо зарекомендовавших себя подходов к решению данной задачи считается применение методов основанных на совершении шагов (например α–расширение) [5]. Мы будем сравнивать α–расширение с TRW–S [8] и методами основанными на двойственном разложении, в которых задача минимизации исходной энергии сводится к максимизации двойственной энергии (это функция в отличие от исходной энергии является вогнутой и кусочно–линейной). Алгоритмы этой группы отличает конкретный метод максимизации.

Полный список методов участвующих в сравнении:

- α–расширение[1] [1] [3] [4]
- TRW–S[2] [8]
- Субградиентный подъём [11]
- Методы на основе *пучков* (bundle methods) [13]
- L–BFGS

6 Двойственное разложение

Заменим каждую K–значную переменную x_a на набор бинарных переменных $\{y_{a,1} \dots y_{a,K}\}$: $y_{a,p} = 1 \Leftrightarrow x_a = p$. Аналогично для каждого ребра графа $(a, b) \in E$ введем бинарные переменные $y_{ab,11}, y_{ab,12} \dots, y_{ab,KK}$: $y_{ab,pq} = 1 \Leftrightarrow x_a = p, x_b = q$. Обозначим $\theta_{a,p} = \varphi_a(p)$, $\theta_{ab,pq} = \varphi_{ab}(p, q)$.
В новых обозначениях энергия становится линейной функцией:

$$E(Y, \Theta) = \sum_{a \in V} \sum_{p=1}^{K} \theta_{a,p} y_{a,p} + \sum_{(a,b) \in E} \sum_{p,q=1,1}^{K} \theta_{ab,pq} y_{ab,pq} \to \min_{Y \in \mathcal{M}} \tag{2}$$

[1] http://www.csd.uwo.ca/faculty/olga/software.html
[2] http://pub.ist.ac.at/ vnk/papers/TRW-S.html

$$\mathcal{M} = \left\{ Y \mid y_{a,p}, y_{ab,pq} \in \{0,\ 1\}, \sum_{p=1}^{K} y_{a,p} = 1, \right.$$
$$\left. \sum_{p=1}^{K} y_{ab,pq} = y_{b,q}, \sum_{q=1}^{K} y_{ab,pq} = y_{a,p} \right\}$$

Заменим множество ограничений $\mathcal{M}$ на более широкое $\mathcal{R}$ (данный прием называется LP–релаксацей):

$$\mathcal{R} = \left\{ Y \mid y_{a,p}, y_{ab,pq} \in [\mathbf{0},\ \mathbf{1}], \sum_{p=1}^{K} y_{a,p} = 1, \right.$$
$$\left. \sum_{p=1}^{K} y_{ab,pq} = y_{b,q}, \sum_{q=1}^{K} y_{ab,pq} = y_{a,p} \right\}$$

Мы получаем оценку снизу:

$$\min_{y \in \mathcal{M}} E(Y, \Theta) \geq \min_{Y \in \mathcal{R}} E(Y, \Theta). \tag{3}$$

Если граф G является деревом, — в (3) достигается равенство.

Разобьем граф на деревья $\{D^t\}_{t=1}^T$ так, чтобы каждая вершина и каждое ребро G входили ходя бы в одно дерево. Обозначим за n_a число деревьев, включающих вершину a.

$$\theta_{a,p}^t = \begin{cases} \frac{\theta_{a,p}}{n_a}, & a \in D^t \\ 0, & a \notin D^t \end{cases}$$

Аналогично, n_{ab} — число деревьев включающих ребро (a, b),

$$\theta_{ab,pq}^t = \begin{cases} \frac{\theta_{ab,pq}}{n_{ab}}, & (a,b) \in D^t \\ 0, & (a,b) \notin D^t \end{cases}$$

