STATISTICAL LEARNING TECHNIQUES IN COMBINATORIAL OPTIMIZATION

Michael Khachay[1]
mkhachay@imm.uran.ru

[1]Krasovsky Institute of Mathematics and Mechanics
Ural Federal University

MMPR-17-2015, Kaliningrad
September 24, 2015

Introduction

- Combinatorial optimization and machine learning appear to be extremely close fields of the modern computer science.
- Various areas in machine learning: PAC-learning, boosting, cluster analysis, feature and model selection, etc. are continuously presenting new challenges for designers of optimization methods due to the steadily increasing demands on accuracy, efficiency, space and time complexity and so on.

CO and ML

- In many cases, learning problem can be successfully reduced to the appropriate combinatorial optimization problem: max-cut, k-means, p-median, TSP, etc.
- To this end, all the results obtained for the latter problem (approximation algorithms, polynomial-time approximation schemas, approximation thresholds) can find their application in design precise and efficient learning algorithms for the former.

CO and ML

- In many cases, learning problem can be successfully reduced to the appropriate combinatorial optimization problem: max-cut, k-means, p-median, TSP, etc.
- To this end, all the results obtained for the latter problem (approximation algorithms, polynomial-time approximation schemas, approximation thresholds) can find their application in design precise and efficient learning algorithms for the former.
- But, in this presentation, I would like to consider several examples of the inverse collaboration, where combinatorial optimization benefits from using of a ML techniques

Contents

Problem statements

Set Cover

Input: A finite *range space* (*hypergraph*) $(X, \mathcal{R})$, where $\mathcal{R} \subset 2^X$.
Required to find a family (*cover*) $\{R_1, \ldots, R_s\} \subset \mathcal{R}$ of minimum size s, s.t. $R_1 \cup \ldots \cup R_s = X$.

Hitting Set

Input: A finite *range space* (*hypergraph*) $(X, \mathcal{R})$, where $\mathcal{R} \subset 2^X$.
Required to find a subset $H \subset X$ of minimum size, s.t., for any $R \in \mathcal{R}$, $H \cap R \neq \varnothing$.

Duality

The Set Cover and Hitting Set problems a related to each other by the *duality principle*. Indeed, the dual instance can be obtained by transposing the *incidence matrix*

	x_1	x_2	x_3	$\dots$	x_n
R_1	0	0	0	$\dots$	1
R_2	1	1	0	$\dots$	1
R_3	0	1	1	$\dots$	0
		$\dots$			
R_m	1	0	1	$\dots$	0

Duality

The Set Cover and Hitting Set problems a related to each other by the *duality principle*. Indeed, the dual instance can be obtained by transposing the *incidence matrix*

	R_1	R_2	R_3	$\dots$	R_m
x_1	0	1	0	$\dots$	1
x_2	0	1	1	$\dots$	0
x_3	0	0	1	$\dots$	1
		$\dots$			
x_n	1	1	0	$\dots$	0

Complexity and approximation

- Both problems are NP-hard. The Hitting Set problem remains intractable even if $|R_i| = 2$ (Vertex Cover Problem).
- D.Johnson's Greedy algorithm (1974): add to cover the current biggest subset iteratively.
- L.Lovász linear relaxation algorithm (1975): solve the corresponding LP-relaxation. Then, round the obtained solution.
- Both algorithms have polynomial time-complexity and approximation ratio of $O(\log |X|)$.

Complexity and approximation

- Both problems are NP-hard. The Hitting Set problem remains intractable even if $|R_i| = 2$ (Vertex Cover Problem).
- D.Johnson's Greedy algorithm (1974): add to cover the current biggest subset iteratively.
- L.Lovász linear relaxation algorithm (1975): solve the corresponding LP-relaxation. Then, round the obtained solution.
- Both algorithms have polynomial time-complexity and approximation ratio of $O(\log |X|)$.

Complexity and approximation

- Both problems are NP-hard. The Hitting Set problem remains intractable even if $|R_i| = 2$ (Vertex Cover Problem).
- D.Johnson's Greedy algorithm (1974): add to cover the current biggest subset iteratively.
- L.Lovász linear relaxation algorithm (1975): solve the corresponding LP-relaxation. Then, round the obtained solution.
- Both algorithms have polynomial time-complexity and approximation ratio of $O(\log |X|)$.

