>>> Distribution Learning

Name: Alkis Kalavasis

Date: May 28, 2022

>>> Contents

>>> Problem Formulation: Distribution Learning

Density estimation or *distribution learning* is the following task: given data generated from an unknown target probability distribution $f^\star$ from a known class $\mathcal{F}$, design/compute $\widehat{f}$ that is close to $f^\star$.

>>> Problem Formulation: Distribution Learning

Density estimation or *distribution learning* is the following task: given data generated from an unknown target probability distribution $f^\star$ from a known class $\mathcal{F}$, design/compute $\widehat{f}$ that is close to $f^\star$.

Example: $\mathcal{F}$ = Gaussian in d dimensions, $f^\star = \mathcal{N}(0, \mathrm{I})$.

>>> Problem Formulation: Distribution Learning

Density estimation or *distribution learning* is the following task: given data generated from an unknown target probability distribution $f^\star$ from a known class $\mathcal{F}$, design/compute $\widehat{f}$ that is close to $f^\star$.

Example: $\mathcal{F}$ = Gaussian in d dimensions, $f^\star = \mathcal{N}(0, \mathrm{I})$.

* Evaluation: Sample Complexity and Computational Complexity
* Data generated i.i.d. from $f^\star$
* Our measure of closeness is the Total Variation distance

>>> TV Distance

$$\|f\|_1 = \sum_{x \in X} |f(x)| \quad \texttt{or} \quad \int_{x \in X} |f(x)| dx$$

>>> TV Distance

$$\|f\|_1 = \sum_{x \in X} |f(x)| \text{ or } \int_{x \in X} |f(x)| dx$$

Total Variation distance:

$$d_{TV}(P, Q) = \frac{1}{2}\|P - Q\|_1$$

Why $1/2$?

>>> TV Distance

$$\|f\|_1 = \sum_{x \in X} |f(x)| \text{ or } \int_{x \in X} |f(x)| dx$$

Total Variation distance:

$$d_{TV}(P, Q) = \frac{1}{2}\|P - Q\|_1$$

Why $1/2$?

$$d_{TV}(P, Q) = \max_{S \in \mathcal{A}} |P(S) - Q(S)|$$

If $\widehat{f}$ is a density estimate from m samples, we define the risk of the estimator with respect to the class $\mathcal{F}$ as

$$\mathcal{R}_m(\widehat{f}, \mathcal{F}) = \sup_{f \in \mathcal{F}} \mathbb{E}[d_{TV}(\widehat{f}, f)]$$

>>> Learning Distributions

If $\widehat{f}$ is a density estimate from m samples, we define the risk of the estimator with respect to the class $\mathcal{F}$ as

$$\mathcal{R}_m(\widehat{f}, \mathcal{F}) = \sup_{f \in \mathcal{F}} \mathbb{E}[d_{TV}(\widehat{f}, f)]$$

The analogue of the optimal sample complexity is the minimax risk of the class $\mathcal{F}$

$$\mathcal{R}_m(\mathcal{F}) = \inf_{\widehat{f}} \sup_{f \in \mathcal{F}} \mathbb{E}[d_{TV}(\widehat{f}, f)]$$

>>> Learning Discrete Distributions over $X = [n]$

$$\mathcal{F} = \left\{ p = (p_1, p_2, ..., p_n) : p_i > 0, \sum_{i \in [n]} p_i = 1 \right\} .$$

>>> Learning Discrete Distributions over $\boldsymbol{X} = [n]$

$$\mathcal{F} = \left\{ p = (p_1, p_2, ..., p_n) : p_i > 0, \sum_{i \in [n]} p_i = 1 \right\}.$$

Problem: Given access to i.i.d. samples from the unknown $p \in \mathcal{F}$, output a hypothesis q s.t. $d_{TV}(p, q) < \epsilon$ w.p. $1 - \delta$.

>>> Learning Discrete Distributions over $\boldsymbol{X} = [\boldsymbol{n}]$

$$\mathcal{F} = \left\{ p = (p_1, p_2, ..., p_n) : p_i > 0, \sum_{i \in [n]} p_i = 1 \right\}.$$

Problem: Given access to i.i.d. samples from the unknown $p \in \mathcal{F}$, output a hypothesis q s.t. $d_{TV}(p, q) < \epsilon$ w.p. $1 - \delta$.