Таким образом, для каждого дерева D^t мы определили массив переменных Θ^t, причем

$$\Theta = \sum_{t=1}^{T} \Theta^t,$$

$$E(Y, \Theta) = \sum_{t=1}^{T} E(Y, \Theta^t).$$

Введем дополнительные переменные $\Lambda = \{\Lambda^t\}_{t=1}^T = \{\{\lambda_{a,p}^t\}, \{\lambda_{ab,pq}^t\}\}_{t=1}^T \in \mathcal{L}$, где $\mathcal{L}$ задается ограничениями:

$$\sum_{t=1}^{T} \lambda_{a,p}^t = 0, \forall a, p,$$

$$\sum_{t=1}^{T} \lambda^t_{ab,pq} = 0, \forall (a,b), p, q.$$

Собирая всё вместе:

$$\min_{Y \in \mathcal{M}} E(Y \mid \Theta) \geq \min_{Y \in \mathcal{R}} E(Y \mid \Theta) =$$
$$= \min_{Y \in \mathcal{R}} E(Y \mid \Theta) + \sum_{t=1}^{T} \left[\sum_{a \in V} \sum_{p=1}^{K} \lambda^t_{a,p} y_{a,p} + \sum_{(a,b) \in E} \sum_{p,q=1}^{K} \lambda^t_{ab,pq} y_{ab,pq} \right] =$$
$$= \min_{Y \in \mathcal{R}} E(Y \mid \Theta + \Lambda) \geq \sum_{t=1}^{T} \min_{Y \in \mathcal{R}} E(Y \mid \Theta^t + \Lambda^t)$$

В каждом слагаемом у нас LP–релаксация задачи (2) для дерева, а значит:

$$\sum_{t=1}^{T} \min_{Y \in \mathcal{R}} E(Y \mid \Theta^t + \Lambda^t) = \sum_{t=1}^{T} \min_{Y \in \mathcal{M}} E(Y \mid \Theta^t + \Lambda^t) = f(\Lambda), \tag{4}$$

где $f(\Lambda)$ называют *двойственной функцией*.

Заметим, что $\min_{Y \in \mathcal{M}} E(Y \mid \Theta^t + \Lambda^t)$ является минимум конечного (хотя и очень большого) числа линейных по Λ^t функций, т. е. вогнутой функцией.

Для применения методов максимизации функции $f(\Lambda)$ нам понадобятся проекции её субградиентов на множество $\mathcal{L}$:

$$P_{\mathcal{L}} \left(\frac{\partial}{\partial \lambda^t_{a,p}} f(\Lambda) \right) = \widehat{y}^t_{a,p} - \frac{\sum_{\{t' \mid a \in D^{t'}\}} \widehat{y}^{t'}_{a,p}}{n_a} \tag{5}$$

, где $\widehat{y}^t_{ap} = arg \min_{Y \in \mathcal{M}} E(Y \mid \Theta^t + \Lambda^t)$, то есть решение внутренней задачи минимизации на дереве. Аналогично,

$$P_{\mathcal{L}} \left(\frac{\partial}{\partial \lambda^t_{ab,pq}} f(\Lambda) \right) = \widehat{y}^t_{ab,pq} - \frac{\sum_{\{t' \mid (a,b) \in D^{t'}\}} \widehat{y}^{t'}_{ab,pq}}{n_{ab}} \tag{6}$$

, где $\widehat{y}^t_{ab,pq} = arg \min_{Y \in \mathcal{M}} E(Y \mid \Theta^t + \Lambda^t)$.

7 Методы оптимизации двойственной функции

На шаге k обозначим за g^k проекцию субградиента $f(\Lambda^k)$:

$$g^k = P_{\mathcal{L}} \left(\frac{\partial}{\partial \Lambda} f(\Lambda) \right) \Big|_{\Lambda = \Lambda^k} \tag{7}$$

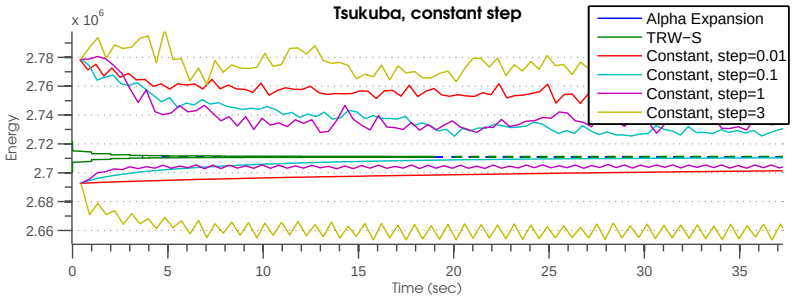


Рис. 1: Субградиентный подъём с константным шагом для различных констант.