Complexity and approximation

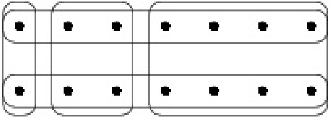
- Both problems are NP-hard. The Hitting Set problem remains intractable even if $|R_i| = 2$ (Vertex Cover Problem).
- D.Johnson's Greedy algorithm (1974): add to cover the current biggest subset iteratively.
- L.Lovász linear relaxation algorithm (1975): solve the corresponding LP-relaxation. Then, round the obtained solution.
- Both algorithms have polynomial time-complexity and approximation ratio of $O(\log |X|)$.

Tightness

- Curious, Johnson has obtained the approximation bound and proved its tightness in 1970s.
- Indeed, let, for some $p > 1$, $|X| = 2^{p+1} - 2$.
- It's easy to show that the Greedy algorithm takes all 'rectangular' ranges, i.e., $APP = p = \log(n+2) - 2$, whilst the $OPT = 2$.
- U.Feige (1998): the Set Cover can not be approximated within $(1 - o(1)) \ln |X|$ unless $NP \subset DTIME(n^{poly \log n})$.

Tightness

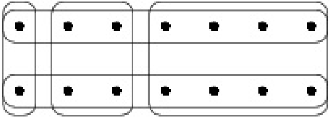
- Curious, Johnson has obtained the approximation bound and proved its tightness in 1970s.
- Indeed, let, for some $p > 1$, $|X| = 2^{p+1} - 2$.



- It's easy to show that the Greedy algorithm takes all 'rectangular' ranges, i.e., $APP = p = \log(n+2) - 2$, whilst the $OPT = 2$.
- U.Feige (1998): the Set Cover can not be approximated within $(1 - o(1)) \ln |X|$ unless $NP \subset DTIME(n^{poly \log n})$.

Tightness

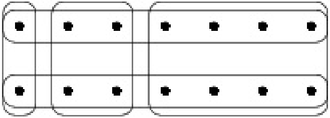
- Curious, Johnson has obtained the approximation bound and proved its tightness in 1970s.
- Indeed, let, for some $p > 1$, $|X| = 2^{p+1} - 2$.



- It's easy to show that the Greedy algorithm takes all 'rectangular' ranges, i.e., $APP = p = \log(n+2) - 2$, whilst the $OPT = 2$.
- U.Feige (1998): the Set Cover can not be approximated within $(1 - o(1)) \ln |X|$ unless $NP \subset DTIME(n^{poly \log n})$.

Tightness

- Curious, Johnson has obtained the approximation bound and proved its tightness in 1970s.
- Indeed, let, for some $p > 1$, $|X| = 2^{p+1} - 2$.



- It's easy to show that the Greedy algorithm takes all 'rectangular' ranges, i.e., $APP = p = \log(n+2) - 2$, whilst the $OPT = 2$.
- U.Feige (1998): the Set Cover can not be approximated within $(1 - o(1)) \ln |X|$ unless $NP \subset DTIME(n^{poly \log n})$.

Can we improve this approximation ratio?

- For the general case of the problem, obviously, no!
- But, in real-life applications (e.g., wireless sensor cover problem), the problem can be very special
- Maybe, some of these subclasses can be approximated much better?
- Inside. Consider spaces of finite VC-dimension and boosting-like optimization procedures

Can we improve this approximation ratio?

- For the general case of the problem, obviously, no!
- But, in real-life applications (e.g., wireless sensor cover problem), the problem can be very special
- Maybe, some of these subclasses can be approximated much better?
- Inside. Consider spaces of finite VC-dimension and boosting-like optimization procedures

Can we improve this approximation ratio?

- For the general case of the problem, obviously, no!
- But, in real-life applications (e.g., wireless sensor cover problem), the problem can be very special
- Maybe, some of these subclasses can be approximated much better?
- Inside. Consider spaces of finite VC-dimension and boosting-like optimization procedures

Can we improve this approximation ratio?

