Algorytmy aproksymacyjne dla problemów stochastycznych

Piotr Sankowski

Uniwersytet Warszawski

PhD Open, listopad 12-13, 2008

Plan - Wykład II

- Boosted sampling:
  - drzewo Steinera,
  - problemy addytywne:
    - lokalizacja Fabryk,
    - las Steinera.
- Metoda prymalnodualna:
  - pokrycie wierzchołkowe.

Boosted sampling

Technika *boosted sampling* została zaproponowana przez Gupta'e, Pál'a, Ravi'ego i Sinha'e '04.

Technika może zostać użyta do problemów addytywnych dla których istnieje aproksymacyjny algorytm deterministyczny spełniający pewne założenia o *podziale kosztów*.

Stochastyczne drzewo Steinera

Rozważać będziemy *stochastyczny problem ukorzenionego drzewa Steinra*, w którym:

- mamy dany graf $G = (V, E)$ z zadanym korzeniu r,
- oraz rozkład prawdopodobieństwa na zbiorach terminali,
- możemy wykupić krawędzie w $1'$wszym etapie płacąc c_e,
- bądź po zrealizowaniu zbioru terminali A płacąc c_e^A.

Naszym celem jest połączenie terminali z korzeniem.

Stochastyczne drzewo Steinera

Nie jest znany żaden algorytm zaakraglający dla tego problemu.

Będziemy zakładać, że $c_e^A = \lambda c_e$, dla całkowitego λ.

Co więcej zakładamy, że G jest pełny i metryczny.

> W przeciwnym przypadku możemy najpierw policzyć najkrótsze ścieżki między wszystkimi parami wierzchołków.

Boosted sampling

Algorytm boosted sampling jest bardzo prosty:

1. wylosuj λ próbek $A_1, \ldots, A_\lambda$ możliwych scenariuszy,
2. rozwiązaniem dla 1'ego etapu będzie minimalne drzewo Steinera Alg_I dla wierzchołków z $S = \bigcup_i A_i$,
3. podłącz zrealizowany zbiór klientów T przy pomocy minimalnego drzewa rozpinającego w G ze ściągniętym Alg_I do wierzchołka.

Boosted sampling

Algorytm boosted sampling jest bardzo prosty:

1. wylosuj λ próbek $A_1, \ldots, A_\lambda$ możliwych scenariuszy,
2. rozwiązaniem dla 1'ego etapu będzie minimalne drzewo rozpinające Alg_I dla wierzchołków z $S = \bigcup_i A_i$,
3. podłącz zrealizowany zbiór klientów T przy pomocy minimalnego drzewa rozpinającego w G ze ściągniętym Alg_I do wierzchołka.

Minimalne drzewo rozpinające jest 2 przybliżone względem drzewa Steinera.

Boosted sampling

Przejdźmy do analizy kosztu algorytmu.

Dla zbioru F niech $c(F)$ oznacza całkowity koszt $\sum e \in Fc_e$.

Niech Z^* będzie optymalnym kosztem algorytmu 2-etapowego.

$$Z^* = c(Opt_I) + \mathbf{E}_{T \subseteq V}[\lambda c(Opt_{II}(T))],$$

gdzie Opt_I i $Opt_I I$ to decyzje algorytmu optymalnego.

Boosted sampling

Zastanówmy się ile kosztuje połączenie wierzchołków w T.

Możemy je połączyć używając $Opt_I + \bigcup_{i=1}^{\lambda} Opt_{II}(A_i)$.

Ponieważ każde A_i jest wygenerowane zgodnie z rozkładem, możemy zapisać wartość oczekiwaną tego kosztu jako:

$$c(Opt_I) + \lambda \mathbf{E}_{T \subseteq V}[c(Opt_{II}(T))],$$

Ponieważ używamy algorytmu 2-aprosksymacyjnego to nasz koszt w 1'wszym etapie wynosi $2Z^*$.

Boosted sampling

Zajmijmy się teraz oszacowaniem kosztu drugiego etapu.

Rozważmy minimalne drzewo rozpinające dla $S \cup T$.

Skierujmy wszystkie krawędzie w tym drzewie do r.

Dla każdego wierzchołka j niech $e(j)$ będzie pierwszą krawędzią na ścieżce do korzenia (tzn. ścieżką do następnego wierzchołka).

Boosted sampling

Przypiszmy $c(e(j))/2$ do każdego wierzchołka $j \in S \cup T - \{r\}$.

Całkowity przypisany w ten sposób koszt to połowa kosztu minimalnego drzewa rozpinającego.

