Алгоритм TRW приближенного вывода в циклических графических моделях

Дмитрий Ветров
МГУ ВМК
VetrovD@yandex.ru

1 Введение

Алгоритм TRW (tree-reweighted message passing) предназначен для решения задачи приближенного нахождения наиболее вероятной конфигурации скрытых переменных в циклических графических моделях общего вида. Несмотря на кажущуюся сложность, алгоритм построен на использовании нескольких простых шагов

- LP-релаксация;
- Переход к двойственной задаче;
- Разбиение двойственной задачи на подзадачи;
- Переход к дискретным подзадачам и согласование их решений.

Рассмотрим более подробно ту красивую математику, которая позволила создать один из наиболее мощных алгоритмов приближенной дискретной оптимизации.

2 LP-релаксация задачи байесовского вывода

Рассмотрим задачу вывода в марковской сети, задаваемой графом $G = (V, E)$, $|V| = N$. Здесь и далее будем предполагать, что максимальный размер клики в графе равен двум.[1]

[1]Противный случай легко сводится рассматриваемому путем замены клики порядка C на новую переменную арности K^C.

Пусть все переменные сети имеют арность K. Тогда задача поиска наиболее вероятной конфигурации марковской сети сводится к нахождению минимума энергии

$$E(X) = \sum_{i \in V} \phi_j(x_j) + \sum_{(i,j) \in E} \phi_{ij}(x_i, x_j) \to \min_{X \in \{1,\dots,K\}^N}.$$

Это задача дискретной оптимизации, которая, в общем случае, решается полным перебором, т.е. имеет экспоеннциальную сложность. Существуют два важных частных случая, когда задача может быть решена за полиномиальное время. Первый случай соответствует ацикличному графу G (дереву), оптимальная конфигурация на котором может быть найдена с помощью алгоритма передачи сообщений (belief propagation). Второй случай соответствует т.н. субмодулярной энергии бинарных переменных, глобальный минимум которой может быть найден с помощью построения минимального разреза графа. Идея метода TRW заключается в разбиении задачи минимизации энергии общего вида на набор легко решаемых подзадач и последующему согласованию их решений.

Рассмотрим т.н. избыточное представление (overcomplete representation) оптимизируемых переменных. Заменим каждую K-значную переменную x_j на набор бинарных переменных $y_{j1}, \dots, y_{jK}$, из которых одна и только одна переменная может принимать значение единица: $y_{jp} = 1 \Leftrightarrow x_j = p$. Аналогично для пары переменных (x_i, x_j), соединенных ребром рассмотрим набор бинарных переменных $y_{ij,11}, y_{ij,12}, \dots, y_{ij,KK}$, такой что $y_{ij,pq} = 1 \Leftrightarrow x_i = p,\ x_j = q$. Обозначим $\theta_{ip} = \phi_i(p)$, $\theta_{ij,pq} = \phi_{ij}(p,q)$. Тогда энергия марковской сети может быть переписана в виде следующей линейной функции бинарных переменных

$$E(Y, \Theta) = \sum_{i \in V} \sum_{p=1}^{K} \theta_{ip} y_{ip} + \sum_{(i,j) \in E} \sum_{p,q=1}^{K} \theta_{ij,pq} y_{ij,pq}.$$

Эту энергию необходимо минимизировать по бинарным переменным Y. Очевидно, что для сохранения целостности задачи[2] необходимо наложить ограничения

$$\sum_{p=1}^{K} y_{ip} = 1, \quad \sum_{p=1}^{K} y_{ij,pq} = y_{jq}, \quad \sum_{q=1}^{K} y_{ij,pq} = y_{ip}, \quad y_{ip}, y_{ij,pq} \in \{0,1\}.$$

Обозначим это множество ограничений $\mathcal{B}$. Такая задача эквивалентна исходной и, следовательно, ничем ее не проще. Но такая система обозначений допускает простую и интуитивно понятную *релаксацию* путем замены ограничений $y_{ip}, y_{ij,pq} \in \{0,1\}$ на ограничения

[2]Чтобы число степеней свободы переменных Y не превышало числа степеней свободы переменных X, т.е. чтобы каждому допустимому значению Y соответствовало одно и только одно допустимое значение X и наоборот.