- For the general case of the problem, obviously, no!
- But, in real-life applications (e.g., wireless sensor cover problem), the problem can be very special
- Maybe, some of these subclasses can be approximated much better?
- Inside. Consider spaces of finite VC-dimension and boosting-like optimization procedures

Can we improve this approximation ratio?

- For the general case of the problem, obviously, no!
- But, in real-life applications (e.g., wireless sensor cover problem), the problem can be very special
- Maybe, some of these subclasses can be approximated much better?
- Inside. Consider spaces of finite VC-dimension and boosting-like optimization procedures

Notation

ε-net [Haussler and Welzl, 1987]

Let $(X, \mathcal{R})$ be a finite range space. A subset $N \subset X$ is called ε-net for $\mathcal{R}$ if $N \cap R \neq \varnothing$ for any R such that $|R| \geq \varepsilon|X|$.

- The concept of ε-net can be easily extended to the case of weighted sets.
- Indeed, let $w : 2^X \to \mathbb{R}_+$ be an arbitrary measure on X. In this case, a subset N is called ε-net if $N \cap R \neq \varnothing$ for any $R \in \mathcal{R}$, s.t. $w(R) \geq \varepsilon w(X)$.

net finder and verifier

For a given non-decreasing function s an algorithm $\mathcal{NF}(s)$ is called a *net-finder* of size s for $(X, \mathcal{R})$ if, for any $\varepsilon \in (0, 1)$ and any measure w, it finds ε-net of size $s(1/\varepsilon)$.
A *verifier* is an algorithm $\mathcal{V}$ that, given a subset $H \subset X$, either states (correctly) that H is a hitting set, or returns a nonempty set $R \in \mathcal{R}$ such that $R \cap H = \varnothing$.

Notation

ε-net [Haussler and Welzl, 1987]

Let $(X, \mathcal{R})$ be a finite range space. A subset $N \subset X$ is called ε-net for $\mathcal{R}$ if $N \cap R \neq \varnothing$ for any R such that $|R| \geq \varepsilon|X|$.

- The concept of ε-net can be easily extended to the case of weighted sets.
- Indeed, let $w : 2^X \to \mathbb{R}_+$ be an arbitrary measure on X. In this case, a subset N is called ε-net if $N \cap R \neq \varnothing$ for any $R \in \mathcal{R}$, s.t. $w(R) \geq \varepsilon w(X)$.

net finder and verifier

For a given non-decreasing function s an algorithm $\mathcal{NF}(s)$ is called a *net-finder* of size s for $(X, \mathcal{R})$ if, for any $\varepsilon \in (0, 1)$ and any measure w, it finds ε-net of size $s(1/\varepsilon)$.
A *verifier* is an algorithm $\mathcal{V}$ that, given a subset $H \subset X$, either states (correctly) that H is a hitting set, or returns a nonempty set $R \in \mathcal{R}$ such that $R \cap H = \varnothing$.

Notation

ε-net [Haussler and Welzl, 1987]

Let $(X, \mathcal{R})$ be a finite range space. A subset $N \subset X$ is called ε-net for $\mathcal{R}$ if $N \cap R \neq \varnothing$ for any R such that $|R| \geq \varepsilon|X|$.

- The concept of ε-net can be easily extended to the case of weighted sets.
- Indeed, let $w : 2^X \to \mathbb{R}_+$ be an arbitrary measure on X. In this case, a subset N is called ε-net if $N \cap R \neq \varnothing$ for any $R \in \mathcal{R}$, s.t. $w(R) \geq \varepsilon w(X)$.

net finder and verifier

For a given non-decreasing function s an algorithm $\mathcal{NF}(s)$ is called a *net-finder* of size s for $(X, \mathcal{R})$ if, for any $\varepsilon \in (0, 1)$ and any measure w, it finds ε-net of size $s(1/\varepsilon)$.

A *verifier* is an algorithm $\mathcal{V}$ that, given a subset $H \subset X$, either states (correctly) that H is a hitting set, or returns a nonempty set $R \in \mathcal{R}$ such that $R \cap H = \varnothing$.

Notation

ε-net [Haussler and Welzl, 1987]

Let $(X, \mathcal{R})$ be a finite range space. A subset $N \subset X$ is called ε-net for $\mathcal{R}$ if $N \cap R \neq \varnothing$ for any R such that $|R| \geq \varepsilon |X|$.

