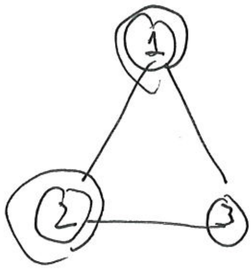
# Approximation and Online Algorithms with Applications

# 5.

2 weeks ago,

Vertex Cover Problem



Linear Program (LP).

$$\min \sum_{i=1}^{n} x_i$$

such that $x_i + x_j \geq 1$ for all $\{i,j\} \in E$

Alg

1: Solve the LP

optimal solution $x_1^*, \dots x_n$.

2: For $i \in \{1, \dots, n\}$

$$x_i' = \begin{cases} 1 & x_i \geq 0.5 \\ 0 & \text{otherwise} \end{cases}$$

3: return $S = \{ i : x_i' \underset{=1}{\cancel{\geq 0.5}} \}$

Theorem Alg returns a vertex cover.

Proof We will show that

$\forall \{i,j\} \in E$, either $i$ or $j \in S$.

either $x_i'$ or $x_j' = 1$

either $x_i^*$ or $x_j \geq 0.5$.

$\Updownarrow$

true because $x_i + x_j \geq 1$

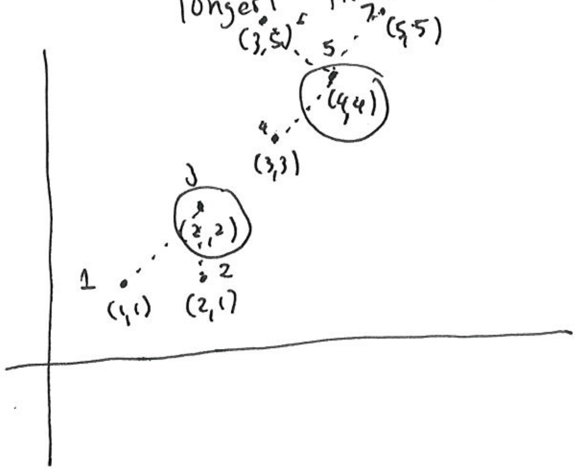
k-center problem

- We want to put k ward offices.
- We want to minimize time to get to the ward offices

↓

longest time to get to the ward offices.

Example.




$k=2.$

Put ward offices to (1,1) and (2,1)

(5,5) to ward offices

↳ go to (2,1) $\quad$ distance $= \sqrt{(5-2)^2+(5-1)^2}$

$= \sqrt{3^2+4^2} = \sqrt{25} = 5.$

Put ward offices to (2,2) and (4,4)

all persons can reach ward offices with distance

$\sqrt{2} \approx 1.4$

Two clusters $\{(1,1),(2,1),(2,2)\}$ and

$\{(3,3),(3,5),(4,4),(5,5)\}.$

## Optimization Model

**Input:** # persons $n$ $\{1, \dots, n\}$ : set of persons

$p_1, \dots, p_n$ : positions of each person , $k$: # ward offices.

**Output:** Positions to build ward offices $S \subseteq \{1, \dots, n\}$.

**Constraint:** $|S| = k$

**Objective Function:** $d_{ij}$ : distance for person $i$ to position $j$.

$D_i^{(s)} := \min_{j \in S} d_{ij}$ : position for person $i$ to get to the closest ward office in $S$.

$\mathrm{Min}\left(\mathrm{Max}_i\, D_i^{(s)}\right)$ : maximum distance that a person need to be at a ward office.

---

## Algorithm

1: $S \leftarrow \{v_1\}$ where $v_1$ is a random person.

2: for $j = 2$ to $k$:

$v_j = \arg\max_i D_i^{(s)}$ [ Select a person that is farthest from selected ward office. ]

$S \leftarrow S \cup \{v_j\}$ [ Add that farthest person to the set. ]

Example $v_1 = 1$ $(p_1 = (1,1))$

Step 2. $S = \{1\}$.

