

Решение задачи МАР для марковской сети типа решётка

Новиков А. В.
МГУ, ВМиК, каф. ММП

12 апреля 2013 г.

1 Введение

Марковские случайные сети (MRF) — это популярный подход к решению задач анализа данных. Одна из самых важных задач возникающих при использовании MRF — максимизации апостериорной вероятности. Для большинства реальных задач эта задача NP-трудная.

В данной статье проводится обзор существующих state-of-the-art подходов к этой задаче.

1.1 Вклад

Платформа для сравнения алгоритмов и выводы о перспективности тех или иных способов оптимизации.

1.2 Обозначения

2 Данные

Для сравнения подходов мы использовали данные опубликованные Karteek Alahari [3]

3 Обзор методов

На сегодняшний день самым лучшим общим методом решения данной задачи считается метод Alpha-Expansion [?]. Мы будем сравнивать его с TRW-S [1] и методами использующими двойственное разложение. В них задача минимизации исходной энергии¹ сводится к задаче максимизации двойственной энергии (это функция в отличие от исходной энергии является вогнутой кусочно-линейной). Алгоритмы этой группы отличаются конкретный метод максимизации.

- Alpha-Expansion [?]
- TRW-S [1]
- Субградиентный подъём [4]
- L-BFGS [?]
- Bundle methods [5]
- «Полная декомпозиция»

¹От максимизации апостериорной вероятности традиционно переходят к минимизации минус логарифма вероятности (его называют энергией).

4 Постановка задачи

4.1 Двойственное разложение

5 Методы оптимизации двойственной функции

5.1 Субградиентный подъём

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

Адаптивный подход нашёл глобальный минимум, показав результат лучше чем α -расширение, но работал примерно в 60 раз дольше. Использование константного подхода показывает примерно одинаковые результаты для разных констант из хорошего диапазона. Если не угадать с константой, то метод либо разойдётся, либо будет сходиться слишком медленно. В случае адаптивного шага использовалась следующая формула:

$$\alpha_t = \frac{Approx_t - Dual_t}{\|\Delta \vec{g}_t\|^2}$$

Где $Dual_t$ — текущее значение двойственной функции, $Approx_t$ — оценка оптимума двойственной функции.

$$Approx_t = BestDual_t + \delta_t,$$

где $BestDual_t$ — лучшее на данный момент значение двойственной функции,

$$\delta_{t+1} = \begin{cases} \gamma_0 \delta_t, & Dual_t > Dual_{t-1}, \\ \max(\gamma_1 \delta_t, \epsilon) & Dual_t \leq Dual_{t-1}. \end{cases} \quad (1)$$

$\gamma_0, \gamma_1, \epsilon$ — параметры метода, выбранные нами эмпирически.

$$\gamma_0 = 1.5$$

$$\gamma_1 = 0.5$$

$$\epsilon_t = \frac{1}{t}$$

Не смотря на то, что двойственная функция является кусочно линейной, она очень близка к гладкой, так что методы неточной одномерной оптимизации кажутся довольно перспективным направлением работы (рис. 1). Нами был опробован метод Флетчера и «backtracking».

На графиках приведены примеры работы алгоритмов (в том числе длинны шагов, выбранных адаптивным методом) на стереопаре Tsukuba.

5.2 Bundle методы

Основная идея bundle методов — 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 стро-