- The concept of ε-net can be easily extended to the case of weighted sets.
- Indeed, let $w : 2^X \to \mathbb{R}_+$ be an arbitrary measure on X. In this case, a subset N is called ε-net if $N \cap R \neq \varnothing$ for any $R \in \mathcal{R}$, s.t. $w(R) \geq \varepsilon w(X)$.

net finder and verifier

For a given non-decreasing function s an algorithm $\mathcal{NF}(s)$ is called a *net-finder* of size s for $(X, \mathcal{R})$ if, for any $\varepsilon \in (0, 1)$ and any measure w, it finds ε-net of size $s(1/\varepsilon)$.
A *verifier* is an algorithm $\mathcal{V}$ that, given a subset $H \subset X$, either states (correctly) that H is a hitting set, or returns a nonempty set $R \in \mathcal{R}$ such that $R \cap H = \varnothing$.

Multiplicative wights update algorithm

Assume, for the time being, that we know the size $c = OPT$ of a smallest hitting set.

1. Initialize w on X by uniform measure
2. Using $\mathcal{NF}$, for $(X, \mathcal{R})$, find a $1/(2c)$-net N of size $s(2c)$
3. Using $\mathcal{V}$ check if N is a hitting set. If it is, then STOP
4. Else double the weights of points of the found subset R and return to step 2

Theorem 1

If there is a hitting set of size c, the MWUA cannot iterate more than $4c\log(n/c)$ times, and the total weight will not exceed n^4/c^3, where $n = |X|$.

Multiplicative wights update algorithm

Assume, for the time being, that we know the size $c = OPT$ of a smallest hitting set.

1. Initialize w on X by uniform measure
2. Using $\mathcal{NF}$, for $(X, \mathcal{R})$, find a $1/(2c)$-net N of size $s(2c)$
3. Using $\mathcal{V}$ check if N is a hitting set. If it is, then STOP
4. Else double the weights of points of the found subset R and return to step 2

Theorem 1

If there is a hitting set of size c, the MWUA cannot iterate more than $4c\log(n/c)$ times, and the total weight will not exceed n^4/c^3, where $n = |X|$.

Multiplicative wights update algorithm

Assume, for the time being, that we know the size $c = OPT$ of a smallest hitting set.

1. Initialize w on X by uniform measure
2. Using $\mathcal{NF}$, for $(X, \mathcal{R})$, find a $1/(2c)$-net N of size $s(2c)$
3. Using $\mathcal{V}$ check if N is a hitting set. If it is, then STOP
4. Else double the weights of points of the found subset R and return to step 2

Theorem 1

If there is a hitting set of size c, the MWUA cannot iterate more than $4c\log(n/c)$ times, and the total weight will not exceed n^4/c^3, where $n = |X|$.

Multiplicative wights update algorithm

Assume, for the time being, that we know the size $c = OPT$ of a smallest hitting set.

1. Initialize w on X by uniform measure
2. Using $\mathcal{NF}$, for $(X, \mathcal{R})$, find a $1/(2c)$-net N of size $s(2c)$
3. Using $\mathcal{V}$ check if N is a hitting set. If it is, then STOP
4. Else double the weights of points of the found subset R and return to step 2

Theorem 1

If there is a hitting set of size c, the MWUA cannot iterate more than $4c\log(n/c)$ times, and the total weight will not exceed n^4/c^3, where $n = |X|$.

Multiplicative wights update algorithm

Assume, for the time being, that we know the size $c = OPT$ of a smallest hitting set.

1. Initialize w on X by uniform measure
2. Using $\mathcal{NF}$, for $(X, \mathcal{R})$, find a $1/(2c)$-net N of size $s(2c)$
3. Using $\mathcal{V}$ check if N is a hitting set. If it is, then STOP
4. Else double the weights of points of the found subset R and return to step 2

Theorem 1

If there is a hitting set of size c, the MWUA cannot iterate more than $4c \log(n/c)$ times, and the total weight will not exceed n^4/c^3, where $n = |X|$.

Multiplicative wights update algorithm

Assume, for the time being, that we know the size $c = OPT$ of a smallest hitting set.

