Algorytmy aproksymacyjne dla problemów stochastycznych

Piotr Sankowski

Uniwersytet Warszawski

PhD Open, listopad 12-13, 2008

Plan

- Wykład I - 2-etapowe algorytmy stochastyczne: Wstęp
- Wykład II - 2-etapowe algorytmy stochastyczne: Rozszerzenie
- Wykład III - algorytmy online i stochastyczne algorytmy online
- Wykład IV - algorytmy uniwersalne i stochastyczne algorytmy uniwersalne

Plan - Wykład I

- 2-etapowe problemy stochastyczne
  - wstęp
  - przegląd wyników
- Stochastyczny problem pokrycia zbiorami
  - definicja
  - metoda zaokrąglania LP
- Stochastyczny problem lokalizacji fabryk
  - definicja
  - metoda zaokrąglania LP

Problemy stochastyczne

W wielu problemach nie znamy dokładnej informacji o przyszłości, ale mamy pewną wiedzę o możliwych scenariuszach oraz ich prawdopodobieństwach.

W stochastycznych problemach optymalizacji próbujemy opisać tą częściową wiedzę oraz wykorzystać ją aby skonstruować jak najlepsze rozwiązania.

Badanie problemów stochastycznych zostało zapoczątkowane w latach 50'tych zeszłego wieku (*Beale '55 oraz Dantzig '55*).

2-etapowe problemy

Problemy te doczekały się szerszego zainteresowania dopiero w ostatnich latach.

Najważniejszym i najlepiej zbadanym modelem jest model 2-etapowych problemów, w których:

- mając daną tylko informację o możliwych scenariuszach możemy zbudować wstępne rozwiązanie (1'wszy etap),
- gdy dany scenariusz zostanie zrealizowany możemy dokupić brakującą część rozwiązania (2'gi etap).

2-etapowe problemy

Zazwyczaj koszt akcji wykonywanych w drugim etapie jest wyższy niż koszt tych samych akcji wykonanych w pierwszym etapie.

Może to być na przykład związane z tym, że akcje drugiego etapu muszą zostać wykonane szybko w reakcji na zaistniałą sytuację.

W problemach tych musimy więc zdecydować o kompromisie między podjęciem tanich akcji na podstawie niepewnych danych, oraz wykonaniem droższych akcji później.

Lokalizacja fabryk

W problemie lokalizacji fabryk mamy za zadanie otworzyć fabryki tak, aby sprostać żądaniom klientów.

Możliwe, że za nim rzeczywiste żądania zostaną zgłoszone poznamy ich statystyczny rozkład, poprzez symulacje, czy przeprowadzenie badań rynku.

W takim przypadku, możemy:

- w 1'wszym etapie zaplanować otworzenie pewnych fabryk,
- w 2'gim etapie otworzyć dodatkowe fabryki i przypisać klientów do fabryk.

2-etapowe problemy

Problemy 2-etapowe możemy sformalizować w następujący sposób:

- mamy dany rozkład prawdopodobieństwa p_A dla zbioru możliwych scenariuszy S,
- wstępne rozwiązanie x kosztuje $c(x)$,
- po zrealizowaniu scenariusza A możemy rozszerzyć rozwiązanie o y_A płacąc $f_A(x, y_A)$.

Naszym celem jest zminimalizowanie oczekiwanego kosztu:

$$c(x) + \mathbf{E}_A \left[f_A(x, y_A) \right].$$

Opis scenariuszy

Pozostała nam do ustalenia jeszcze kwestia sposobu zapisu listy scenariuszy.

Możemy zadać ich listę razem z ich prawdopodobieństwami.

Taki zapis może spowodować jednak, że dane wejściowe przestaną być wielomianowe względem standardowych parametrów problemu.

Możemy wprowadzić *wielomianowy stochastyczny model*, w którym aby uniknąć tego problemu ograniczamy liczbę scenariuszy do wielomianowej.