Субградиентный подъём

В основе метода субградиентного подъёма лежит идея на каждом шаге идти в сторону проекции субградиента $f(\Lambda)$:

$$\Lambda^{k+1} = \Lambda^k + \alpha^k \cdot g^k \tag{8}$$

Результаты работы данного метода зависят от выбора последовательности шагов α^k. Мы использовали константный, адаптивный шаг, неточную одномерную оптимизацию по величине шага, а так же точную одномерную оптимизацию.

Как и ожидалось, субградиентный подъём с константным шагом сходится плохо (рис. 1).

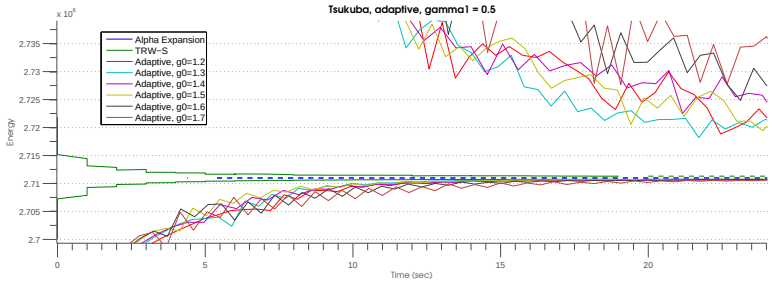
В случае адаптивного шага использовалась формула предложенная Комодакисом [6]:

$$\alpha^k = \frac{Approx^k - Dual^k}{\|g^k\|^2} \tag{9}$$

$Dual^k$ — это текущее значение двойственной функции, а $Approx^k$ — оценка оптимума двойственной функции, вычисляемая по формуле:
$Approx^k = BestDual^k + \delta^k$,
$BestDual^k = \max_{t\in\{1\dots k\}} f(\Lambda^t)$ — это лучшее на данный момент значение двойственной функции,

$$\delta^{k+1} = \begin{cases} \gamma_0\delta^k, & Dual^k > Dual^{k-1}, \\ max(\gamma_1\delta^k, \epsilon) & Dual^k \leqslant Dual^{k-1}. \end{cases} \tag{10}$$

γ_0, γ_1, ϵ — параметры метода, выбранные нами эмпирически (рис. 2, 8).
$\gamma_0 = 1.4$
$\gamma_1 = 0.5$
$\epsilon^k = \frac{1}{k}$



(a) Стерео–пара tsukuba.

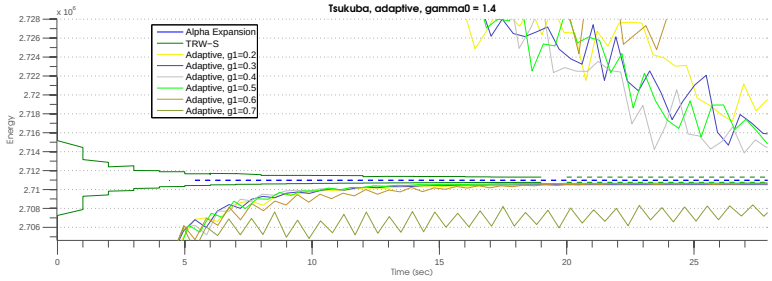


Рис. 2: Подбор параметров адаптивного субградиетного подъёма.

Одномерная оптимизация

Несмотря на то, что профиль[3] двойственной функции является кусочно линейным, он очень близок к гладкому (рис. 3), так что методы неточной одномерной оптимизации кажутся довольно перспективным направлением работы (мы верим, что данную задачу почти гладкой выпуклой одномерной оптимизации можно решить эффективно). Нами были опробованы метод Флетчера и «backtracking» [14].
Тем не менее мы выяснили что одномерная оптимизация не приносит ожидаемой пользы (рис. 4, видно что даже если использовать точную оптимизацию и не учитывать потраченное на неё время, результат получается хуже чем при использовании адаптивного подхода).