1. Initialize w on X by uniform measure
2. Using $\mathcal{NF}$, for $(X, \mathcal{R})$, find a $1/(2c)$-net N of size $s(2c)$
3. Using $\mathcal{V}$ check if N is a hitting set. If it is, then STOP
4. Else double the weights of points of the found subset R and return to step 2

Theorem 1

If there is a hitting set of size c, the MWUA cannot iterate more than $4c\log(n/c)$ times, and the total weight will not exceed n^4/c^3, where $n = |X|$.

Proof sketch

- let H be an optimal hitting set of size c
- any time when $\mathcal{V}$ returns subset R, $w(R) < w(X)/(2c)$. Hence, $w(X)$ increases at most by $(1 + 1/(2c))$ in any iteration and, after k iterations,
$$w(X) \leq n(1 + \frac{1}{2c})^k \leq ne^{\frac{k}{2c}}$$
- by assumption, $H \cap R = \varnothing$, therefore, at any iteration, there exist a point $h \in H$ to double a weight
- let, after k iterations, each point $h \in H$ has measure 2^{z_h}
- then,
$$w(H) = \sum_{h \in H} 2^{z_h}, \sum_{z_h} \geq k$$
- or, by convexity of the exponential function, $w(H) \geq c2^{k/c}$.
- finally,
$$c2^{k/c} \leq w(H) \leq w(X) \leq ne^{\frac{k}{2c}}$$
and $k \leq 4c \log(n/c)$.

Accuracy and time-complexity

- time complexity bound is $4c\log(n/c) = O(n\log n)(TB(\mathcal{NF}) + TB(\mathcal{V}))$
- But what about approximation ratio?
- It can be achieved for $(X, \mathcal{R})$ of a fixed VC-dimension

VC-dimension

A subset $Y \subset X$ is called *shattered* by $\mathcal{R}$ if factor set $\mathcal{R} \setminus Y = 2^Y$. A number d is called VC-dimension of the range space $(X, \mathcal{R})$ if the largest shattered subset $Y \subset X$ has $|Y| \leq d$.

Therefore, the MWUA has approximation ratio of $O(\log c)$, not $O(logn)$. (Example!)

Accuracy and time-complexity

- time complexity bound is $4c\log(n/c) = O(n\log n)(TB(\mathcal{NF}) + TB(\mathcal{V}))$
- But what about approximation ratio?
- It can be achieved for $(X, \mathcal{R})$ of a fixed VC-dimension

VC-dimension

A subset $Y \subset X$ is called *shattered* by $\mathcal{R}$ if factor set $\mathcal{R} \setminus Y = 2^Y$. A number d is called VC-dimension of the range space $(X, \mathcal{R})$ if the largest shattered subset $Y \subset X$ has $|Y| \leq d$.

Therefore, the MWUA has approximation ratio of $O(\log c)$, not $O(logn)$. (Example!)

Accuracy and time-complexity

- time complexity bound is $4c\log(n/c) = O(n\log n)(TB(\mathcal{NF}) + TB(\mathcal{V}))$
- But what about approximation ratio?
- It can be achieved for $(X, \mathcal{R})$ of a fixed VC-dimension

VC-dimension

A subset $Y \subset X$ is called *shattered* by $\mathcal{R}$ if factor set $\mathcal{R} \setminus Y = 2^Y$.
A number d is called VC-dimension of the range space $(X, \mathcal{R})$ if the largest shattered subset $Y \subset X$ has $|Y| \leq d$.

Therefore, the MWUA has approximation ratio of $O(\log c)$, not $O(logn)$. (Example!)

Accuracy and time-complexity

- time complexity bound is
  $4c\log(n/c) = O(n\log n)(TB(\mathcal{NF}) + TB(\mathcal{V}))$
- But what about approximation ratio?
- It can be achieved for $(X, \mathcal{R})$ of a fixed VC-dimension

VC-dimension

A subset $Y \subset X$ is called *shattered* by $\mathcal{R}$ if factor set $\mathcal{R} \setminus Y = 2^Y$.
A number d is called VC-dimension of the range space $(X, \mathcal{R})$ if the largest shattered subset $Y \subset X$ has $|Y| \leq d$.

Therefore, the MWUA has approximation ratio of $O(\log c)$, not $O(logn)$. (Example!)