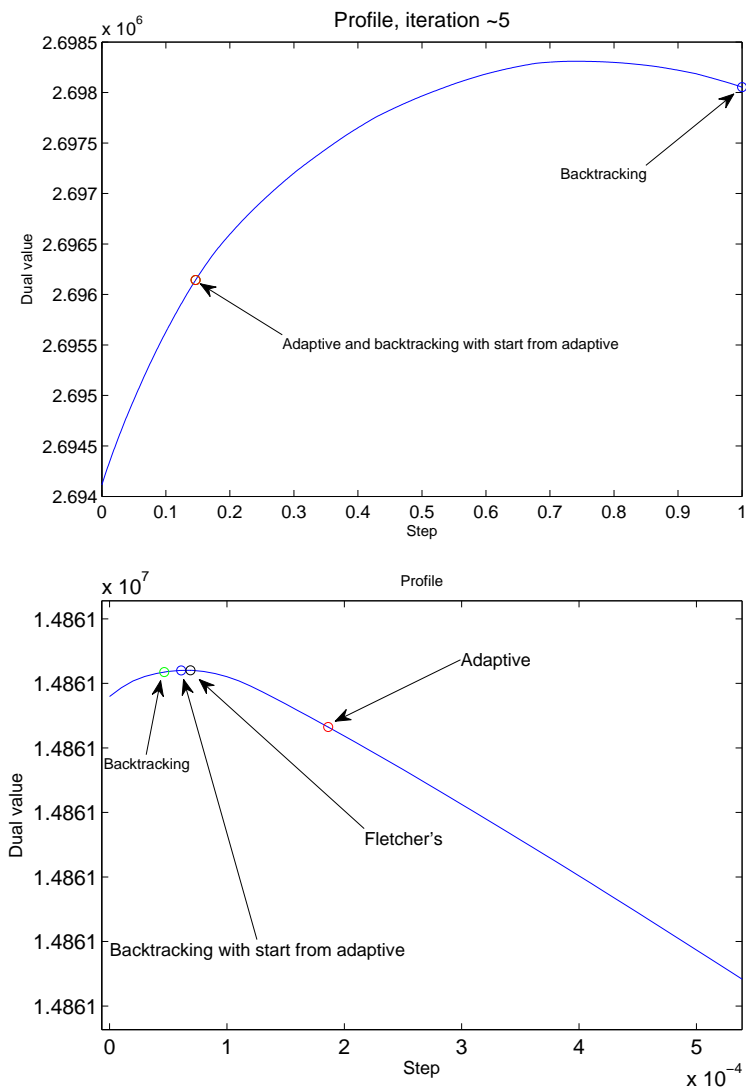
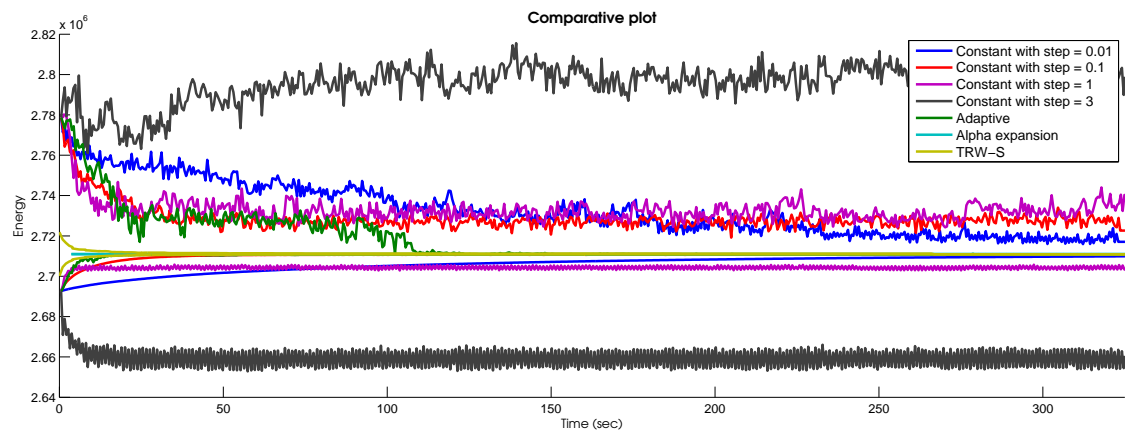
