Вероятностное округление для «MAX-CUT»

Н. Н. Кузюрин С. А. Фомин

13 декабря 2011 г.

Задачи полуопределенного и векторного программирования. Вероятностное округление для задачи о максимальном разрезе в графе («MAX-CUT»).

Определение

Пусть есть неориентированный граф $G = (V, E)$. ***Разрезом*** *(сечением, cut) называется разбиение множества вершин V на непересекающиеся множества S и T. т. е. $V = S \cup T$ и $S \cap T = \emptyset$.*

Определение

Для неориентированного графа $G = (V, E)$ и разреза (S, T) ребро $e = (v, t)$ считается ***пересекающим разрез****, если $v \in S$, а $t \in T$.*

Определение

Для графа $G = (V, E)$ ***размером*** *разреза (S, T) считается число ребер, пересекающих этот разрез.*
Если граф — взвешенный, т. е. каждому ребру $e \in E$ соответствует некоторый вес w_e, то размером разреза (S, T) считается сумма весов ребер пересекающих этот разрез:

$$R(S, T) = \sum_{e=(v,t)\in E:\ v\in S, t\in T} w_e.$$

Задача

«Максимальный разрез/MAX-CUT».
Для взвешенного неориентированного графа $G = (V, E)$ с весами $w_e > 0$ найти разрез (S, T) с максимальным весом $R(S, T)$.

Определение

Вероятностный приближенный алгоритм A гарантирует точность C, если для всех входов I

$$1 \geq \frac{\mathbf{E}\, m_A(I)}{m_0(I)} \geq C > 0,$$

где $m_0(I)$ — оптимум, $m_A(I)$ — значение, найденное алгоритмом, и решается задача максимизации.

Для задачи о поиске максимального разреза в простом, невзвешенного графе можно применить простую стратегию: каждую вершину равновероятно ($p = 1/2$) приписать к множеству T или S.

Упражнение

Докажите, что этот вероятностный алгоритм является 0.5-приближенным.

Задача

«MAX-CUT(ЦП)»

$G = (V, E)$ — *входной граф,* $|V| = n$*;*

$W = (w_{ij})$ — *веса ребер,* $n \times n$ *матрица. Для отсутствующего между* v_i *и* v_j *ребра —* $w_{ij} = 0$*;*

y_i — *принадлежность вершины части разреза:*
$v_i \in S \to y_i = 1$, $v_i \in T \to y_i = -1$.
Ребро $(v_i, v_j) \in (S, T) \Leftrightarrow y_i y_j = -1$.

$R(S, T)$ — *Вес разреза* (S, T). $R(S, T) = \sum_{i<j} \frac{1-y_i y_j}{2} w_{ij}$.

Задача целочисленного квадратичного программирования:

$$z_{ЦП} = \sum_{i<j} \frac{1 - y_i y_j}{2} w_{ij} \quad \to \quad \max$$

$$\forall i \; y_i \quad \in \quad \{-1, 1\}.$$

Линейная релаксация 2 «MAX-CUT(ЦП)»

Задача

«MAX-CUT(VP)»

$$
\begin{aligned}
Z_{VP} = \sum_{i<j} \frac{1 - \bar{v}_i \cdot \bar{v}_j}{2} w_{ij} \quad &\rightarrow \quad \text{max} \\
\forall i \; \bar{v}_i \cdot \bar{v}_i \quad &= \quad 1, \\
\forall i \; \bar{v}_i \quad &\in \quad R^n.
\end{aligned}
$$

Линейная релаксация 2 «MAX-CUT(ЦП)»

Задача

«MAX-CUT(VP)»

$$\begin{aligned} Z_{VP} = \sum_{i<j} \frac{1 - \overline{v}_i \cdot \overline{v}_j}{2} w_{ij} &\quad\to\quad \text{max} \\ \forall i \; \overline{v}_i \cdot \overline{v}_i &\quad=\quad 1, \\ \forall i \; \overline{v}_i &\quad\in\quad R^n. \end{aligned}$$

«VP» $\Rightarrow$ можем решать эффективно.

Линейная релаксация 2 «MAX-CUT(ЦП)»

Задача

«MAX-CUT(VP)»

$$\begin{aligned} Z_{VP} = \sum_{i<j} \frac{1 - \bar{v}_i \cdot \bar{v}_j}{2} w_{ij} &\quad \to \quad \max \\ \forall i \; \bar{v}_i \cdot \bar{v}_i &\quad = \quad 1, \\ \forall i \; \bar{v}_i &\quad \in \quad R^n. \end{aligned}$$

«VP» $\Rightarrow$ можем решать эффективно.

$$Z^*_{\text{ЦП}} \leq Z^*_{VP}.$$

Определение

Матрица $X \in R^{n\times n}$ *является* ***положительно полуопределенной*** *если*

$$\forall a \in R^n, \;\; a^{\mathrm{T}} X a \geq 0.$$

Обозначение: $X \succcurlyeq 0$.

Для симметрической $X \in R^{n\times n}$, следующее эквивалентно:

- $X \succcurlyeq 0$;
- X имеет неотрицательные собственные значения;
- $X = V^{\mathrm{T}} V$, для некоторого $V \in R^{m\times n}$, где $m \leq n$.