Fact: $\Theta(\frac{n+\log(1/\delta)}{\epsilon^2})$ (or $R_m(\mathcal{F}) = \sqrt{n/m}$).

>>> Learning Discrete Distributions over $\boldsymbol{X} = [\boldsymbol{n}]$

$$\mathcal{F} = \left\{ p = (p_1, p_2, ..., p_n) : p_i > 0, \sum_{i \in [n]} p_i = 1 \right\}.$$

Problem: Given access to i.i.d. samples from the unknown $p \in \mathcal{F}$, output a hypothesis q s.t. $d_{TV}(p, q) < \epsilon$ w.p. $1 - \delta$.
Fact: $\Theta(\frac{n + \log(1/\delta)}{\epsilon^2})$ (or $R_m(\mathcal{F}) = \sqrt{n/m}$). The upper bound:

>>> Learning Discrete Distributions over $\boldsymbol{X} = [n]$

$$\mathcal{F} = \left\{ p = (p_1, p_2, ..., p_n) : p_i > 0, \sum_{i \in [n]} p_i = 1 \right\}.$$

Problem: Given access to i.i.d. samples from the unknown $p \in \mathcal{F}$, output a hypothesis q s.t. $d_{TV}(p, q) < \epsilon$ w.p. $1 - \delta$.

Fact: $\Theta(\frac{n+\log(1/\delta)}{\epsilon^2})$ (or $R_m(\mathcal{F}) = \sqrt{n/m}$). The upper bound:

* Compute the empirical distribution $\hat{p}$ given m samples $x_1, ..., x_m \sim p$.

>>> Learning Discrete Distributions over $\boldsymbol{X} = [n]$

$$\mathcal{F} = \left\{ p = (p_1, p_2, ..., p_n) : p_i > 0, \sum_{i \in [n]} p_i = 1 \right\}.$$

Problem: Given access to i.i.d. samples from the unknown $p \in \mathcal{F}$, output a hypothesis q s.t. $d_{TV}(p, q) < \epsilon$ w.p. $1 - \delta$.

Fact: $\Theta(\frac{n+\log(1/\delta)}{\epsilon^2})$ (or $R_m(\mathcal{F}) = \sqrt{n/m}$). The upper bound:

* Compute the empirical distribution $\hat{p}$ given m samples $x_1, ..., x_m \sim p$.
* $d_{TV}(\hat{p}, p) > \epsilon \iff \exists S \subset [n]$ s.t. $\hat{p}(S) - p(S) > \epsilon$.

>>> Learning Discrete Distributions over $X = [n]$

$$\mathcal{F} = \left\{ p = (p_1, p_2, ..., p_n) : p_i > 0, \sum_{i \in [n]} p_i = 1 \right\}.$$

Problem: Given access to i.i.d. samples from the unknown $p \in \mathcal{F}$, output a hypothesis q s.t. $d_{TV}(p, q) < \epsilon$ w.p. $1 - \delta$.

Fact: $\Theta(\frac{n + \log(1/\delta)}{\epsilon^2})$ (or $R_m(\mathcal{F}) = \sqrt{n/m}$). The upper bound:

* Compute the empirical distribution $\hat{p}$ given m samples $x_1, ..., x_m \sim p$.
* $d_{TV}(\hat{p}, p) > \epsilon \iff \exists S \subset [n]$ s.t. $\hat{p}(S) - p(S) > \epsilon$.
* Step 1: Fix $S \subset [n]$

$$\hat{p}(S) = \sum_{j \in S} \hat{p}(j) =$$

>>> Learning Discrete Distributions over $\boldsymbol{X} = [n]$

$$\mathcal{F} = \left\{ p = (p_1, p_2, ..., p_n) : p_i > 0, \sum_{i \in [n]} p_i = 1 \right\}.$$

Problem: Given access to i.i.d. samples from the unknown $p \in \mathcal{F}$, output a hypothesis q s.t. $d_{TV}(p, q) < \epsilon$ w.p. $1 - \delta$.

