Final Review

Mathematical Foundations for the Information Age

Peking University

December 26th, 2024

Contents

Contents

Instructions

- **Thursday, January 9th 14:00-16:30**
- **Room 422, No.2 Teaching Building**
- **Closed-book exam**
- No paper materials or electronic devices are allowed. You need to take your **student ID card** to verify your identity.
- Contents of the entire semester will be covered in the final exam.

Instructions

- This exam consists of about 7 problems.
- All problems are given in English. You can raise your hand to ask TA to translate certain terms that you do not understand.
- You are allowed to write your answers in Chinese, English, or a combination of both languages.
- Please clearly indicate the problem numbers before your answers.
- Please manage your time wisely.

Instructions

- Problem 1: Fill in the blanks. About 15pts.
    - **You don't need to prove your results in this problem.**
    - Basic definitions, properties, applications in the course.
    - Sample problem: Calculate the surface area of the unit ball.
- Problem 2-7: Problem solving. About 85pts.
- Major topics: Machine Learning, Streaming Algorithms, Random Graph. (There may still be several problems about high dimensional geometry and singular value decomposition!)

Contents

High Dimensional Geometry

- Properties for unit ball in $\mathbb{R}^d$.
  - Volume and surface area.
  - Concentration properties.
  - Relations with high dimensional Gaussian random variables. (How to sample uniformly in the unit ball?)
- Johnson-Lindenstrauss Lemma.

Contents

Singular Value Decomposition

- Definition and geometric interpretation.
- Best fit subspace and "greedy" construction.
- Low rank approximations: F-norm, 2-norm.
- Left singular vectors and its properties.
- Relations with the eigen decomposition of $\boldsymbol{A}^\top \boldsymbol{A}$.
- Power method.
- Centering data.

Contents

Perceptron Algorithm

- Algorithm procedure.
- We need to add an extra coordination to the original data. (Why?)
- Theoretical justification. (Condition: linearly separable.)
- Kernel perceptron algorithm.

Kernel Function

- Definition.
- Kernel matrix $\boldsymbol{K}(\boldsymbol{x}_i, \boldsymbol{x}_j) \iff$ positive semi-definite matrix. (Note: not necessarily positive definite!)
- If k_1, k_2 are kernel functions, then $k_1 + k_2, k_1 \cdot k_2, f(\boldsymbol{x})f(\boldsymbol{y})k_1(\boldsymbol{x}, \boldsymbol{y})$ are all kernel functions.

VC Dimension

- Definition. ($\forall$ or $\exists$?)
- Shatter function / Growth function: $\pi_{\mathcal{H}}(n)$.
  - $\pi_{\mathcal{H}}(n) \le \sum_{i=0}^{d} \binom{n}{i}$, where d is the VC Dimension of $\mathcal{H}$.
  - $\pi_{\mathcal{H}_1 \cap \mathcal{H}_2}(n) \le \pi_{\mathcal{H}_1}(n) \pi_{\mathcal{H}_2}(n)$.
- VC Dimension for several hypothesis classes: linear separator, convex set, ...

Uniform Convergence and Generalization Bound

- Finite hypothesis class: union bound ($+$ concentration analysis). (Chapter 5.4)
- Infinite hypothesis class. (Theorem 5.14)

Online Learning

- Problem formulation.
- Halving algorithm, (randomized) weighted majority algorithm.
- Potential function method.

Boosting Algorithm

Algorithm 2: Boosting algorithm

Input: Number of iterations M (where M is odd), a sample S of n labeled examples $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n$ with labels $y_1, \cdots, y_n$, a γ-weak ($\gamma > 0$) learner (i.e., an algorithm that given n labeled examples and a non-negative weight $\boldsymbol{w} \in \mathbb{R}^n$, gives an hypothesis with at least $\frac{1}{2} + \gamma$ accuracy on the weight $\boldsymbol{w}$).

$\boldsymbol{w}_1 \leftarrow (1, 1, \cdots, 1)$ ▷ Initialize each example $\boldsymbol{x}_i$ to have a weight $\boldsymbol{w}_1(i) = 1$.

for $t = 1, 2, \cdots, M$ **do**

  Call the γ-weak learner on the sample S with weight $\boldsymbol{w}_t$ to get the hypothesis h_t.

  for $i = 1, 2, \cdots, n$ **do**

    if $\boldsymbol{h}_t(x_i) \neq y_i$ **then**

      $\boldsymbol{w}_{t+1}(i) \leftarrow \boldsymbol{w}_t(i) \cdot \frac{\frac{1}{2}+\gamma}{\frac{1}{2}-\gamma}$

    else

      $\boldsymbol{w}_{t+1}(i) = \boldsymbol{w}_t(i)$

    end

  end

end

Output: The classifier $\mathrm{Maj}(h_1, \cdots, h_M)$.