вида $y_{ip}, y_{ij,pq} \in [0,1]$. Обозначим релаксированное множество допустимых значений Y за $\mathcal{R}$. Заметим, что после релаксации у нас получилась задача минимизации линейной функции, определенной на множестве с линейными ограничениями, т.е. задача линейного программирования. Для такой задачи существуют полиномиальные алгоритмы решения, например, симплекс-метод.

Из-за релаксации нам пришлось расширить множество допустимых значений переменных, поэтому

$$\min_{Y \in \mathcal{B}} E(Y, \Theta) \geq \min_{Y \in \mathcal{R}} E(Y, \Theta).$$

Это связано с тем, что множество угловых точек[3] $\mathcal{R}$ не совпадает с множеством $\mathcal{B}$, т.е. задача линейного прогрраммирования может давать дробные решения. **Можно показать, что для ациклических графов решение релаксированной задачи всегда будет совпадать с решением исходной задачи.**

3 Двойственное разложение

Решать полученную задачу линейного программирования все еще довольно сложно (размерность числа переменных может достигать десятков и сотен тысяч). Заметим, что массив переменных $\Theta = \{\{\theta_{ip}\}, \{\theta_{ij,pq}\}\}$ полностью определяет решаемую задачу. Разобьем граф G на деревья $\{D_t\}_{t=1}^{T}$, так чтобы каждое ребро и каждая вершина исходного графа входила хотя бы в одно дерево. Пусть n_i — число подграфов, включающих i-ую вершину. Определим для каждого дерева величину θ_{ip}^t следующим образом

$$\theta_{ip}^t = \begin{cases} \frac{\theta_{ip}}{n_i}, & i \in D_t; \\ 0, & i \notin D_t. \end{cases}$$

Аналогично, обозначив n_{ij} количество деревьев, содержащих ребро, соединяющее i-ую и j-ую вершины, определим для каждого дерева величину $\theta_{ij,pq}$ так

$$\theta_{ij,pq}^t = \begin{cases} \frac{\theta_{ij,pq}}{n_{ij}}, & (i,j) \in D_t; \\ 0, & (i,j) \notin D_t. \end{cases}$$

Таким образом, для каждого дерева мы определили массив переменных Θ^t, размер которого равен размеру Θ, причем

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

[3]Как известно из теории линейного программирования, оптимальное значение функционала всегда достигается по меньшей мере, на одной угловой точке допустимого множества.

где под суммой понимается операция покомпонентного сложения. Заметим, что энергия является линейной функцией не только по Y, но и по Θ. Следовательно,

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

Введем вспомогательные переменные $\Lambda=\{\Lambda^t\}_{t=1}^T=\{\{\lambda_{ip}^t\},\{\lambda_{ij,pq}^t\}\}_{t=1}^T\in\mathcal{L}$, где множество $\mathcal{L}$ задается соотношениями

$$\sum_{t=1}^{T}\lambda_{ip}^t=0,\quad\forall i,\forall p$$

и

$$\sum_{t=1}^{T}\lambda_{ij,pq}^t=0,\quad\forall(i,j)\in E,\ \forall p,q.$$

Тогда справедлива следующая цепочка неравенств

$$\min_{Y\in\mathcal{B}}E(Y,\Theta)\geq\min_{Y\in\mathcal{R}}E(Y,\Theta)=\min_{Y\in\mathcal{R}}E(Y,\Theta)+\sum_{t=1}^{T}\left[\sum_{i\in V}\sum_{p=1}^{K}\lambda_{ip}^t y_{ip}+\sum_{(i,j)\in E}\sum_{p,q=1}^{K}\lambda_{ij,pq}^t y_{ij,pq}\right]=$$