Z drugiej strony koszt ten jest nie większy niż koszt optymalnego drzewa Steinera dla $S \cup T$.

Boosted sampling

Algorytm w 2'gim etapie konstruuje drzewo Steinera dla $T \cup \bigcup_{i=1}^{\lambda} A_i$.

Zauważmy, że między tymi $\lambda + 1$ zbiorami jest pełna symetria.

Możemy je wygenerować najpierw losując $\lambda + 1$ zbiorów a następnie wybrać losowo z rozkładem jednostajnym jeden aby był T.

Boosted sampling

Możemy teraz spojrzeć na całkowity koszt drzewa jako na sumę kosztów $c(e(j))$ dla wierzchołków wybranych w tych $\lambda + 1$ drzewach.

Niektóre wierzchołki mogą być wybrane więcej niż raz i wtedy przypisujemy $c(e(j))$ do jednego z nich wybranego losowo.

Możemy teraz zauważyć, że ze względu na losowy wybór T:

$$\sum_{j \in T - S} c(e(j)) \leq \frac{1}{\lambda + 1} \sum_{j \in S \cup T} c(e(j)).$$

Boosted sampling

Rozważmy teraz drzewo Steinera $\mathcal{T}$ dla $T \cup S$ uformowane przez krawędzie:

$$Opt_I \cup Opt_{II} \cup \bigcup_{i=1}^{\lambda} Opt_{II}(A_i).$$

Z podobnych powodów co poprzednio otrzymujemy:

$$\mathbf{E}[c(\mathcal{T})] \leq c(Opt_I) + (\lambda+1)\mathbf{E}[c(Opt_{II}(T))] \leq \frac{\lambda+1}{\lambda} Z^*.$$

Tak więc optymalne drzewo Steinera na $S \cup T$ kosztuje nie więcej niż $\frac{\lambda+1}{\lambda} Z^*$.

Boosted sampling

Oczekiwany koszt 2'iego etapu wynosi:

$$\mathbf{E}\left[\sum_{j \in T-S} c(e(j))\right] \leq \mathbf{E}\left[\frac{1}{\lambda+1}\sum_{j \in S \cup T} c(e(j))\right] \leq$$

$$\leq \frac{2}{\lambda+1}\mathbf{E}[c(\mathcal{T})] \leq \frac{2}{\lambda+1}\frac{\lambda+1}{\lambda}Z^* = \frac{2Z^*}{\lambda}.$$

Całkowity koszt algorytmu wynosi, więc

$$2Z^* + \lambda \times \frac{2Z^*}{\lambda} = 4Z^*.$$

Boosted sampling

Theorem 1 (Gupta, Pál, Ravi i Sinha '04) *Algorytm* boosted sampling *jest 4-aproksymacyjnym algorytmem dla 2-etapowego problemu drzewa Steinera w modelu czarnej skrzynki.*

Idea tego algorytmu jest próbkowanie każdego scenariusza z prawdopodobieństwem λp_A, po to aby wziąć pod uwagę przyszły wzrost kosztów.

Zaskakujące jest jednak to, że w tym celu potrzeba tylko λ próbek.

Problemy addytywne

Technika *boosted sampling* może zostać zastosowana do szerszej gamy problemów.

Niech U będzie uniwersum żądań, oraz niech X będzie zbiorem elementów które możemy zakupić.

Niech tak jak poprzednio $c(F)$ dla $F \subseteq X$ oznacza sumaryczny koszt elementów w F.

Niech $Sol(S) \subseteq 2^X$ oznacza zbiór możliwych rozwiązań dla S.

Problemy addytywne

W przypadku deterministycznym naszym zadaniem jest dla danego S wybrać element $Sol(S)$ o minimalnym koszcie; niech $OPT(S)$ oznacza koszt tego optymalnego rozwiązania.

Powiemy, że problem jest *addytywny* jeżeli dla dwóch danym zbiorów żądań S i S' oraz rozwiązania dla nich F i F' zbiór $S \cup S'$ definiuje poprawny zbiór żądań oraz $F \cup F'$ jest dla niego poprawnym rozwiązaniem.

Problemy addytywne

W przypadku stochastycznego addytywnego problemu 2-etapowego:

- mamy dany rozkład prawdopodobieństwa na zbiorach żądań,
- możemy w $1'$wszym etapie kupić elementy $e \in X$ w cenie c_e,
- po zrealizowaniu się scenariusza możemy wykupić elementy $e \in X$ w cenie λc_e.