$D_2^{(S)} = 1$ $D_3^{(S)} = \sqrt{2}$ $D_4^{(S)} = 2\sqrt{2}$ $D_5^{(S)} = 3\sqrt{2}$

$D_6^{(S)} = 2\sqrt{5}$ $D_7^{(S)} = 4\sqrt{2}$.

Farthest person 7

$\therefore$ Add 7 to the set. i.e. $v_2 = 7$

$S = \{1, 7\}$ → Solution obtained from the algorithm

---

$D_1^{(S)} = 0$, $D_2^{(S)} = 1$, $D_3^{(S)} = \sqrt{2}$, $\underbrace{D_4^{(S)} = 2\sqrt{2}}$ $D_5^{(S)} = \sqrt{2}$ $D_6^{(S)} = 2$,

$D_7^{(S)} = 0$

Objective Value for our solution

$SOL = 2\sqrt{2}$

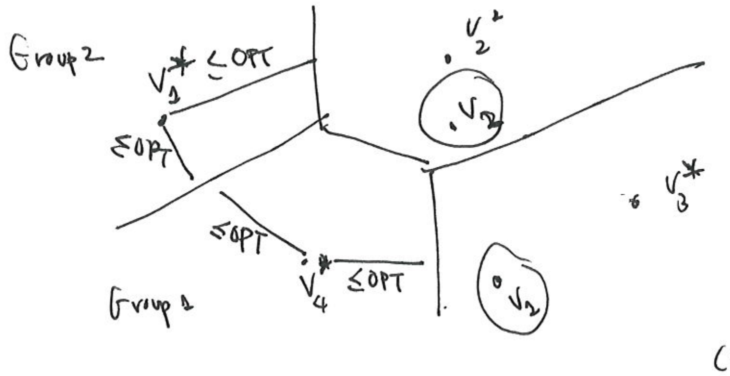
Optimal solution $S^* = \{3, 5\}$

optimal value $OPT = \sqrt{2}$

---

Theorem: The algorithm is a 2-approximation algorithm for k-center problem, i.e.

$$SOL \leq 2 \cdot OPT.$$




In the optimal solution, distance from all ~~points~~ centers to ~~center~~ ward offices $\leq$ OPT.

Distance between two points in the same group $\leq 2 \cdot$ OPT.

---


Case 1 : $d_{v_1, v_2} \leq 2 \cdot \mathrm{OPT}$

$\max_i D_i^{(S)} \leq 2 \cdot \mathrm{OPT}$ because $\max_i D_i^{(S)} = D_{v_2}^{(S)}$

$D_i^{(S)} \leq 2 \cdot \mathrm{OPT}$ for all person $i$.

When we add $v_3, \ldots, v_k$ to the set $S$.

$D_i^{(S)}$ will get smaller. for all person $i$.

At the end, all $D_i^{(S)} \leq 2 \cdot \mathrm{OPT}$.

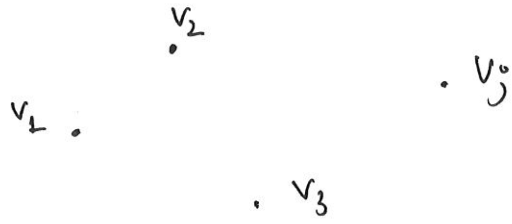
$\mathrm{SOL} = \max D_i^{(S)} \leq 2 \cdot \mathrm{OPT}$

$\mathrm{SOL} \leq 2 \cdot \mathrm{OPT}$.

Case 2 : $d_{v_1, v_2} > 2 \cdot \mathrm{OPT}$

$v_1$ and $v_2$ is in different groups. → (continue our analysis.)

At arbritary $j$:




Case 1 $\quad \exists_v \; D^{(s)}_{v_j} \leq 2 \cdot OPT.$

$SOL = \max_p D^{(s)}_p \leq 2 \cdot OPT.$ at this step, and also at last step.

$$SOL \leq 2 \cdot OPT.$$

Case 2. $\quad v_1, v_2, \ldots, v_j$ are all in different groups. (continue our analysis.)