Fact: $\Theta(\frac{n + \log(1/\delta)}{\epsilon^2})$ (or $R_m(\mathcal{F}) = \sqrt{n/m}$). The upper bound:

* Compute the empirical distribution $\hat{p}$ given m samples $x_1, ..., x_m \sim p$.
* $d_{TV}(\hat{p}, p) > \epsilon \iff \exists S \subset [n]$ s.t. $\hat{p}(S) - p(S) > \epsilon$.
* Step 1: Fix $S \subset [n]$

$$\hat{p}(S) = \sum_{j \in S} \hat{p}(j) = \sum_{j \in S} \left(\frac{1}{m} \sum_{i=1}^{m} \mathbf{1}\{x_i = j\} \right) =$$

>>> Learning Discrete Distributions over $X = [n]$

$$\mathcal{F} = \left\{ p = (p_1, p_2, ..., p_n) : p_i > 0, \sum_{i \in [n]} p_i = 1 \right\}.$$

Problem: Given access to i.i.d. samples from the unknown $p \in \mathcal{F}$, output a hypothesis q s.t. $d_{TV}(p, q) < \epsilon$ w.p. $1 - \delta$.

Fact: $\Theta(\frac{n+\log(1/\delta)}{\epsilon^2})$ (or $R_m(\mathcal{F}) = \sqrt{n/m}$). The upper bound:

* Compute the empirical distribution $\hat{p}$ given m samples $x_1, ..., x_m \sim p$.
* $d_{TV}(\hat{p}, p) > \epsilon \iff \exists S \subset [n]$ s.t. $\hat{p}(S) - p(S) > \epsilon$.
* Step 1: Fix $S \subset [n]$

$$\hat{p}(S) = \sum_{j \in S} \hat{p}(j) = \sum_{j \in S} \left(\frac{1}{m} \sum_{i=1}^{m} \mathbf{1}\{x_i = j\} \right) = \frac{1}{m} \sum_{i=1}^{m} X_i$$

where $X_i \sim \text{Be}(p(S))$ (i.i.d.)

>>> Learning Discrete Distributions over $X = [n]$

$$\mathcal{F} = \left\{ p = (p_1, p_2, ..., p_n) : p_i > 0, \sum_{i \in [n]} p_i = 1 \right\}.$$

Problem: Given access to i.i.d. samples from the unknown $p \in \mathcal{F}$, output a hypothesis q s.t. $d_{TV}(p, q) < \epsilon$ w.p. $1 - \delta$.

Fact: $\Theta(\frac{n + \log(1/\delta)}{\epsilon^2})$ (or $R_m(\mathcal{F}) = \sqrt{n/m}$). The upper bound:

* Compute the empirical distribution $\hat{p}$ given m samples $x_1, ..., x_m \sim p$.
* $d_{TV}(\hat{p}, p) > \epsilon \iff \exists S \subset [n]$ s.t. $\hat{p}(S) - p(S) > \epsilon$.
* Step 1: Fix $S \subset [n]$

$$\hat{p}(S) = \sum_{j \in S} \hat{p}(j) = \sum_{j \in S} \left(\frac{1}{m} \sum_{i=1}^{m} \mathbf{1}\{x_i = j\} \right) = \frac{1}{m} \sum_{i=1}^{m} X_i$$

  where $X_i \sim \text{Be}(p(S))$ (i.i.d.)
* Step 2: Hoeffding: $\Pr[\hat{p}(S) - p(S) > \epsilon] \leq \exp(-2\epsilon^2 m)$

>>> Learning Discrete Distributions over $\boldsymbol{X} = [\boldsymbol{n}]$

$$\mathcal{F} = \left\{ p = (p_1, p_2, ..., p_n) : p_i > 0, \sum_{i \in [n]} p_i = 1 \right\}.$$

Problem: Given access to i.i.d. samples from the unknown $p \in \mathcal{F}$, output a hypothesis q s.t. $d_{TV}(p,q) < \epsilon$ w.p. $1-\delta$.