Методы на основе *пучков* («bundle methods»)

Основная идея методов этого класса — ограничить двойственную функцию $f(\Lambda)$ сверху с помощью вогнутой кусочно–линейной функции $\hat{f}(\Lambda)$ и дальше оптимизировать (уточняя на каждом шаге) именно $\hat{f}(\Lambda)$. Эта функция строится по последовательности точек $\{\Lambda^k\}$, значений в этих точках $\{f(\Lambda^k)\}$ и субградиентов $g^k \in \partial f(\Lambda^k)$ так, что $f(\Lambda) \leq \hat{f}(\Lambda)$ и $f(\Lambda^k) = \hat{f}(\Lambda^k)$. Вместе

[3] профилем функции мы будем называть график значения функции вдоль определенного направления.

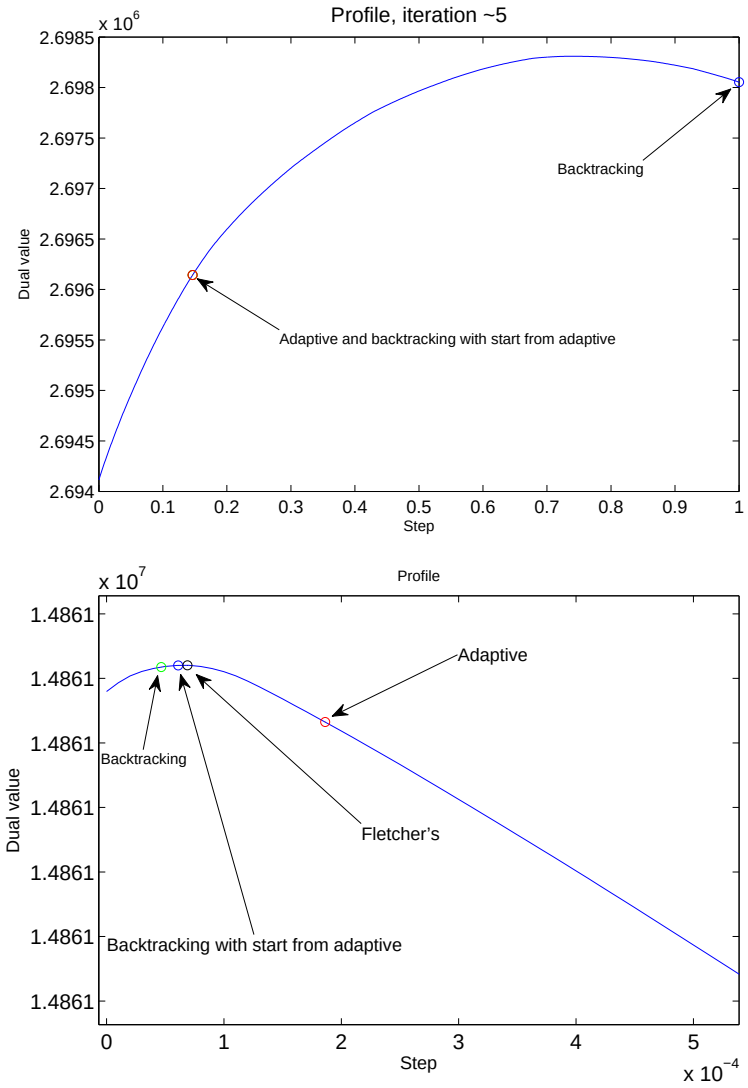


Рис. 3: Профиль двойственной функции вдоль направления оптимизации и неточные максимумы найденные различными алгоритмами.

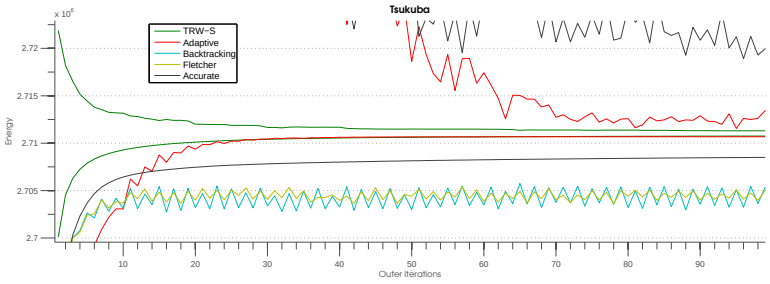


Рис. 4: Субградиентный подъём с разными схемами выбора шага. На горизонтальной оси отложены внешние итерации метода, то есть время потраченное на одномерную оптимизацию не учитывается.