Contents

Outline

- Streaming Model.
- Algorithm for random sampling of the input "on the fly".
- Majority Algorithm and Algorithm Frequent.
- Several other algorithms in class (Chapter 6).

Streaming Model

Streaming Model

n items $a_1, a_2, \ldots, a_n$ arrive one at a time.
You can never use information about $a_{t+1}, \ldots, a_n$ at time t.

- Why Streaming Model?
  - n too large, while $1 \leq a_i \leq m$ and m is not too large.
  - Some real-world scenarios are online.
- Goal: design algorithms with $poly(\log n, \log m)$ bit space.

Sampling from a Stream

Key Step:

- Suppose we have the solution at time t, now a_{t+1} comes, decide how should the solution change.

Example: Proportion to a_i

- When a_{t+1} comes,
- The probability for sampling a_{t+1} is $\dfrac{a_{t+1}}{\sum_{i=1}^{t} a_i + a_{t+1}}$
- The probability for sampling $a_i (i \leq t)$ changes from $\dfrac{a_i}{\sum_{i=1}^{t} a_i}$ to $\dfrac{a_i}{\sum_{i=1}^{t} a_i + a_{t+1}}$, becoming $\dfrac{\sum_{i=1}^{t} a_i}{\sum_{i=1}^{t} a_i + a_{t+1}}$ times.
- So we need to maintain $s = \sum_{i=1}^{t} a_i$

Sampling from a Stream

Key Step:

- Suppose we have the solution at time t, now a_{t+1} comes, decide how should the solution change.

Algorithm Frequent

Count the frequency (within an error of $n/(k+1)$) of each element of $\{1, 2, \ldots, m\}$ in the stream.

Algorithm

Maintain k counter and a k size list.
When encounter an item,

- Increment a counter
- Add the element to the list, and set counter to 1
- Decreases each counter by 1

Key:

- Whenever an counter decreases 1, the gap between the sum of all counters and the element number we already encounter increases with $k+1$.

$$f_i - \frac{n}{k+1} \leq \hat{f}_i \leq f_i.$$

Contents

Outline

- n-Universal.
- Counting number of distinct elements.

n-Universal

A set of hash functions

$$H = \{h \mid h : \{1, 2, \cdots, m\} \to \{0, 1, \cdots, M-1\}\}$$

is n-universal if $\forall x_1, \cdots, x_n$ where $x_i \in \{1, 2, \cdots, m\}$ and $x_i \neq x_j$, $\forall y_1, \cdots, y_n \in \{0, 1, \cdots, M-1\}$,

$$\mathbb{P}_{h \sim H}(\forall i \in [n], h(x_i) = y_i) = \frac{1}{M^n}.$$

Key:

- Randomness comes from h.

Examples

2-universal

$h_{ab}(x) = ax + b \ (mod \ M)$ with $a, b \in [0, M-1]$.
$h(x) = w$ and $h(y) = z$ if and only if

$$\begin{bmatrix} x & 1 \\ y & 1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} w \\ z \end{bmatrix} \ (mod \ M).$$

Here a, b are the variables to be solved.

Count Distinct Elements

Lower bound for deterministic algorithm

- Consider the number of possible states the algorithm can represent.

Nondeterministic Algorithm

Algorithm

- Keep track of the minimum of $h(a_i)$
- Use M/min as estimation

For a random set S, the expected value of the minimum is approximately $|S|+1$.

$$\frac{d}{6} \leq \frac{M}{min} \leq 6d$$

Contents

Outline

- $G(n, p)$.
- Second Moment Methods.

Second Moment Methods

Second Moment Methods

Suppose $E(X) > 0$. If $Var(X) = o(E^2(X))$, then X is almost surely greater than 0.

Basic idea for proving the threshold of the existence of a structure

- Denote X as indicator for the existence of this structure.
- Calculate $E(X)$, prove one side.
- Calculate $Var(X)$, prove the other side.