Fact: $\Theta(\frac{n+\log(1/\delta)}{\epsilon^2})$ (or $R_m(\mathcal{F}) = \sqrt{n/m}$). The upper bound:

* Compute the empirical distribution $\hat{p}$ given m samples $x_1, ..., x_m \sim p$.
* $d_{TV}(\hat{p}, p) > \epsilon \iff \exists S \subset [n]$ s.t. $\hat{p}(S) - p(S) > \epsilon$.
* Step 1: Fix $S \subset [n]$

$$\hat{p}(S) = \sum_{j \in S} \hat{p}(j) = \sum_{j \in S} \left(\frac{1}{m} \sum_{i=1}^{m} \mathbf{1}\{x_i = j\} \right) = \frac{1}{m} \sum_{i=1}^{m} X_i$$

  where $X_i \sim \mathrm{Be}(p(S))$ (i.i.d.)
* Step 2: Hoeffding: $\Pr[\hat{p}(S) - p(S) > \epsilon] \leq \exp(-2\epsilon^2 m)$
* Step 3: U.B.: $\Pr[\exists S \subset [n] : \hat{p}(S) - p(S) > \epsilon] \leq 2^n \exp(-2\epsilon^2 m) \leq \delta$.

>>> Continuous Case

>>> Continuous Case

For continuous distributions the learning problem is not solvable with no assumptions.

>>> Continuous Case

For continuous distributions the learning problem is not solvable with no assumptions.
Intuition : $n \to \infty$

>>> Continuous Case

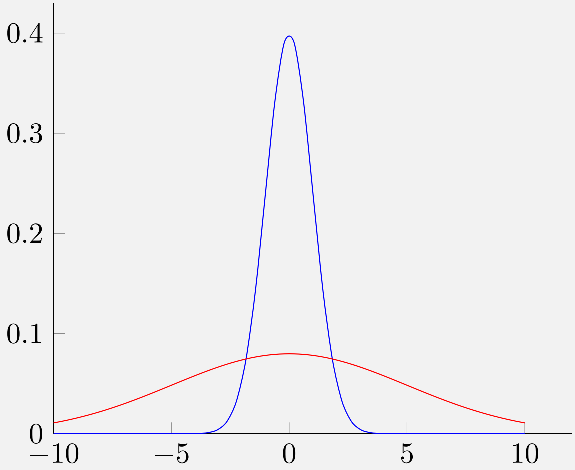
For continuous distributions the learning problem is not solvable with no assumptions.

Intuition : $n \to \infty$

Focus on structured distribution families, e.g., parametric families.

>>> Univariate Gaussian: MLE

$x \sim \mathcal{N}(\mu, \sigma^2)$



>>> Univariate Case

How many parameters? Can we accurately estimate them?

>>> Univariate Case

How many parameters? Can we accurately estimate them? N samples from $\mathcal{N}(\mu, \sigma^2)$

Empirical mean

$$\hat{\mu} = \frac{1}{N} \sum_{i=1}^{N} x_i \to \mu$$

Empirical variance

$$\frac{1}{N} \sum_{i=1}^{N} (x_i - \hat{\mu})^2 \to \sigma^2$$

>>> Maximum Log-Likelihood

$$x_1, ..., x_N \sim \mathcal{N}(\mu, \sigma^2)^{\otimes N}$$

>>> Maximum Log-Likelihood

$x_1, ..., x_N \sim \mathcal{N}(\mu, \sigma^2)^{\otimes N}$

$\mathcal{L}(x_1, ..., x_N | \mu, \sigma^2) = \prod_{i \in [N]} \mathcal{N}(x_i | \mu, \sigma^2) =$

>>> Maximum Log-Likelihood

$$x_1, ..., x_N \sim \mathcal{N}(\mu, \sigma^2)^{\otimes N}$$

$$\mathcal{L}(x_1, ..., x_N|\mu, \sigma^2) = \prod_{i \in [N]} \mathcal{N}(x_i|\mu, \sigma^2) = \prod_{i \in [N]} \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{(x_i-\mu)^2}{2\sigma^2})$$