Задача

«Полуопределенное программирование»[a].

$$
\begin{aligned}
\sum_{i,j} c_{ij}x_{ij} &\to \max(\min)\\
\forall k \sum_{i,j} a_{ijk}x_{ij} &= b_k,\\
X = (x_{ij}) &\succcurlyeq 0,\\
\forall i,j\; x_{ij} &= x_{ji}.
\end{aligned}
$$

[a] В англоязычной литературе SDP, semidefinite programming.

Для задачи есть эффективные полиномиальные алгоритмы, находящие приближенное решение с некоторой аддитивной ошибкой ϵ, и временем, ограниченным полиномом по длине входа и $O(\log(\frac{1}{\epsilon}))$.

Задача

«Векторное программирование»[a].

$$\begin{aligned}\sum_{i,j} c_{ij}(\overline{v}_i \cdot \overline{v}_j) &\to \max(\min)\\ \forall k \sum_{i,j} a_{ijk}(\overline{v}_i \cdot \overline{v}_j) &= b_k,\\ \forall i\ \overline{v}_i \in R^n.\end{aligned}$$

[a] В англоязычной литературе VP, vector programming.

Эквивалентность задач «SDP» и «VP» следует из факторизации положительно полуопределенной матрицы X в виде

$$X = V^{\mathrm{T}} V, \quad (x_{ij} = \overline{v}_i \cdot \overline{v}_j).$$

«SDP-округление MAX-CUT»

Вход: 1 «MAX-CUT» в виде 2 «MAX-CUT(ЦП)»

$(\bar{v}_1,\ \ldots, \bar{v}_n) \leftarrow$ решения релаксации 3 «MAX-CUT(VP)»

случайно выбираем $\bar{r}$ из равномерного распределения векторов единичной длины

$S \leftarrow T \leftarrow \emptyset$

for all $i \in \{1..n\}$ **do**

  if $\bar{v}_i \cdot \bar{r} \geq 0$ **then**

    $S \leftarrow S \cup \{i\}$

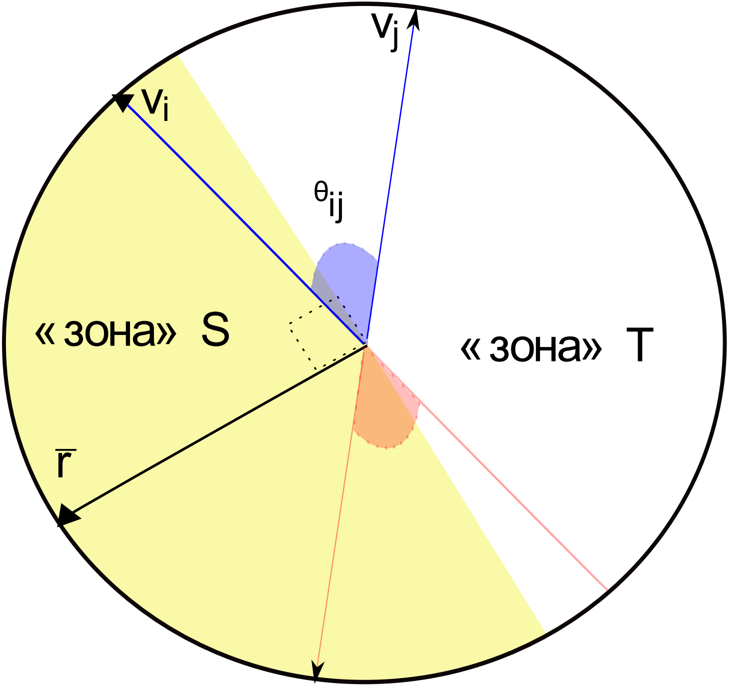
  else

    $T \leftarrow T \cup \{i\}$

  end if

end for

Выход: (S,T) — приближенное решение (1).


v_j
v_i
θ_{ij}
« зона» S
« зона» T
$\bar{r}$

Теорема

Пусть (S^*, T^*) *— оптимальный разрез для задачи 1 «MAX-CUT», тогда для математического ожидания величины разреза* (S', T')*, полученного вероятностным алгоритмом «SDP-округление MAX-CUT» выполняется:*

$$\mathbf{E}[R(S', T')] \geq 0.878 \cdot R(S^*, T^*).$$

$$\mathsf{P}(y_i \neq y_j) = \mathsf{P}(y_i y_j = -1) = \frac{\theta_{ij}}{\pi},$$

$$\mathsf{P}(y_i \neq y_j) = \mathsf{P}(y_i y_j = -1) = \frac{\theta_{ij}}{\pi},$$

$$\mathbf{E} \sum_{i<j} \frac{1 - y_i y_j}{2} w_{ij} = \sum_{i<j} \frac{2 \cdot \mathsf{P}(y_i \neq y_j)}{2} w_{ij} = \sum_{i<j} \frac{\theta_{ij}}{\pi} w_{ij}.$$

$$\mathsf{P}(y_i \neq y_j) = \mathsf{P}(y_i y_j = -1) = \frac{\theta_{ij}}{\pi},$$

$$\mathbf{E} \sum_{i<j} \frac{1 - y_i y_j}{2} w_{ij} = \sum_{i<j} \frac{2 \cdot \mathsf{P}(y_i \neq y_j)}{2} w_{ij} = \sum_{i<j} \frac{\theta_{ij}}{\pi} w_{ij}.$$