Opis scenariuszy

Innym sposobem opisu jest *model niezależnej aktywacji* wprowadzony przez Kargera *et al.* '04.

W modelu tym każdy element zbioru bazowego włączany jest do aktywnego scenariusza niezależnie z zadanym prawdopodobieństwiem.

Pozwala zadać wykładniczo wiele scenariuszy i może zostać użyty do modelowania niepewności w różnych przypadkach.

Opis scenariuszy

Często mamy jednak doczynienia z danymi, które są w jakiś sposób skorelowane.

W takim przypadku nie możemy użyć modelu niezależnej aktywacji.

Bardziej ogólnym rozwiązaniem jest *model czarnej skrzynki*, gdzie rozkład prawdopodobieństwa dostępny jest tylko poprzez procedurę próbkującą scenariusze.

Łączy zalety poprzednich modeli.

Algorytmy aproksymacyjne

Interesuje nas stworzenie algorytmów aproksymacyjnych dla tych problemów.

Koszt rozwiązania dla nas to koszt 1'ego i oczekiwany koszt 2'ego etapu.

ρ aproksymacyjny algorytm to algorytm działający w czasie wielomianowym, który zwraca poprawne rozwiązanie o koszcie co najwyżej ρ razy większym niż koszt rozwiązania optymalnego.

Schematy aproksymacyjne

Wielomianowy schemat aproksymacyjny to rodzina algorytmów $\{A_\epsilon\}$, dla $\epsilon > 0$, gdzie A_ϵ jest $1+\epsilon$ przybliżony.

Jeżeli czas działania algorytmu A_ϵ może być ograniczony przez wielomian w $1/\epsilon$ to schemat taki nazywamy *w pełni wielomianowym schematem aproksymacyjnym*.

Możliwe rozwiązania

Algorytmy rozwiązujące te problemy można podzielić na kilka grup:

- rozwiązujące program liniowy dla problemu stochastycznego,
  metody prymalnodualne, zaaokraglanie itp.
- algorytmu próbkujące,
  rozwiązują problem na małej próbce.
- sprowadzenia między różnymi modelami,
  pozwalają użyć rozwiązania z prostszego modelu.

Pokrycie zbiorami

W problemie pokrycia zbiorami mamy dane:

- n elementowe uniwersum U,
- rodzinę zbiorów $\mathcal{S}$ o wagach w_S,

Naszym celem jest znalezienie podrodziny $\mathcal{S}$ o najmniejszej wadze, której suma zawiera każdy element z U.

Najlepszy algorytm dla tego problemu ma współczynnik aproksymacji $\ln n$, oraz nie istnieje lepszy algorytm pod warunkiem, że $P \neq NP$.

Pokrycie zbiorami

Relaksacja programu liniowego ma postać:

$$\min \sum_{S \in \mathcal{S}} w_S x_S,$$

$$\sum_{S \in \mathcal{S}: e \in S} x_S \geq 1 \quad \forall e \in U,$$

$$x_S \geq 0 \qquad \forall S.$$

Z rozwiązania tego LP możemy otrzymać rozwiązanie problemu całkowitego przez $\log n$-krotne randomizowane zaokrąglenie.

Stochastyczne pokrycie zbiorami

W problemie stochastycznym mamy dane:

- n elementowe uniwersum U,
- rodzinę zbiorów $\mathcal{S}$ o wagach w_S,
- rozkład prawdopodobieństwa nad 2^U,

Naszym celem jest:

- w 1'wszym etapie wybrać zbiory z S o koszcie w_X^I,
- w 2'gim etapie dla scenariusza A wybieramy zbiory z S o koszcie w_S^A.

Tutaj $c(x) = \sum_S w_S^I x_S$ i $f_A(x, r_A) = \sum_S w_S^A r_{A,S}$.