\>>> Maximum Log-Likelihood

$$x_1, ..., x_N \sim \mathcal{N}(\mu, \sigma^2)^{\otimes N}$$

$$\mathcal{L}(x_1, ..., x_N | \mu, \sigma^2) = \prod_{i \in [N]} \mathcal{N}(x_i | \mu, \sigma^2) = \prod_{i \in [N]} \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{(x_i - \mu)^2}{2\sigma^2})$$

$$\ln(\mathcal{L}(x_1, ..., x_N | \mu, \sigma^2)) =$$

>>> Maximum Log-Likelihood

$x_1, ..., x_N \sim \mathcal{N}(\mu, \sigma^2)^{\otimes N}$

$\mathcal{L}(x_1, ..., x_N | \mu, \sigma^2) = \prod_{i \in [N]} \mathcal{N}(x_i | \mu, \sigma^2) = \prod_{i \in [N]} \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{(x_i - \mu)^2}{2\sigma^2})$

$$\ln(\mathcal{L}(x_1, ..., x_N | \mu, \sigma^2)) = -\frac{N}{2} \ln(2\pi) - \frac{N}{2} \ln(\sigma^2) - \frac{1}{2\sigma^2} \sum_{i=1}^{N} (x_i - \mu)^2$$

>>> Maximum Log-Likelihood

$x_1, ..., x_N \sim \mathcal{N}(\mu, \sigma^2)^{\otimes N}$

$\mathcal{L}(x_1, ..., x_N | \mu, \sigma^2) = \prod_{i \in [N]} \mathcal{N}(x_i | \mu, \sigma^2) = \prod_{i \in [N]} \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{(x_i - \mu)^2}{2\sigma^2})$

$$\ln(\mathcal{L}(x_1, ..., x_N | \mu, \sigma^2)) = -\frac{N}{2}\ln(2\pi) - \frac{N}{2}\ln(\sigma^2) - \frac{1}{2\sigma^2}\sum_{i=1}^{N}(x_i - \mu)^2$$

Optimize the negative log-likelihood over the space of parameters (μ, σ).

>>> KL divergence and MLE

θ^* true parameters, θ guess.

$$\mathrm{KL}(\mathcal{D}_{\theta^*}, \mathcal{D}_\theta) = \mathbb{E}_{x \sim \mathcal{D}_{\theta^*}}\left[\log\left(\frac{\mathcal{D}_{\theta^*}(x)}{\mathcal{D}_\theta(x)}\right)\right]$$

>>> KL divergence and MLE

θ^* true parameters, θ guess.

$$\mathrm{KL}(\mathcal{D}_{\theta^*}, \mathcal{D}_\theta) = \mathbb{E}_{x \sim \mathcal{D}_{\theta^*}}\left[\log\left(\frac{\mathcal{D}_{\theta^*}(x)}{\mathcal{D}_\theta(x)}\right)\right]$$

$$\mathrm{KL}(\mathcal{D}_{\theta^*}, \mathcal{D}_\theta) = \Theta(1) - \mathbb{E}_{\theta^*}[\log(\mathcal{D}_\theta)]$$

>>> KL divergence and MLE

θ^* true parameters, θ guess.

$$\mathrm{KL}(\mathcal{D}_{\theta^*}, \mathcal{D}_\theta) = \mathbb{E}_{x \sim \mathcal{D}_{\theta^*}}\left[\log\left(\frac{\mathcal{D}_{\theta^*}(x)}{\mathcal{D}_\theta(x)}\right)\right]$$

$\mathrm{KL}(\mathcal{D}_{\theta^*}, \mathcal{D}_\theta) = \Theta(1) - \mathbb{E}_{\theta^*}[\log(\mathcal{D}_\theta)]$
Estimate $\mathbb{E}_{x \sim \mathcal{D}_{\theta^*}}[h(x)]$ with $\frac{1}{N}\sum_{i \in [N]} h(x_i)$

>>> KL divergence and MLE

θ^* true parameters, θ guess.