$$\min_{Y\in\mathcal{R}}\left\{E(Y,\Theta)+\sum_{t=1}^{T}\left[\sum_{i\in V}\sum_{p=1}^{K}\lambda_{ip}^t y_{ip}+\sum_{(i,j)\in E}\sum_{p,q=1}^{K}\lambda_{ij,pq}^t y_{ij,pq}\right]\right\}=\min_{Y\in\mathcal{R}}E(Y,\Theta+\Lambda)\geq$$

$$\geq\sum_{t=1}^{T}\min_{Y\in\mathcal{R}}E^t(Y,\Theta^t+\Lambda^t)=\{\text{LP-relaxation on trees}\}=\sum_{t=1}^{T}\min_{Y\in\mathcal{B}}E^t(Y,\Theta^t+\Lambda^t).$$

Обратим внимание на два обстоятельства. Во-первых, мы заменили в последнем равенстве задачу линейного программирования на дереве на задачу дискретной оптимизации, для решения которой существуют сверхэффективные методы передачи сообщений, работающие существенно быстрее общих методов решения задачи линейного программирования. Так можно сделать в силу того, что LP-релаксация на деревьях не меняет решения оптимизационной задачи (свойство ацикличных графов). Во-вторых, последнее равенство справедливо $\forall\Lambda\in\mathcal{L}$. Мы получили нижнюю границу для решения исходной дискретной задачи, которая зависит от свободных переменных Λ. Теперь вполне естественно попробовать максимально уточнить нижнюю оценку, взяв по этим свободным переменным максимум. Заметим, что $\min_{Y\in\mathcal{B}}E^t(Y,\Theta^t+\Lambda^t)$ является минимумом конечного числа линейных по Λ^t функций, т.е. так называемой нижней огибающей семейства линейных функций.

Но такая функция вогнута. Таким образом, функционал $\sum_{t=1}^{T} \min_{Y \in \mathcal{B}} E^t(Y, \Theta^t + \Lambda^t)$ является вогнутой по Λ^t, а значит и по Λ функцией, заданной на выпуклом множестве $\mathcal{L}$. Такие функции имеют единственный максимум и могут быть эффективно оптимизированы. Заметим, что эта функция не всюду дифференцируема (например, в изломах нижней огибающей), поэтому для ее оптимизации необходимо воспользоваться методом т.н. условного субградиента. В частности, легко получить формулы пересчета для метода условного субградиентного подъема

$$\lambda_{ip}^{t,\text{new}} = \lambda_{ip}^{t,\text{old}} + \alpha_n \left(\hat{y}_{ip}^t - \frac{\sum_{\{t' | i \in D_{t'}\}} \hat{y}_{ip}^{t'}}{\sum_{\{t' | i \in D'_t\}} 1} \right),$$

где $\hat{y}_{ip}^t = \arg_{(ip)} \min_{Y \in \mathcal{B}} E^t(Y, \Lambda^{t,\text{old}})$, а $\alpha_n > 0$ — шаг по условному субградиенту на n-ой итерации. Аналогично,

$$\lambda_{ij,pq}^{t,\text{new}} = \lambda_{ij,pq}^{t,\text{old}} + \alpha_n \left(\hat{y}_{ij,pq}^t - \frac{\sum_{\{t' | (i,j) \in D_{t'}\}} \hat{y}_{ij,pq}^{t'}}{\sum_{\{t' | (i,j) \in D'_t\}} 1} \right),$$

$\hat{y}_{ij,pq}^t = \arg_{(ij,pq)} \min_{Y \in \mathcal{B}} E(Y, \Lambda^{\text{old}})$.

Но функция $E^t(Y, \Theta^t)$ задает энергию на дереве, а значит может быть легко минимизирована с помощью интерфейса передачи сообщений.

4 Теоретические свойства метода

Теорема Максимум нижней границы совпадает минимумом LP-релаксированной энергии

$$\max_{\Lambda \in \mathcal{L}} \sum_{t=1}^{T} \min_{Y \in \mathcal{B}} E^t(Y, \Theta^t + \Lambda^t) = \min_{Y \in \mathcal{R}} E(Y, \Theta).$$

Задача максимизации нижней границы является двойственной к LP-релаксированной задаче линейного программирования. Как известно из теории выпуклых задач, решения прямой и двойственной задач линейного программирования совпадают.