Stochastyczne pokrycie zbiorami

Relaksację programu liniowego dla problemu stochastyczngo możemy zapisać:

$$\min \sum_{S \in \mathcal{S}} w_S^I x_S + \sum_{A \subseteq U, S} p_A w_S^A r_{A,S},$$

$$\sum_{S \in \mathcal{S}: e \in S} (x_S + r_{A,S}) \geq 1 \qquad \forall A \subseteq U, e \in A,$$

$$x_S, r_{A,S} \geq 0 \qquad \forall A, S.$$

Zmienna x_S oznacza czy zbiór został wybrany w 1'wszym etapie, a $r_{A,S}$ czy został wybrany w drugim etapie dla scenariusza A.

Stochastyczne pokrycie zbiorami

Program ten możemy przepisać do nowej postaci:

$$h(x) := \min \sum_{S \in \mathcal{S}} w_S^I x_S + \sum_{A \subseteq U} p_A f_A(x)$$

$$0 \leq x_S \leq 1 \qquad \forall S,$$

gdzie:

$$f_A(x) := \min \sum_S w_S^A r_{A,S},$$

$$\sum_{S:e \in S} r_{A,S} \geq 1 - \sum_{S:e \in S} x_S \qquad \forall e \in A,$$

$$r_{A,S} \geq 0 \qquad \forall S.$$

Stochastyczne pokryci zbiorami

Nowa postać programu liniowego

- jest równoważna poprzedniej,
- jej funkcja celu $h(x)$ jest wypukła.

W takim przypadku minimum lokalne jest minimum globalnym.

Metoda gradientów daje nam minimum lokalne (wystarczą przybliżone gradienty).

Program ten daje tylko rozwiązanie dla 1'ego etapu.

Stochastyczne pokrycie zbiorami

Zdefiniujmy $\lambda = \max\left(1, \max_{S,A} \frac{w_S^A}{w_S^I}\right)$.

Twierdzenie 1 (Shmoys and Swamy '04)
Istnieje algorytm znajdujący poprawne rozwiązanie dla powyższego programu liniowego o koszcie co najwyżej $(1+\epsilon)\mathsf{OPT}$ *z prawdopodobieństwem co najmniej* $1-2\delta$ *w czasie wielomianowym ze względu na rozmiar wejścia* λ, $\frac{1}{\epsilon}$ *oraz* $\ln(\frac{1}{\delta})$.

Wynik ten może być uogólniony do problemu lokalizacji fabryk.

Stochastyczne pokrycie zbiorami

Twierdzenie 2 (Shmoys and Swamy '04) *Mając dany algorytm dla detrministycznego pokrycia zbiorami o współczynniku aproksymacji ρ możemy przekształcic dowolne rozwiązanie x powyższego programu liniowego do rozwiązania całkowitoliczbowego o koszcie $2\rho h(x)$.*

Niech r_A^* będzie optymalnym rozwiązaniem $f_A(x)$ 2'ego tapu dla x.

Zauważmy, że każdy element e jest pokryty co najmniej w połowie zmiennymi x_S bądź zmiennymi $r_{A,S}^*$ w każdym scenariuszu zawierającym e.

Stochastyczne pokrycie zbiorami

Zdefiniujmy $E = \{e : \sum_{S:e \in S} x_S \geq \frac{1}{2}\}$.

Wtedy $2x$ jest rozwiązaniem ułamkowym dla instancji o uniwersum E.

Używając algorytmu ρ przybliżonego możemy otrzymać rozwiązanie $\tilde{x}$ dla E o koszcie $2\rho \sum_S w_S^I x_S$.

Przyjmiemy $\tilde{x}$ jako rozwiązanie dla 1'ego etapu.

Stochastyczne pokrycie zbiorami

Dla zrealizowanego scenariusza A wiemy, że $2r^*_A$ jest ułamkowym pokryciem dla $A - E$.

Ponieważ dla każdego elementu poza E mamy $\sum_{S:e\in S} r^*_{A,S} \geq \frac{1}{2}$.