$\{\Lambda^k\}$, $\{f(\Lambda^k)\}$ и $g^k \in \partial f(\Lambda^k)$ составляют *пучок* $\mathcal{B}$.

$$\hat{f}(\Lambda) = \min_{(\Lambda', f(\Lambda'), g') \in \mathcal{B}} \{f(\Lambda') + < g', \Lambda - \Lambda' >\} \tag{11}$$

Для генерации последовательности точек мы используем проксимальный алгоритм:

$$\Lambda^{k+1} = \arg\max_{\Lambda} \{\hat{f}(\Lambda) - \frac{w^k}{2} \left\|\Lambda - \overline{\Lambda}\right\|_2^2\} \tag{12}$$

где $w^k > 0$ нужен чтобы удержать Λ^{k+1} около текущего кандидата на решение $(\overline{\Lambda})$, где $\hat{f}(\Lambda)$ близок к $f(\Lambda)$. По смыслу данный параметр соответствует величине обратной длине шага — чем он больше, тем ближе новая точка будет к исходной.
Если новая точка Λ^{k+1} не ведет к значительному прогрессу, мы не меняем текущую оценку решения $\overline{\Lambda}$, а только уточняем $\hat{f}(\Lambda)$ добавляя в пучок $(\Lambda^{k+1}, f(\Lambda^{k+1}), g^{k+1})$. k–ый шаг в таком случае называют *нулевым шагом*. В противном случае мы обновляем $\overline{\Lambda} = \Lambda^{k+1}$, это называется *значительным шагом*. Чтобы понять какой вид шага нужно выполнять сейчас, мы сравниваем увеличение $f(\Lambda^{k+1})$ и $\hat{f}(\Lambda^{k+1})$ относительно $f(\overline{\Lambda})$. Если отношение этих величин больше чем заранее зафиксированный параметр m_L, тогда аппроксимация $\hat{f}(\Lambda)$ достаточно точна чтобы предпринять значительный шаг.

Задача (12) может быть сведена к квадратичному программированию размерности равной количеству элементов в пучке [13].

В описанном алгоритме осталось два аспекта сильно влияющих на итоговую скорость оптимизации — это управление размером пучка $\mathcal{B}$ (пучок должен быть достаточно маленьким чтобы можно было быстро решать задачу квадратичного программирования возникающую на каждой итерации) и выбор последовательности весов $\{w^k\}$.

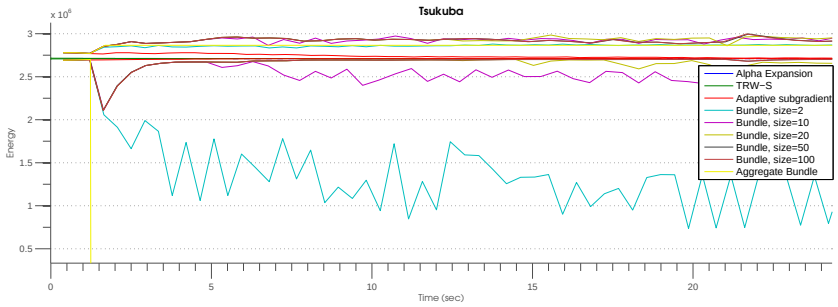


Рис. 5: Использование предложенной схемы выбора весов (13) ведет к расхождению и метода пучка и метода агрегированного пучка.

Управление размером пучка

Чтобы ограничить размер пучка есть два основных подхода [13]:

- Удалять самое малонарушаемое ограничение ограничение как только размер пучка превысит некоторой фиксированный порог n. Метод гарантированно сойдётся если выбрать n достаточно большим, но к сожалению достаточное n сравнимо с размерностью двойственного пространства (в нашем случае порядка миллионов). Тем не менее даже с небольшим размером n методы успешно работают на практике [13], хотя для них нельзя гарантировать сходимость.

- Другой подход предложен Kiwiel [9]. Он предлагает заменить весь пучок одним *агрегированным субградиентом* (выпуклой комбинацией всех субградиентов виденных до сих пор) без потерь теоретических гарантий.