Теорема Максимум ниженй границы, найденный с помощью TRW, **не зависит** от выбора разбиения на поддеревья.

Это прямое следствие предыдущей теоремы. В самом деле, минимум LP-релаксированной энергии никак от разбиения на деревья не зависит и к нему сходится максимум нижней границы для любого разбиения исходного графа на деревья. Единственным требованием является то, что каждое ребро и каждая вершина исходного графа

должны входить хотя бы в одно дерево разбиения. Стоит отметить, что, хотя можно разбить граф и на отдельные ребра (которые в совокупности образуют допустимое разбиение на деревья), такое разбиение является неэффективным с вычислительной точки зрения. Наиболее разумным с вычислиельной точки зрения является разбиение графа на деревья максимально большего размера (например, остовные), либо на цепочки как можно большей длины. Одна из наиболее эффективных схем оптимизации нижней границы по системе т.н. монотонных цепочек предложена В. Колмогоровым и известна как TRW-S.[4]

Теорема Для субмодулярной энергии бинарного аргумента минимум LP-релаксированной задачи равен минимуму исходной энергии

$$\min_{Y \in \mathcal{R}} E(Y, \Theta) = \min_{Y \in \mathcal{B}} E(Y, \Theta),$$

т.е. TRW сходится к точному решению.

Этот важный результат, с одной стороны, связывает TRW с методами поиска разрезов графов (которые, как мы помним, как раз могут эффективно находить оптимум субмодулярных энергий бинарных переменных), а, с другой стороны, показывают что между бинарными энергиями на циклическом графе и произвольными энергиями на дереве существует определенная связь. Последний факт лег в основу другого подхода по разбиению исходной задачи на более простые, получившего название субмодулярного разложения SMD.

Таким образом, TRW позволяет эффективно находить значение минимума LP-релаксированной энергии. Так как последний представляет собой решение задачи линейного программирования на множестве $\mathcal{R}$, он достигается в одной из угловых точек $\mathcal{R}$. Часть угловых точек является точками множества $\mathcal{B}$, т.е. это точки, в которых все переменные $y_{ip}, y_{ij,pq}$ лежат в множестве $\{0, 1\}$. В случае, если решение оказывается в одной из таких точек, LP-релаксация не приводит к искажению исходной задачи, а в TRW $\arg\min_{Y \in \mathcal{B}} E(Y, \Theta^1 + \Lambda^1_*) = \ldots = \arg\min_{Y \in \mathcal{B}} E(Y, \Theta^T + \Lambda^T_*) = Y_*$, где $\Lambda_* = \arg\max_{\Lambda \in \mathcal{L}} \sum_{t=1}^{T} \min_{Y \in \mathcal{B}} E^t(Y, \Theta^t + \Lambda^t)$. Такая ситуация называется strong tree agreement (STA). В этом случае TRW сходится к точному решению $Y_* = \arg\min_{Y \in \mathcal{B}} E(Y, \Theta)$. Если же решение LP-релаксации не принадлежит $\mathcal{B}$, TRW сходится к точке с т.н. weak tree agreement, в которой решения отдельных подзадач совпадают не полностью. Процесс получения решения исходной задачи в таком случае затруднителен. Наиболее распространенной страгеией является следующая: фиксиурем метки всех вершин, для которых решения подзадач совпали, а по оставшимся[5] выполняем полный или частичный перебор. Среди такх конфигураций находим конфигурацию с минимальной энергией и берем ее в

[4]В алгоритме TRW-S также реализована более эффективная оптимизация нижней границы, построенная на усреднении мин-маргиналов, рассчитанных по деревьям.

[5]как правило, их доля невелика и составляет не более нескольких процентов от общего числа вершин

качестве приближенного минимума исходной задачи. Заметим, что даже полный перебор по множеству, на которых решения подзадач не совпали, **не гарантирует** нахождения глобального оптимума. В тех же случаях, когда это так (это верно, например, при $K = 2$), говорят о частичной оптимальности TRW.