If there is no $j$ such that $D^{(s)}_{v_j} \leq 2 \cdot OPT$,

$v_1, v_2, \ldots, v_k$ are all in different groups.

Distance between two points in the same group $\leq 2 \cdot OPT$.

Distance from a point to $v_i$ (which is in the same group)

$\leq 2 \cdot OPT$.

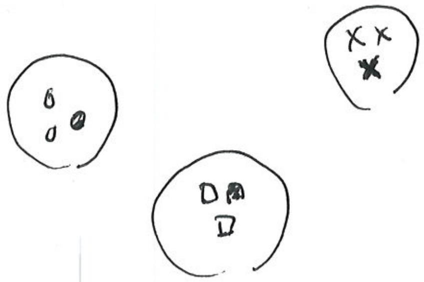
Distance for a person to any ward offices $\leq 2 \cdot OPT$.

$$SOL = \max D^{(s)}_p \leq 2 \cdot OPT$$

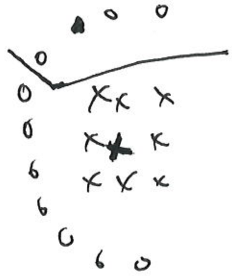
$$SOL \leq 2 \cdot OPT \qquad \square$$

# Kernel Method [Cortés and Scott, ICASSP'14]

Easy case → can use k-center

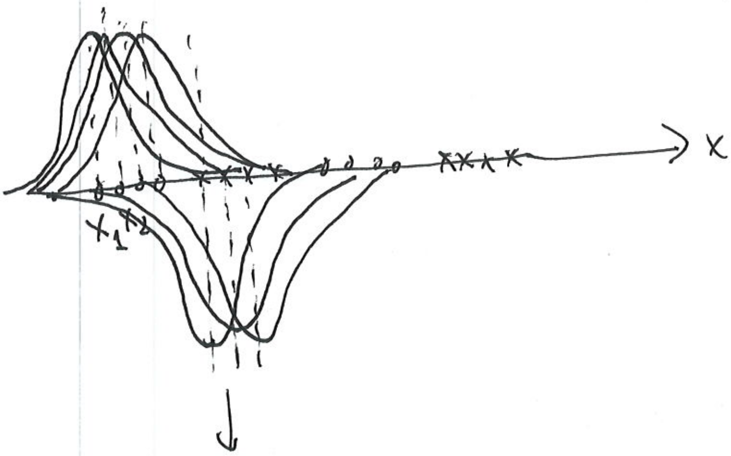


difficult case



wrong classification for any k-center selection

Let see it in 1-dimensional case




Gussian Kernel

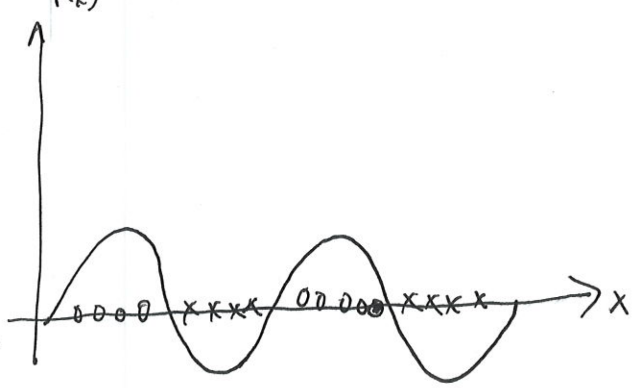
$$k(x, x_p) = \frac{1}{(\sqrt{2\pi}\sigma)^d} e^{-\frac{|x - x_p|^2}{\sigma^2}}$$

a parameter

\# dimension

$$f(x) = \sum_{i=1}^{n} \frac{k(x, x_i)}{n}$$

average all the curve together




If f(●) ≥ 0 :

● could be o.

Else

● could be x.