Możemy więc skonstruować pokrycie tych elementów o koszcie co najwyżej $2\rho \sum_S w^A_S r^*_{A,S}$.

Oczekiwany koszt rozwiązania wynosi więc nie więcej niż $2\rho h(x)$.

Stochastyczne pokrycie zbiorami

Twierdzenie 3 (Shmoys and Swamy '04)
Dla każdego $\epsilon > 0$*, istnieje algorytm* $(2 \ln + \epsilon)$*-aproksymacyjny dla stochastycznego problemu pokrycia zbiorami.*

Pokrycie wierzchołkowe

Rozważmy stochastyczny problem pokrycia wierzchołkowego, w którym:

- zbiór krawędzi A jest scenariuszem,
- mamy za zadanie pokryć go wierzchołkami:
  - w 1'wszym etapie płacąc koszt w_v^I,
  - w 2'gim etapie płacąc koszt w_v^A.

Problem pokrycia wierzchołkowego jest szczególnym przypadkiem pokrycia zbiorami, ale współczynnik aproksymacji wynosi 2.

Pokrycie wierzchołkowe

Twierdzenie 4 (Shmoys and Swamy '04)
Dla każdego $\epsilon > 0$, istnieje algorytm $(4+\epsilon)$-aproksymacyjny dla stochastycznego problemu pokrycia wierzchołkowego.

Problem multicut

Rozważmy problem multicut na drzewach, w którym:

- zbiór par wierzchołków jest scenariuszem,
- mamy zadanie rozciąć wszystkie pary poprzez usunięcie krawędzi:
  - w 1'wszym etapie płacąc koszt w_e^I,
  - w 2'gim etapie płacąc koszt w_e^A.

Problem multicut jest szczególnym przypadkiem pokrycia zbiorami, ale współczynnik aproksymacji wynosi 2.

Problem multicut

Twierdzenie 5 (Shmoys and Swamy '04)
Dla każdego $\epsilon > 0$, istnieje algorytm $(4+\epsilon)$-aproksymacyjny dla stochastycznego problemu multicut na drzewach.

Używając wyniku Räcke '08 o uniwersalnym routingu otrzymujemy.

Twierdzenie 6 (Shmoys and Swamy '04)
Dla każdego $\epsilon > 0$, istnieje algorytm $\log n$-aproksymacyjny dla stochastycznego problemu multicut.

Problem lokalizacji fabryk

W deterministycznym problemie lokalizacji fabryk mamy dane:

- zbiór możliwych do zbudowania fabryk $\mathcal{F}$,
  - koszt otwarcia fabryki i wynosi f_i,
- zbiór klientów $\mathcal{D}$,
  - koszt podłączenia klienta j do fabryki i wynosi c_{ij}.

Naszym celem jest otworzenie fabryk i podłączenie od nich klientów, w taki sposób aby zminimalizować całkowity koszt.

Problem lokalizacji fabryk

Relaksacja programu liniowego ma postać:

$$\min \sum_{i \in \mathcal{F}} f_i y_i + \sum_{i \in \mathcal{F}, j \in \mathcal{D}} c_{ij} x_{ij},$$

$$x_{ij} \leq y_i \qquad \forall i \in \mathcal{F}, j \in \mathcal{D},$$

$$\sum_{i \in \mathcal{F}} x_{ij} \geq 1 \qquad \forall j \in \mathcal{D},$$

$$x_{ij}, y_i \geq 0 \qquad \forall i \in \mathcal{F}, j \in \mathcal{D}.$$

zmienne y_i kodują otwarte fabryki, a x_{ij} połączenia klientów.

Problem lokalizacji fabryk

Dla deterministycznego problemu lokalizacji fabryk Mahdian, Ye, and Zhang '02 pokazali, że istnieje algorytm 1.52-aproksymacyjny.