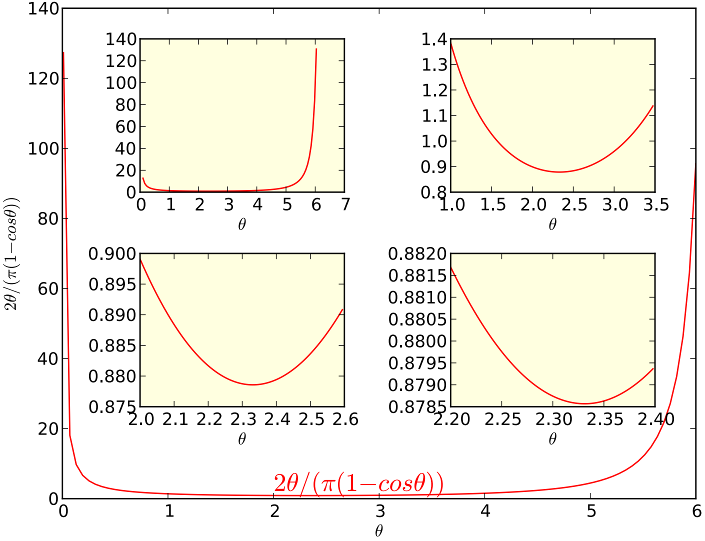
$$Z_{VP} = \sum_{i<j} \frac{1 - \bar{v}_i \cdot \bar{v}_j}{2} w_{ij} = \sum_{i<j} \frac{1 - \cos\theta_{ij}}{2} w_{ij}.$$

$$\mathsf{P}(y_i \neq y_j) = \mathsf{P}(y_i y_j = -1) = \frac{\theta_{ij}}{\pi},$$

$$\mathbf{E}\sum_{i<j} \frac{1 - y_i y_j}{2} w_{ij} = \sum_{i<j} \frac{2 \cdot \mathsf{P}(y_i \neq y_j)}{2} w_{ij} = \sum_{i<j} \frac{\theta_{ij}}{\pi} w_{ij}.$$

$$Z_{VP} = \sum_{i<j} \frac{1 - \bar{v}_i \cdot \bar{v}_j}{2} w_{ij} = \sum_{i<j} \frac{1 - \cos\theta_{ij}}{2} w_{ij}.$$

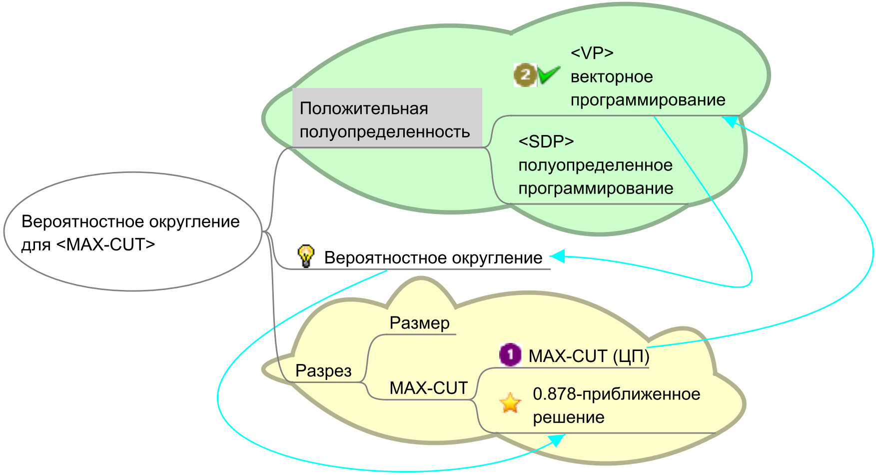
$$\begin{aligned} \frac{\mathbf{E}[R(S', T')]}{R(S^*, T^*)} &\geq \frac{\mathbf{E}[R(S', T')]}{Z_{VP}} \\ &= \frac{\sum_{i<j} \frac{\theta_{ij}}{\pi} w_{ij}}{\sum_{i<j} \frac{1-\cos\theta_{ij}}{2} w_{ij}} \\ &\geq \min_{0 \leq \theta \leq \pi} \frac{2\theta}{\pi(1 - \cos\theta)} \geq 0.878. \end{aligned}$$


$2\theta/(\pi(1-cos\theta))$
θ
$2\theta/(\pi(1-cos\theta))$

$$\min_{0<\theta\leq\pi} \frac{2\theta}{\pi(1-\cos\theta)} \geq 0.878.$$

```
< load(”newton”);
< y:2*x/%pi/(1-cos(x));
< x0:newton(diff(y,x),3);
> 2.331122370414422B0
< y(x0),numer;
> 0.87856720578485
```

«Карта памяти» лекции



Интернет поддержка курса

http://discopal.ispras.ru/

Вопросы?