Lecture 6: Materials, May 23rd

Approximation Algorithms: Clustering Problem

1. Announcement

   Contents to be included in quiz on next week (May 30th)
   (If you understand our content, then reminding yourself by reading our lecture note is enough for the quiz. The reading list below is required for students who cannot understand the content or want to know more detail about the topics.

   - Optimization Models and Linear Programming
     Reading: "*Formulating an Optimization Model: An Introductory Example*" (http://www.4er.org/CourseNotes/Book%20A/A-I.pdf), Page A-1 to A-17.
   - NP-Hardness
     Reading: "*Computers and Intractability*: A Guide to the Theory of NP-Completeness", Pages 1-11.
   - Approximation Algorithm for Knapsack Problem
     Reading: "*The Design of Approximation Algorithms*", Pages 13-16.
     Reading: "*15-854 Approximation Algorithms: Lecture 10 - Dynamic Programming*", Pages 1-2.
   - Approximation Algorithm for Vertex Cover Problem
     Reading: "*The Design of Approximation Algorithms*", Pages 16-20.

2. Approximation Algorithm for clustering problem
   Our main textbook for the first half of this course is the following book.
   Williamson and Shmoys, "*The Design of Approximation Algorithms*", Cambridge University Press, 2010.

   The book can be downloaded for free from the following URL.
   http://www.designofapproxalgs.com/book.pdf
   The content for $k$-center problem can be found at Chapter 2.2, and the content for $k$-median problem (last week) can be found at Chapter 9.2.

3. Sparse Approximation of Kernel Methods
   Contents we have discussed in this class can be found in the following papers.

   Cortés and Scott, "*Scalable Sparse Approximation of a Sample Mean*", Proceedings of the 39th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2014), pages 5274-5278, 2014.

   Cortés and Scott, "*Sparse Approximation of a Kernel Mean*", IEEE Transactions on Signal Processing, Vol. 65, No. 5, pages 1310-1323, 2017.

Problem : To justify the cluster of ●, we have to calculate

$$f(●) = \sum_{i=1}^{n} \frac{k(x, x_i)}{n} = \frac{1}{n}\sum_{i=1}^{n} \frac{e^{-\frac{|x-x_i|^2}{\sigma^2}}}{(\sqrt{2\pi}\sigma)^d}$$

we have to calculate this (slow computation for Gaussian)

for $n$ times!

→ very slow when $n$ is large!! (100,000)

This paper: Speed up the classification in the kernel method.

- Minimize # Guassian calculation for each classification $f$.
- Select $S \subseteq \{1, \dots, n\}$

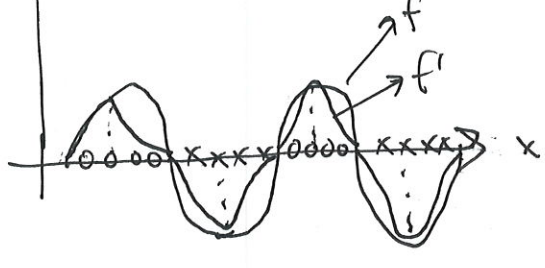
Approximate $f$

$$f(x) = \frac{1}{n}\sum_{i=1}^{n} k(x, x_i) \longrightarrow f'(x) = \sum_{i \in S} \alpha_i k(x, x_i).$$

$$\# \text{ Guassian calculation} = |S| = k \ll n.$$

---

Example




We can approximate $f$ (16 Guassian calculations)

with $f'$ (14 Guassian calculations).

Approximation error: $err := \sqrt{\int_x (f(x) - f'(x))^2 dx}$

<u>Theorem</u> $\quad err \leq \left(1 - \frac{k}{n}\right) \boxed{\sqrt{1 - r_s^2}}$

~~very small~~
very close to 1
when $k \ll n$.
(~~for~~ $k$ is constant)

Small when $r_s$ close to 1
large when $r_s$ close to 0.

$\therefore$ err is small when $r_s$ is large?