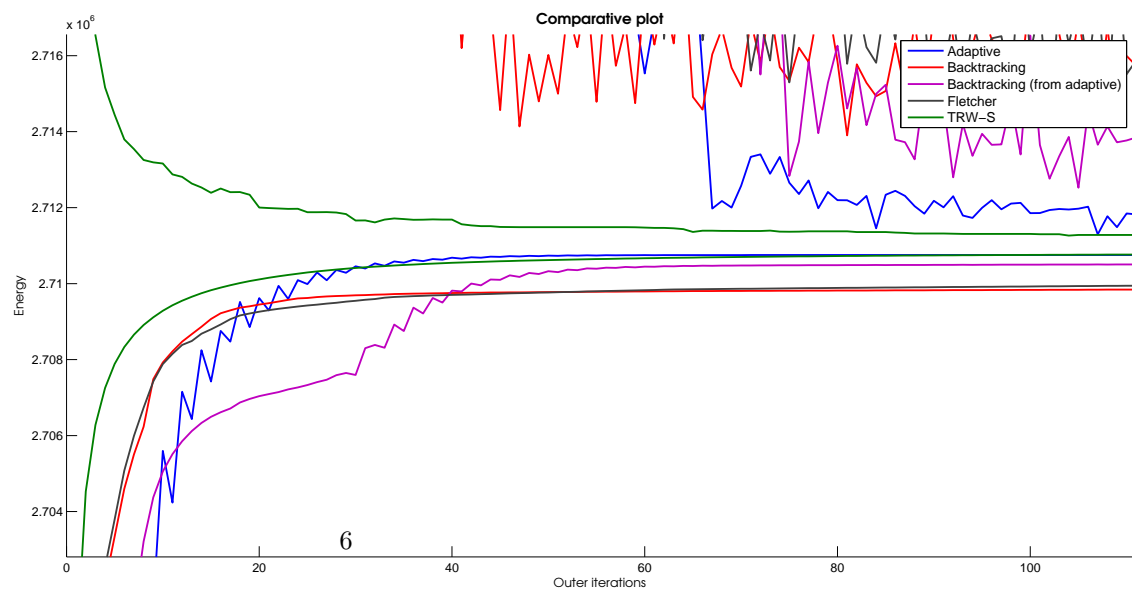
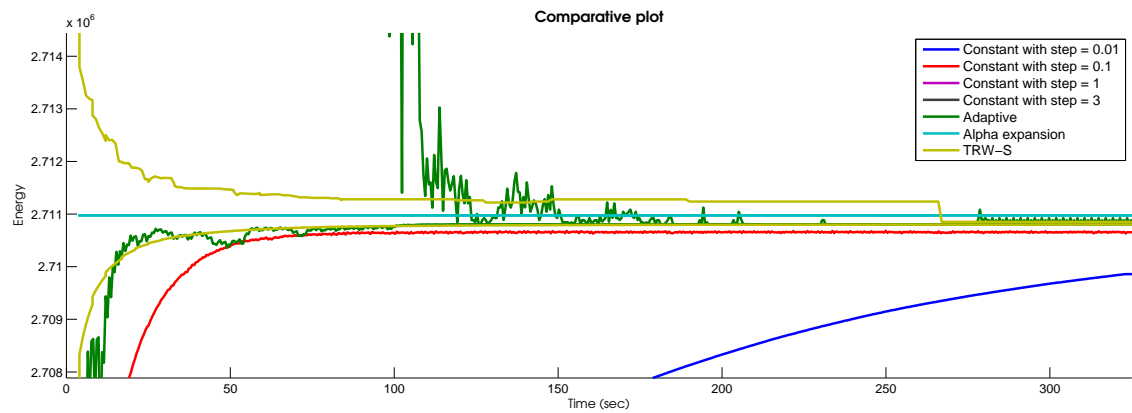