Последовательность весов

Для выбора весов авторы подхода предлагают следующую схему:

$$w^k = P_{[w_{\min}, w_{\max}]}\left(\left(\gamma \cdot \frac{\min_{t\in\{1\dots k\}} Upper^t - \max_{t\in\{1\dots k\}} Dual^t}{\|g^k\|}\right)^{-1}\right) \quad (13)$$

и добиваются значительных успехов [13]. Хотя мы использовали те же данные и параметры алгоритмов для тестирования, методы на основе пучков с таким выбором весов расходится (рис. 5). Схема (13) на всех протестированных стерео–парах выбирает веса порядка 0.01. Если же отказаться от схемы (13) и положить вес равным большой константе, метод работает хуже чем у авторов подхода, но с разумным качеством (рис. 6, 9). На данном этапе мы будем использовать $w^k = 5$ для метода пучка и $w^k = 1$ для

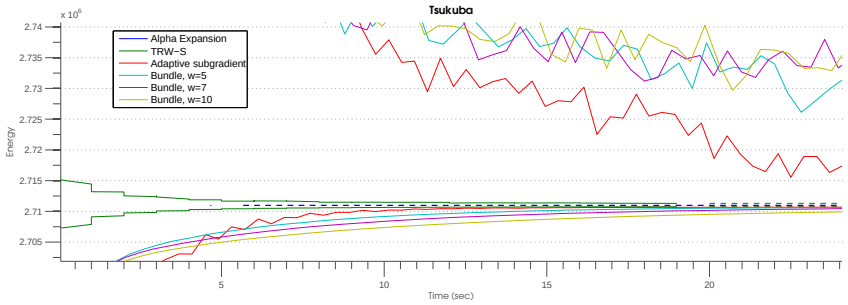


Рис. 6: Метод пучка с различными константными весами.

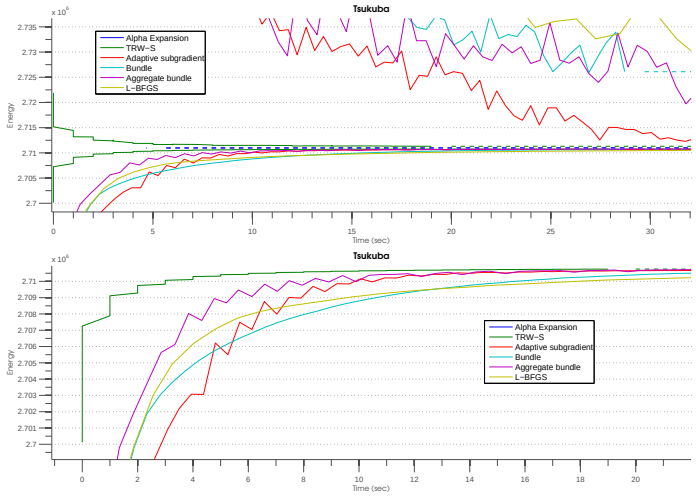


Рис. 7: Итоговое сравнение алгоритмов оптимизации энергии.

метода агрегированного пучка, но в будущем рассчитываем разобраться с проблемой выбора весов и отказаться от примитивного подхода.

L–BFGS

Мы использовали реализацию L–BFGS из библиотеки HANSO[4] (версия для негладкой оптимизации) и применяли её для максимизации двойственной функции.

8 Сравнение подходов

В наших экспериментах (рис. 7, 10) лучше всего работает метод агрегированного пучка (несмотря на константный выбор веса $w^k = 1$). Неожиданно

[4] http://www.cs.nyu.edu/overton/software/hanso/

неплохо работает метод адаптивного субградиентного подъёма.

9 Выводы, вклад

Нами был реализован фреймоворк для удобного сравнивания алгоритмов оптимизации двойственной энергии (исходные коды и примеры использования есть в открытом доступе[5]). С его помощью были выявлены перспективные направления для дальнейшей работы и показано, что метод субградиентного подъёма работает на уровне подходов на основе пучков и применения метода L–BFGS.

10 Приложение

В этом разделе приведены дополнительные графики, иллюстрирующие тезисы на большем числе примеров.

Список литературы

[1] Boykov Y., Veksler O., Zabih R. Fast approximate energy minimization via graph cuts // IEEE Trans. Pattern Anal. Mach. Intell., 2001. — С. 1222–1239.