Problemy addytywne

Powiemy, że α-aproksymacyjny algorytm $\mathcal{A}$ pozwala na *β-ścisły mechanizm dzielenia kosztów* jeżeli istnieje funkcja $\xi : 2^U \times U \longrightarrow \mathcal{R}_+$ taka, że dla każdego $S, T \subseteq U$, $S \cap U = \emptyset$:

- $\xi(S, u) = 0$ dla $u \notin S$,
- $\sum_{u \in S} \xi(S, u) \leq c(OPT(S))$,
- istnieje procedura $Aug_{\mathcal{A}}$, która rozszerza rozwiązanie $\mathcal{A}(S)$ do rozwiązania z $Sol(S \cup T)$ generując koszt

$$c(Aug_{\mathcal{A}}(S, T)) \leq \beta \sum_{u \in T} \xi(S \cup T, u).$$

Problemy addytywne

Łatwo jest zauważyć, że w przypadku drzewa Steinera minimalne drzewo rozpinające pozwala na β-ścisły mechanizm dzielenia kosztów.

Minimalne drzewo rozpinające jest 2 przybliżone a więc $\alpha = 2$.

Koszty $c(e(j))/2$ jakie zdefiniowaliśmy dla drzewa dają mechanizm dla $\beta = 2$.

Problemy addytywne

Algorytm boosted sampling dla problemów addytywnych:

1. wylosuj λ próbek $D_1, \ldots, D_\lambda$ możliwych scenariuszy, niech $D = \bigcup_i D_i$,
2. używając algorytmu $\mathcal{A}$ skonstruuj α-aproksymacyjne rozwiązanie pierwszego stopnia $F_0 \in Sol(D)$,
3. podłącz zrealizowany zbiór klientów S używając procedury $Aug_{\mathcal{A}}$.

Problemy addytywne

Theorem 2 (Gupta, Pál, Ravi i Sinha '04) *Algorytm* boosted sampling *jest* $(\alpha + \beta)$*-aproksymacyjnym algorytmem dla* 2*-etapowych problemów addytywnych w modelu czarnej skrzynki.*

Dowód tego twierdzenia przeprowadzimy w analogiczny sposób jak to miało miejsce w przypadku drzewa Steinera.

Problemy addytywne

Niech F_0^* będzie rozwiązaniem optymalnym dla 1'ego etapu, oraz niech F_S^* będzie rozwiązaniem optymalnym dla zrealizowanego zbioru żądań S w drugim etapie.

Koszt optymalnego rozwiązania wynosi:

$$Z^* = c(F_0^*) + \sum_S p_S \lambda c(F_S^*).$$

Zdefiniujmy $Z_0^* = c(F_0^*)$ oraz $Z_r^* = \sum_S p_S \lambda c(F_S^*)$.

Problemy addytywne

Zdefiniujmy $\hat{F}_1 = F_0^* \cup F_{D_1}^* \cup F_{D_2}^* \cup \ldots \cup F_{D_\lambda}^*$. Jest to poprawne rozwiązanie dla D i:

$$\mathbf{E}_D[c(\hat{F}_1)] \leq c(F_0^*) + \mathbf{E}_D[\sum_{i=1}^{\lambda} c(F_{D_i}^*)] =$$

$$= c(F_0^*) + \sum_{i=1}^{\lambda} \mathbf{E}_{D_i}[c(F_{D_i}^*)] =$$

$$= c(F_0^*) + \lambda \sum_{S} p_S c(F_S^*) = Z^*.$$

Ponieważ używamy algorytmy α-aproksymacyjnego to $c(F_0) \leq \alpha c(\hat{F}_1) \leq \alpha Z^*$.

Problemy addytywne

Zajmijmy się teraz drugim etapem.

Niech S będzie zrealizowanym zbiorem żądań, oraz niech F_S będzie wynikiem działania $Aug_{\mathcal{A}}$ takim, że $F_0 \cup F_S \in Sol(S)$.

Potrzebujemy ograniczyć oczekiwany koszt $\lambda\mathbf{E}[c(F_S)]$.

Z istnienia β-ścisłej funkcji ξ wynika, że $c(F_S) \leq \beta\xi(D \cup S, S - D)$.

Problemy addytywne

Wylosujmy $\lambda + 1$ zbiorów $\hat{D}_1, \ldots, \hat{D}_{\sigma+1}$ zgodnie z rozkładem.

Wybierzmy K z rozkładem jednostajnym z $\{1, 2, \ldots, \sigma + 1\}$ i zdefinujmy:

$$S = \hat{D}_K \quad \text{oraz} \quad D = \bigcup_{i \neq K} \hat{D}_i.$$

Ten proces losowania jest zbiorów daje ten sam wynik, co wybór zbiorów w algorytmie, ponieważ każdy zbiór wybrany jest w sposób niezależny.