My pokażemy jak przy jego pomocy przekształcić ułamkowe rozwiązanie dla stochastycznej wersji programu liniowego w rozwiązanie całkowitoliczbowe.

Stochastyczny program liniowy rozwiązujemy przy pomocy metody gradientów.

Problem lokalizacji fabryk

W przypadku 2-etapowego stochastycznego problemu lokalizacji fabryk:

- aktywacja klienta j jest zmienną losową,
- możemy otworzyć fabryki w 1'wszym etapie płacąc f_i^I,
- bądź w 2'gim etapie po zrealizowaniu scenariusza A płacąc f_i^A,
- następnie możemy podłączyć klientów do otwartych fabryk.

Problem lokalizacji fabryk

Relaksacja programu liniowego to:

$$\min \sum_{i \in \mathcal{F}} f_i^I y_i + \sum_{A \subseteq \mathcal{D}} p_A \left[\sum_{i \in \mathcal{F}} f_i^A y_{A,i} + \sum_{i \in \mathcal{F}, j \in A} c_{ij} x_{A,ij} \right],$$

$$x_{A,ij} \leq y_i + y_{A,i} \qquad \forall i \in \mathcal{F}, A \subseteq \mathcal{D}, j \in A,$$

$$\sum_{i \in \mathcal{F}} x_{A,ij} \geq 1 \qquad \forall A \subseteq \mathcal{D}, j \in A,$$

$$x_{ij}, y_i, y_{A,i} \geq 0 \qquad \forall i \in \mathcal{F}, A \subseteq \mathcal{D}, j \in A.$$

$y_i, y_{A,i}$ otwarte fabryki, a $x_{A,ij}$ połączenia.

Problem lokalizacji fabryk

Niech y będzie optymalnym ułamkowym rozwiązaniem dla 1-go etapu.

Niech (x_A, y_A) będzie optymalnym ułamkowym rozwiązaniem dla 2-go etapu przy zadanym A i x.

Pokażemy, że można otrzymać całkowitoliczbowe rozwiązanie poprzez niezależne rozwiązanie problemów deterministycznych dla 1'ego i 2'ego etapu.

Problem lokalizacji fabryk

Ustalmy scenariusz A oraz klienta $j \in A$.

Niech $F_{A,j} = \{i : x_{A,ij} > 0\}$.

Rozpiszmy $x_{A,ij} = x^I_{A,ij} + x^{II}_{A,ij}$, gdzie

$$x^I_{A,ij} \leq y_i \quad \text{oraz} \quad x^{II}_{A,ij} \leq y_{A,i}.$$

Ponieważ $x_{A,ij} \leq y_i + y_{A,i}$ to zawsze możemy dokonać takiego rozbicia.

Problem lokalizacji fabryk

Zauważmy teraz, że j musi być przypisane powyżej $\frac{1}{2}$ przez $\{x^I_{A,ij}\}$ bądź $\{x^{II}_{A,ij}\}$.

W pierwszym przypadku przypiszemy j do fabryki otwartej w 1'wszym etapie, a w drugim do fabryki otwartej w 2'gim etapie.

Dla klienta j zdefiniujmy zbiór scenariuszy $S_j = \{A \subseteq D : \sum_{i \in \mathcal{F}} x^I_{A,ij} \geq \frac{1}{2}\}$.

Problem lokalizacji fabryk

Skonstuujmy teraz instancje, w której mamy klienta (j, A) dla każdego $A \in S_j$ o ułamkowym żądaniu wynoszącym p_A.

Dla takiego problemu możemy otrzymać poprawne rozwiązanie $\hat{x}_{A,ij} = \min(1, 2x^I_{A,ij})$ oraz $\hat{y}_i = \min(1, 2y_i)$.

Zauważmy, że wartości $\tilde{y}_i$ nie zależą od zrealizowanego scenariusza.