Accuracy and time-complexity

- time complexity bound is $4c\log(n/c) = O(n\log n)(TB(\mathcal{NF}) + TB(\mathcal{V}))$
- But what about approximation ratio?
- It can be achieved for $(X, \mathcal{R})$ of a fixed VC-dimension

VC-dimension

A subset $Y \subset X$ is called *shattered* by $\mathcal{R}$ if factor set $\mathcal{R} \setminus Y = 2^Y$.
A number d is called VC-dimension of the range space $(X, \mathcal{R})$ if the largest shattered subset $Y \subset X$ has $|Y| \leq d$.

Theorem 2 (Brönnimann, Chazelle, Matoušhek, 1993)

For a range space $(X, \mathcal{R})$ of finite VC-dimension d, there is $1/\varepsilon$-net of size $\frac{d}{\varepsilon}\log\frac{d}{\varepsilon}$

Therefore, the MWUA has approximation ratio of $O(\log c)$, not $O(logn)$. (Example!)

Definitions and Notation

Committee decision rule (CDR)

Suppose $X \subset \mathbb{R}^n$, $f_1, \ldots, f_q : X \to \mathbb{R}$ — affine functions. *Committee decision rule* is a $f_1, \ldots, f_q$ is a partial function $\varphi : X \to \Omega$, defined by

$$\varphi(x) = \begin{cases} 1, & \text{if } \sum_{j=1}^{q} \operatorname{sign}(f_j(x)) > 0, \\ 0, & \text{if } \sum_{j=1}^{q} \operatorname{sign}(f_j(x)) < 0, \\ \Delta, & \text{otherwise.} \end{cases}$$

CDR φ is called *correct* on the sample ξ, if

$$\varphi(x_i) = \omega_i \qquad (i \in \mathbb{N}_m).$$

MASC problem

Affine separating committee

Let $f_1, \ldots, f_q : \mathbb{R}^n \to \mathbb{R}$ be affine functions, and $A, B \subset \mathbb{R}^n$. A finite sequence $K = (f_1, \ldots, f_q)$ is called *affine committee separating A and B*, if

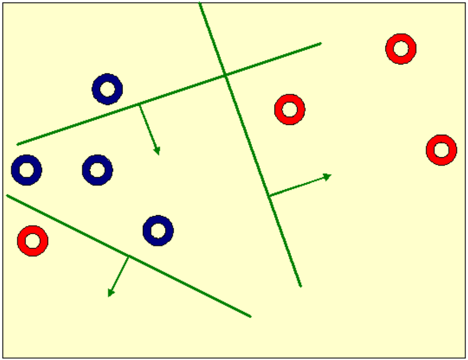
$$\begin{aligned} |\{i \in \mathbb{N}_q : \; f_i(a) > 0\}| &> \frac{q}{2} \qquad (a \in A), \\ |\{i \in \mathbb{N}_q : \; f_i(b) < 0\}| &> \frac{q}{2} \qquad (b \in B). \end{aligned}$$

The number q is called *a length* of K, and the sets A and B — *separatable* by K.

'Minimum Affine Separating Committee (MASC) Problem'

For given finite subsets $A, B \subset \mathbb{Q}^n$ it is required to find an affine separating committee K of minimum length.

MASC problem



- $n = 2$
- set A consists of red points, B — blue pts
- $q_{min} = 3$
- one of the minimum ASC is presented

Complexity and approximation

Theorem

MASC is NP-hard (in the strong sense) and remains intractable whether $A \cup B \subset \{x \in \{0,1,2\}^n \,:\, |x| \leq 2\}$.
ASC is NP-complete and remains intractable for each fixed $q \geq 3$.

Theorem

BGC is correct approximation algorithm for MASC-GP(n) with ratio

$$\frac{\mathrm{BGC}(A,B)}{\mathrm{OPT}(A,B)} \leq \lceil 2\bar{m} \ln((m+1)/2) \rceil^{1/2}, \ \bar{m} = 2\left\lceil \frac{\lfloor (m-n)/2 \rfloor}{n} \right\rceil + 1$$

and time complexity $O(m^{n+3}/n \ln m) + \Theta_{GC}$.

Thank you for your attention!