Problemy addytywne

Zdefiniujmy $\hat{\mathbb{D}}$ jako sumę wszystkich $\hat{D}_i$ oraz $\hat{\mathbb{D}}_{-i}$ jako sumę $\bigcup_{l \neq i} \hat{D}_l$.

Z własności funkcji ξ otrzymujemy:

$$\sum_{i=1}^{\lambda+1} \xi(\hat{\mathbb{D}}, \hat{D}_i - \hat{\mathbb{D}}_{-i}) \leq c(OPT(\hat{\mathbb{D}})).$$

Ponieważ K jest wybrane losowo to:

$$\mathbf{E}_K[\xi(\hat{\mathbb{D}}, \hat{D}_K - \hat{\mathbb{D}}_{-K})] \leq \frac{1}{\lambda + 1} c(OPT(\hat{\mathbb{D}})).$$

Problemy addytywne

Ponieważ ten dwa procesy generowania zbiorów są identyczne to:

$$\mathbf{E}_{D,S}[\xi(D \cup S, S - D)] = \mathbf{E}_{\hat{\mathbb{D}},K}[\xi(\hat{\mathbb{D}}, \hat{D}_K - \hat{\mathbb{D}}_{-K})] \leq$$

$$\leq \frac{1}{\lambda + 1}\mathbf{E}_{\hat{\mathbb{D}}}[c(OPT(\hat{\mathbb{D}}))].$$

Aby zakończyć dowód pokażemy ograniczenie:

$$\mathbf{E}_{\hat{\mathbb{D}}}[c(OPT(\hat{\mathbb{D}}))] \leq \frac{\lambda + 1}{\lambda} Z^*.$$

Problemy addytywne

Poprawne rozwiązanie do $\hat{\mathbb{D}}$ może być ze względu na addytywność otrzymane jako:

$$\hat{F}_2 = F_0^* \cup F_{\hat{D}_1}^* \cup F_{\hat{D}_2}^* \cup \ldots \cup F_{\hat{D}_{\lambda+1}}^*.$$

Mamy więc:

$$\mathbf{E}_{\hat{\mathbb{D}}}[c(OPT(\hat{\mathbb{D}}))] \leq c(F_0^*) + \sum_{i=1}^{\lambda+1} \mathbf{E}_{\hat{D}_i}[c(F_{\hat{D}_i}^*)] \leq$$

$$\leq Z_0^* + (\lambda+1)\frac{Z_r^*}{\lambda} \leq \frac{\lambda+1}{\lambda}(Z_0^* + Z_r^*) = \frac{\lambda+1}{\lambda} Z^*.$$

Problemy addytywne

Otrzymaliśmy więc nierówność:

$$\mathbf{E}_{D,S}[\xi(D \cup S, S - D)] \leq \frac{\lambda + 1}{\lambda} Z^*.$$

Ponieważ nasz algorytm $Aug_{\mathcal{A}}$ jest β-ścisły to:

$$\mathbf{E}_S[c(F_S)] \leq \beta \mathbf{E}_{D,S}[\xi(D \cup S, S - D)] \leq \frac{\beta}{\lambda} Z^*.$$

Ponieważ w 2'gim etapie płacimy λ razy więcej to nasz koszt wynosi βZ^*.

Sumaryczny koszt wynosi więc $(\alpha + \beta) Z^*$.

Problemy addytywne

Theorem 3 (Pál i Tardos '03, Mettu i Plaxton '00) *Istnieje 3-aproksymacyjny 5.45-ścisły algorytm dla problemu lokalizacji fabryk.*

Corollary 4 *Istnieje 8.45-aproksymacyjny algorytm dla sotchastycznego problemu lokalizacji fabryk w modelu czarnej skrzynki.*

Pokrycie wierzchołkowe

Theorem 5 (Gupta, Pál, Ravi i Sinha '04) *Istnieje 2-aproksymacyjny 6-ścisły algorytm dla problemu pokrycia wierzchołkowego.*

Corollary 6 *Istnieje 8-aproksymacyjny algorytm dla sotchastycznego problemu pokrycia wierzchołkowego w modelu czarnej skrzynki.*

Las Steinera

Theorem 7 (Gupta, Kumar, Pál i Roughgarden '03) *Istnieje 2-aproksymacyjny 3-ścisły algorytm dla problemu lasu Steinera.*

Corollary 8 *Istnieje 5-aproksymacyjny algorytm dla sotchastycznego problemu lasu Steinera w modelu czarnej skrzynki.*

Metoda prymalnodualna

Primal	**Dual**
$\min c^T x$	$\max y^T b$
$Ax \geq b$	$y^T A \leq c^T$
$x \geq 0$	$y \geq 0$

Theorem 9 (Słaba dualność) *Dla dowolnej pary rozwiązań* (x, y), *mamy* $y^T b \leq c^T x$.