[2] Szeliski R. A comparative study of energy minimization methods for markov random fields with smoothness-based priors // IEEE Trans. Pattern Anal. Mach. Intell., 2008. — С. 1068–1080.

[3] Kolmogorov V., Zabin R. What Energy Functions can be Minimized via Graph Cuts? // IEEE Trans. Pattern Anal. Mach. Intell., be. — С. 147–159.

[4] Boykov Y., Kolmogorov V. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision // IEEE Trans. Pattern Anal. Mach. Intell., 2004. — С. 1124–1137.

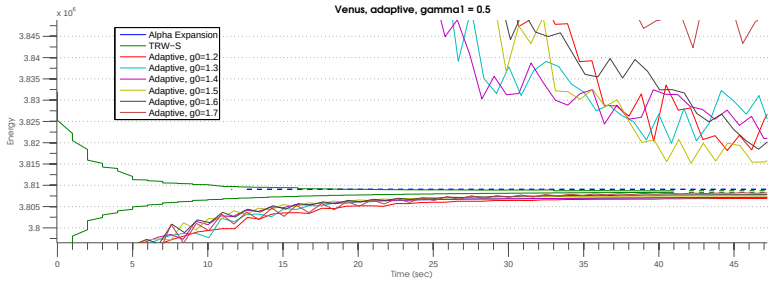
[5] Jörg H. K., Bjoern A., Fred A. H. A Comparative Study of Modern Inference Techniques for Discrete Energy Minimization Problems // CVPR 2013

[6] Komodakis N., Paragios N., Tziritas G. MRF energy minimization and beyond via dual decomposition // IEEE Trans. Pattern Anal. Mach. Intell., 2011. — С. 531–552.

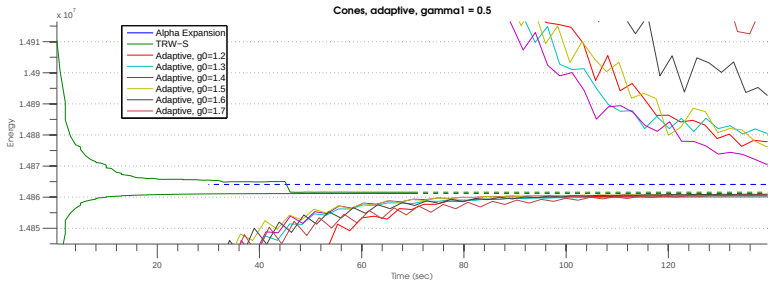
[7] Sun J., Zheng N. N., Shum H. Y. Stereo matching using belief propagation // IEEE Pattern Analysis and Machine Intelligence, 2003. — С. 787–800.

[8] Kolmogorov V. Convergent Tree-Reweighted Message Passing for Energy Minimization // IEEE Trans. Pattern Anal. Mach. Intell., 2006. — С. 1568–1583.

[5] https://github.com/AlexHomework/TRW



(a) Стерео–пара venus.



(b) Стерео–пара cones.

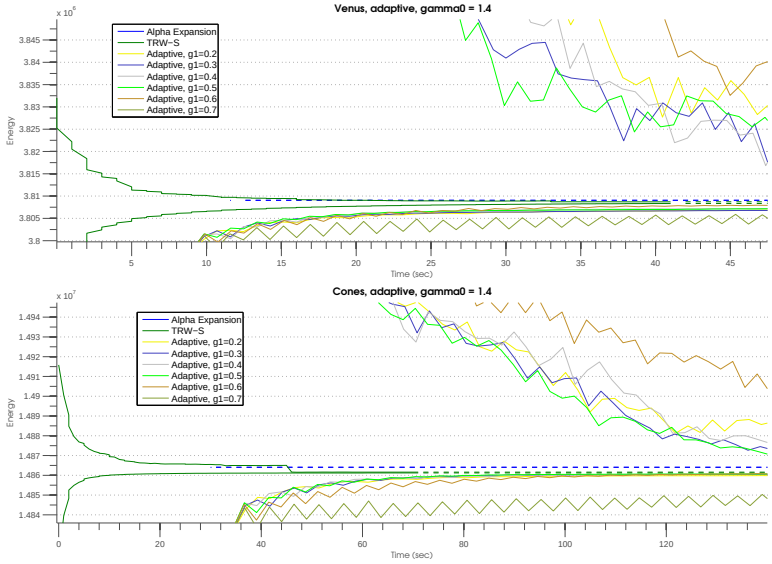


Рис. 8: Подбор параметров адаптивного субградиетного подъёма.