<u>Aim</u>: Find $S$ that maximize $r_s$.

<u>Definition of $r_s$.</u>

$$\langle i, j \rangle = \int_x [k(x, x_i)\, k(x, x_j)]\, dx$$

This value if large when $k(x, x_i)$ is close to $k(x, x_j)$

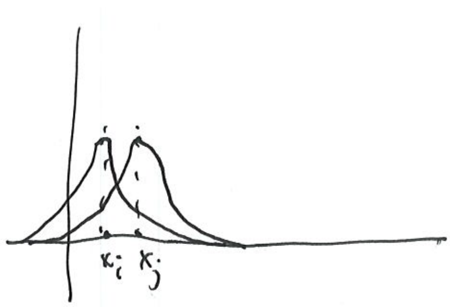
$$= \int_x \frac{e^{\frac{-|x-x_i|^2}{\sigma^2}}}{(\sqrt{2\pi}\,\sigma)^d} \cdot \frac{e^{\frac{-|x-x_j|^2}{\sigma^2}}}{(\sqrt{2\pi}\,\sigma)^d}\, dx$$

$$= \frac{e^{\frac{-|x_i-x_j|^2}{\sigma^2}}}{(\sqrt{2\pi}\,\sigma)^d}$$

$\langle i, j \rangle$ will be large if $x_i$ and $x_j$ is closed together

<u>Conclusion</u>

$\langle i, j \rangle$ is a value indicating how $k(x, x_i)$ is close to $k(x, x_j)$.
The value ~~is large~~ depends on the distance between $x_i$ and $x_j$.



$$\gamma_{S,j} = \max_{i \in S} \langle i, j \rangle \to i \text{ is the closest point to } j \text{ in } S.$$

$\to$ $\gamma_{S,j}$ is the distance from $j$ to the closest point in $S$.

- We want all $x_j$ to be as close as $S$ as possible.

$$\gamma_S = \min_j \gamma_{S,j} \to j \text{ is the farthest point from } S.$$

$\to$ $\gamma_S$ is the distance from the farthest point from $S$.

---

Conclusion

We want to maximize $\gamma_S = \min_j \gamma_{S,j} = \min_j \max_{i \in S} \langle i, j \rangle$

$$= \min_j \max_{i \in S} \frac{e^{\frac{-|x_i - x_j|^2}{\sigma^2}}}{(\sqrt{2\pi}\,\sigma)^d}$$

$e^x$ grows in the same way as $x$

constant

$$= \min_j \max_{i \in S} \left( \cancel{\frac{e^{\frac{-|x_i - x_j|^2}{\sigma^2}}}{(\sqrt{2\pi}\,\sigma)^d}} \; \frac{-|x_i - x_j|^2}{\sigma^2} \right)$$

$x^2$ grows in the same way as $x$

constant

$$= \min_j \max_{i \in S} \left( -|x_i - x_j| \right).$$

$$= \min_j \left( - \min_{i \in S} |x_i - x_j| \right)$$

$$= -\max_j \min_{i \in S} |x_i - x_j|$$

maximize $-\max_j \min_{i \in S} |x_i - x_j|.$

minimize $\max_j \min_{i \in S} |x_i - x_j|.$

$$\text{minimize} \quad \max_{j} \min_{i \in S} |x_i - x_j| \quad d_{i,j}$$

$$= \max_{j} \min_{i \in S} d_{i,j}$$

$$= \max_{j} D_S^{(j)}$$

$$\text{minimize} = D_S \longrightarrow \text{k-center problem}$$

---

## Optimization Model

**Input**: # n sampling points

$p_1, \ldots, p_n$: positions of points

**Output**: $S \subseteq \{1, \ldots, n\}$

**Constraint**: $|S| = k$

**Objective Function**: Minimize error in approximating kernel.

$\Updownarrow$ by theorem

Maximize $V_S$.

$\Updownarrow$

k-center problem (2-approximation algorithm).