Theorem 10 (Silna dualność) *Jeżeli istnieje jedno rozwiązania optymalne to istnieje drugie i*

$$(y^*)^T b = c^T x^*.$$

Metoda prymalnodualna

Rozważmy problem pokrycia wierzchołkowego w grafie $G = (V, E)$. Możemy dla niego zapisać relaksacje programu liniowego:

$$\min \sum_{v \in V} c_v x_v,$$

$$\sum_{v \in e} x_v \geq 1 \quad \forall e \in E,$$

$$x_v \geq 0 \qquad \forall v \in V.$$

Metoda prymalnodualna

Program dualny ma postać:

$$\max \sum_{e \in E} y_e,$$

$$\sum_{e \in \delta(v)} y_e \leq c_v \quad \forall v \in V,$$

$$y_e \geq 0 \qquad \forall e \in E.$$

Jeżeli uda nam się ograniczyć $\sum_v x_v$ przez $\rho \sum_e y_e$ to:

$$\sum_v x_v \leq \rho \sum_e y_e \leq \rho VC-LP-OPT \leq \rho VC-OPT.$$

Metoda prymalnodualna

W metodzie prymalnodualnej zaczynamy od $x = 0$ i $y = 0$.

Zwiększamy powoli y'eki aż jakaś nierówność jest spełniona z równością.

Dla tej spełnionej nierówności zwiększamy wartość x.

Można rozumieć ten schemat jako kupowanie x'ów za y'ki.

Metoda prymalnodualna

Algorytm:

- zainicjalizuj $x = 0, y =,$
- dopóki $E \neq \emptyset$
  - wybierz dowolny niepusty zbiór $E' \subseteq,$
  - podnieś y_e dla każdego $e \in E'$ do osiągnięcia jakiejś równości,
  - niech S będzie zbiorem wierzchołków dla równych nierówności,
  - przypisz $x_v = 1$ dla każdego $v \in S$ oraz usuń krawędzie sąsiednie z S z E.
- zwróć $A = \{v : x_v = 1\}$.

Metoda prymalnodualna

Rozwiązania prymalne x jest poprawne.

Każda krawędź zostanie usunięta a więc dla każdej z nich istnieje sąsiedni wierzchołek o $x_v = 1$.

Rozwiązanie dualne y jest poprawne.

Żadne ograniczenie nie jest przekroczone, bo w momencie otrzymania równości usuwamy krawędzie należące do tego ograniczenia.

Metoda prymalnodualna

Mamy teraz:

$$c^T x = \sum_{v \in A} c_v = \sum_{v \in A} \left(\sum_{e \in \delta(v)} y_e \right) =$$

Jak ustaliliśmy c_v to mieliśmy równość.

$$= \sum_{e \in E} \left(\sum_{v \in A \cap e} 1 \right) y_e \leq 2 \cdot \sum_{e} y_e.$$

Ostatnia nierówność wynika z tego, że krawędź ma dwa końce.

Metoda prymalnodualna

Ćwiczenie...

Uogólnić tą metodę do stochastycznego pokrycia wierzchołkami w modelu wielomianowym.

Metoda prymalnodualna

Rozważmy problem pokrycia wierzchołkowego w grafie $G = (V, E)$. Możemy dla niego zapisać relaksacje programu liniowego:

$$\begin{aligned} \min \sum_{v \in V} c_v x_v + \sum_{k} p_k \sum_{v \in V} c_v^k x_{k,v}, & \\ \sum_{v \in e} x_v + \sum_{v \in e} x_{k,v} \geq 1 & \quad \forall k, e \in E_k, \\ x_v, x_{k,v} \geq 0 & \quad \forall k, v \in V. \end{aligned}$$

Metoda prymalnodualna

Program dualny ma postać:

$$\max \sum_{k} \sum_{e \in E_k} y_{k,e},$$

$$\sum_{e \in \delta(v), e \in E_k} y_{k,e} \leq p_k c_v^k \quad \forall k, v \in V,$$

$$\sum_{k} \sum_{e \in \delta(v), e \in E_k} y_{k,e} \leq c_v \quad \forall v \in V,$$

$$y_{k,e} \geq 0 \qquad \forall k, e \in E_k.$$