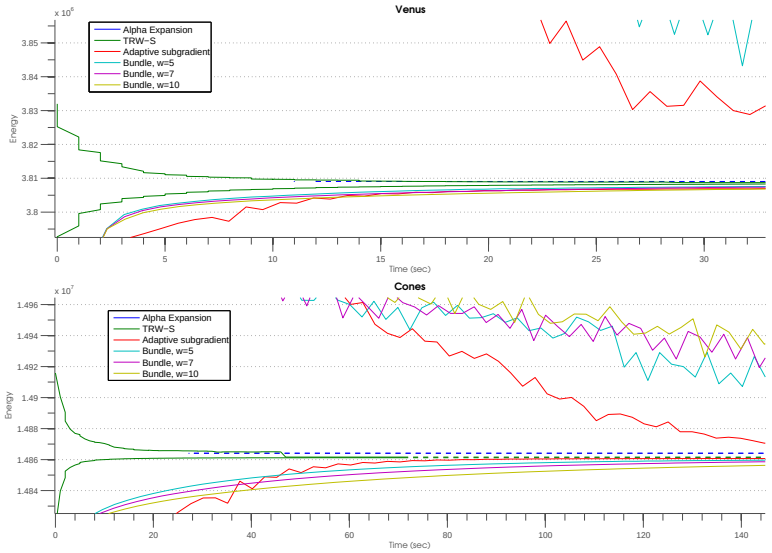


Рис. 9: Метод пучка с различными константными весами.

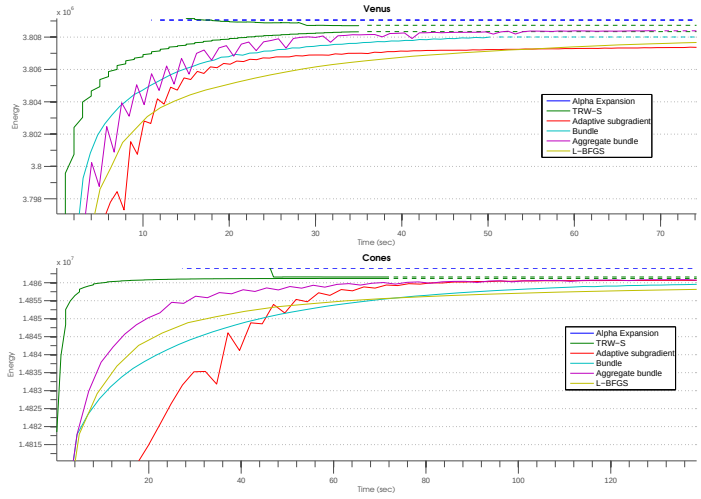


Рис. 10: Итоговое сравнение алгоритмов оптимизации энергии.

[9] Kiwiel K. An aggregate subgradient method for nonsmooth convex minimization // Mathematical Programming, 1983, 27:320–341.

[10] Alahari K., Kohli P., Torr P. H. S. Dynamic Hybrid Algorithms for MAP Inference in Discrete MRFs // IEEE Trans. Pattern Anal. Mach. Intell., 2010. — С. 1846–1857.

[11] Komodakis N., Paragios N., Tziritas G. MRF energy minimization and beyond via dual decomposition // Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2011. — С. 531–552.

[12] Kappes J. H., Bogdan Savchynskyy, Christoph Schnorr A Bundle Approach To Efficient MAP-Inference by Lagrangian Relaxation // Computer Vision and Pattern Recognition (CVPR), IEEE Conference 2012. — С. 1688–1695.

[13] Kappes J. H., Bogdan Savchynskyy, Christoph Schnorr A Bundle Approach To Efficient MAP-Inference by Lagrangian Relaxation // Computer Vision and Pattern Recognition (CVPR), IEEE Conference 2012. — С. 1688–1695.

[14] `http://www.machinelearning.ru/wiki/images/a/a8/MOMO12_min1d.pdf` — Кропотов Д. А., Методы одномерной минимизации, 2012.