Рис. 1: Профиль двойственной функции вдоль направления оптимизации и неточные максимумы найденные различными алгоритмами



(a) Сравнение скорости и качества работы различных методов. Для α -расширения указана только прямая энергия, для остальных алгоритмов указана прямая и двойственная энергия.



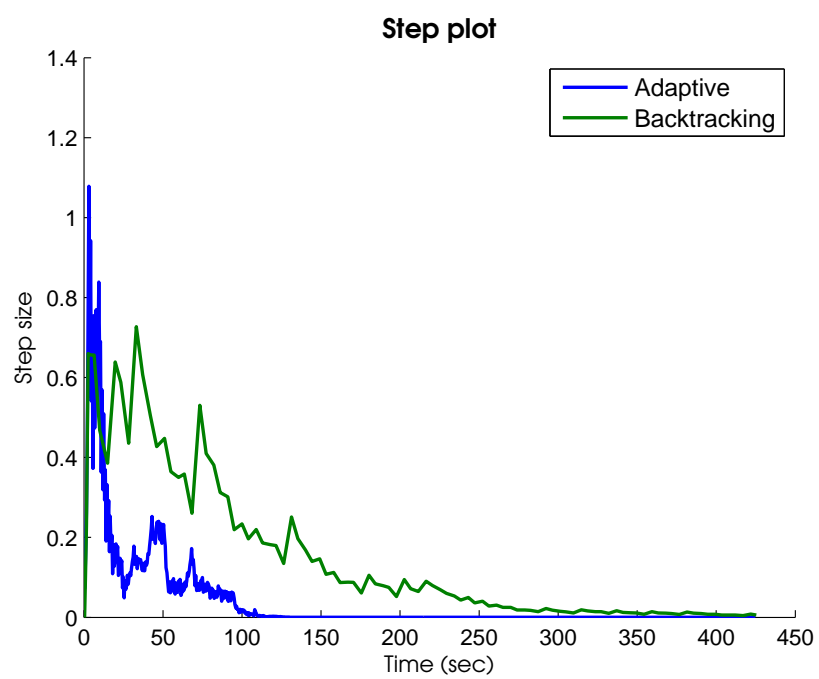


Рис. 2: Длина шага выбранная на каждой итерации адаптивным методом и методом «backtracking»



(a) Итоговая разметка выбранная наилучшим методом (адаптивным)



(b) Разметка выбранная константным методом с длиной шага 0.1

ится по последовательности точек $\{\lambda_i\}$, значений в этих точках $\{f(\lambda_i)\}$ и субградиентов $g^i \in \partial f(\lambda_i)$ так, что $f(\lambda) \leq \hat{f}(\lambda)$ и $f(\lambda_i) = \hat{f}(\lambda_i)$. Вместе $\{\lambda_i\}$, $\{f(\lambda_i)\}$ и $g^i \in \partial f(\lambda_i)$ составляют *bundle* \mathcal{B} .

$$\hat{f}(\lambda) = \min_{(\lambda', f(\lambda'), g') \in \mathcal{B}} \{f(\lambda') + \langle g', \lambda - \lambda' \rangle\} \quad (2)$$

Для генерации последовательности точек мы используем проксимальный алгоритм:

$$\lambda^{k+1} = \arg \max_{\lambda} \left\{ \hat{f}(\lambda) - \frac{w^k}{2} \|\lambda - \bar{\lambda}\|_2^2 \right\} \quad (3)$$

где $w^k > 0$ нужен чтобы удерживать λ^{k+1} около текущего кандидата на решение $(\bar{\lambda})$, где $\hat{f}(\lambda)$ близок к $f(\lambda)$. По смыслу данный параметр соответствует величине обратной длине шага.

Если новая точка λ^{k+1} не ведет к значительному прогрессу, мы не меняем текущую оценку решения $\bar{\lambda}$, а только уточняем $\hat{f}(\lambda)$ добавляя в bundle $(\lambda^{k+1}, f(\lambda^{k+1}), g^{k+1})$. k -ый шаг в таком случае называют *нулевым шагом*. В противном случае мы обновляем $\bar{\lambda} = \lambda^{k+1}$, 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 $f(\bar{\lambda})$. Если отношение этих величин больше чем заранее зафиксированный параметр m_L , тогда аппроксимация $\hat{f}(\lambda)$ достаточно точна чтобы предпринять значительный шаг.

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

5.2.1 Управление размером бандла

5.3 L-BFGS

6 Сравнение подходов

7 Выводы

Список литературы

- [1] Kolmogorov V. Convergent Tree-Reweighted Message Passing for Energy Minimization // IEEE Trans. Pattern Anal. Mach. Intell., 2006. — С. 1568–1583.
- [2] Kiwiel K. An aggregate subgradient method for nonsmooth convex minimization // Mathematical Programming, 1983, 27:320–341.

- [3] Alahari K., Kohli P., Torr P. H. S. Dynamic Hybrid Algorithms for MAP Inference in Discrete MRFs // IEEE Trans. Pattern Anal. Mach. Intell., 2010. — C. 1846–1857.
- [4] Komodakis N., Paragios N., Tziritas G. MRF energy minimization and beyond via dual decomposition // Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2011. — C. 531–552.
- [5] 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. — C. 1688–1695.