Problem lokalizacji fabryk

W związku z tym możemy połączyć wszystkich a klientów (j, A) w jednego nadając mu ułamkowe żądanie wynoszące $\sum_{A \in S_j} p_A$.

Po takiej operacji koszt rozwiązania na pewno nie wzrósł.

Koszt otwarcia fabryk wynosi $2 \sum_{i \in \mathcal{F}} f_i^I y_i$.

A koszt podłączenia klientów wynosi $2 \sum_{i,j} \sum_{A \in S_j} p_A c_{ij} x^I_{A,ij} \leq 2 \sum_{i,j} \sum_{A \in S_j} p_A c_{ij} x_{A,ij}$.

Problem lokalizacji fabryk

Używając istnienia algorytmu 1.52 aproksymacyjnego zamieniamy rozwiązanie ułamkowe $(\hat{x}, \hat{y})$ na rozwiązanie całkowitoliczbowe $(\tilde{x}, \tilde{y})$.

Koszt otrzymanego rozwiązania wynosi
$3.04 \times \left(\sum_{i \in \mathcal{F}} f_i^I y_i + \sum_{i,j} \sum_{A \in S_j} p_A c_{ij} x_{A,ij} \right)$.

Wyznacza ono fabryki do otworzenia w 1 etapie.

Każdy j taki, że $A \in S_j$ będzie przypisany do fabryki zadanej przez $\tilde{x}$ otwartej w pierwszym etapie.

Problem lokalizacji fabryk

Aby przypisać pozostałych klientów rozwiążemy instancje problemu deterministycznego dla zbioru klientów $\{j \in A : A \notin S_j\}$.

Ponieważ $A \notin S_j$, to mamy $\sum_i x^{II}_{A,ij} \geq \frac{1}{2}$.

Teraz przypisując $\hat{x}_{A,ij} = \min(1, 2x^{II}_{A,ij})$ oraz $\hat{y}_{A,i} = \min(1, 2y_{A,i})$ otrzymujemy poprawne rozwiązanie dla tego zbioru klientów.

Problem lokalizacji fabryk

Ponownie używając algorytmu 1.52-aproksymacyjnego otrzymujemy rozwiązanie całkowitoliczbowe.

Koszt tego rozwiązania wynosi

$$3.04 \times \left(\sum_i f_i^A y_{A,i} + \sum_{i,j \in A : A \notin S_j} c_{ij} x_{A,ij}. \right)$$

Rozwiązanie to mówi nam jakie fabryki musimy otworzyć w 2'gim etapie i jakich klientów będziemy do nich podłączać.

Problem lokalizacji fabryk

Całkowity koszt naszego rozwiązania to:

$$3.04 \times \left(\sum_i f_i^I y_i + \sum_{i,j} \sum_{A \in S_j} p_A c_{ij} x_{A,ij} \right) +$$

$$3.04 \times \sum_A p_A \left(\sum_i f_i^A y_{A,i} + \sum_{i,j \in A : A \notin S_j} c_{ij} x_{A,ij}. \right) \leq$$

$$3.04 \times \left(\sum_i f_i^I y_i + \sum_A p_A \left(\sum_{i \in \mathcal{F}} f_i^A y_{A,i} + \sum_{i,j \in A} c_{ij} x_{A,ij} \right) \right)$$

Problem lokalizacji fabryk

Pokazaliśmy jak z rozwiązania ułamkowego otrzymać rozwiązanie całkowitoliczbowe o koszcie 3.04 razy większym.

Nie daje to nam jednak algorytmu aproksymacyjnego ponieważ w tej redukcji potrzebujemy wiedzy o wartościach $\sum_{A \in S_j} p_A$.

Istnieją algorytmy zaaokrąglające dla problemu lokalizacji fabryk działające bez wiedzy o żądaniach klientów. Najlepszy z nich ma stała 1.858 (Swamy '04).

Plan - Wykład II

Plan jutrzejszego wykładu.

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