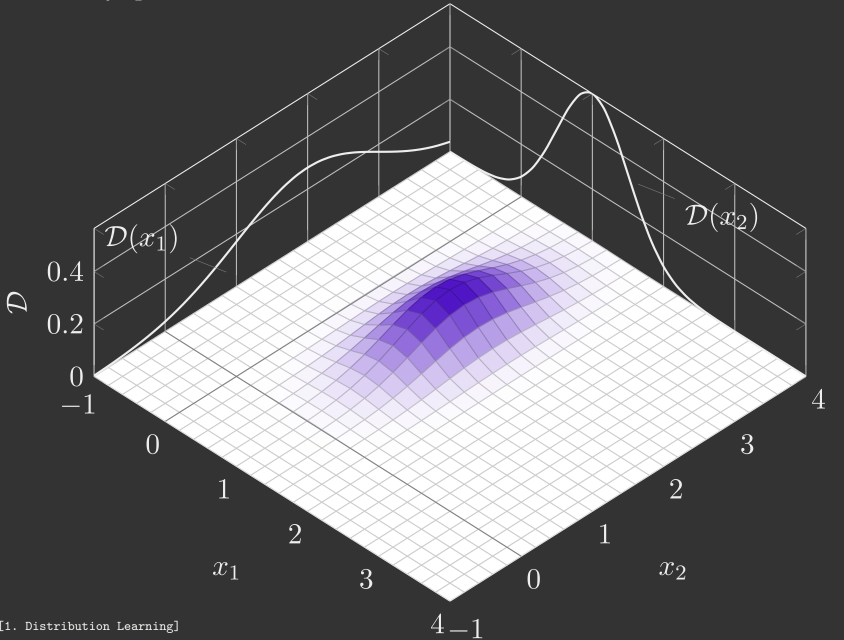
$$\mathrm{KL}(\mathcal{D}_{\theta^*}, \mathcal{D}_\theta) = \mathbb{E}_{x\sim\mathcal{D}_{\theta^*}}\left[\log\left(\frac{\mathcal{D}_{\theta^*}(x)}{\mathcal{D}_\theta(x)}\right)\right]$$

$\mathrm{KL}(\mathcal{D}_{\theta^*}, \mathcal{D}_\theta) = \Theta(1) - \mathbb{E}_{\theta^*}[\log(\mathcal{D}_\theta)]$
Estimate $\mathbb{E}_{x\sim\mathcal{D}_{\theta^*}}[h(x)]$ with $\frac{1}{N}\sum_{i\in[N]} h(x_i)$

$$\min_{\theta\in\Theta} \widehat{\mathrm{KL}}(\mathcal{D}_{\theta^*}, \mathcal{D}_\theta) = \min_{\theta\in\Theta} -\frac{1}{N}\sum_{i=1}^{N}\log(\mathcal{D}_\theta(x_i)) = \max_{\theta\in\Theta}\prod_{i=1}^{N}\mathcal{D}_\theta(x_i)$$

>>> Multivariate Case

How many parameters?



>>> Gaussian density estimation

* d-dimensional Gaussian $\mathcal{N}(\mu, \Sigma)$, $\mu_{d\times 1}, \Sigma_{d\times d}$:

$$\mathcal{N}(\mu, \Sigma)(x) = \frac{1}{\sqrt{(2\pi)^d \texttt{det}(\Sigma)}} \exp(-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)) .$$

* Ellipsoid: $\{x : (x-v)^\top A(x-v) = 1\}$ where $A \succeq 0$

> $\mathcal{N}_d$ using $O(d^2/\epsilon^2), \tilde{\Omega}(d^2/\epsilon^2)$ samples.

>>> Gaussian Upper Bound via Yatracos Class

For a class $\mathcal{F}$ of functions from $\mathbb{X}$ to $\mathbb{R}$, the Yatracos class of $\mathcal{F}$ is

$$\mathcal{Y}(\mathcal{F}) = \{\{x \in \mathbb{X} : f_1(x) \geq f_2(x)\} : f_1, f_2 \in \mathcal{F}\}\,.$$

>>> Gaussian Upper Bound via Yatracos Class

For a class $\mathcal{F}$ of functions from $\mathbb{X}$ to $\mathbb{R}$, the Yatracos class of $\mathcal{F}$ is

$$\mathcal{Y}(\mathcal{F}) = \{\{x \in \mathbb{X} : f_1(x) \geq f_2(x)\} : f_1, f_2 \in \mathcal{F}\}.$$

Exercise: $d_{TV}(f_1, f_2) = \|f_1 - f_2\|_{\mathcal{Y}(\mathcal{F})}$

>>> Gaussian Upper Bound via Yatracos Class

For a class $\mathcal{F}$ of functions from $\mathbb{X}$ to $\mathbb{R}$, the Yatracos class of $\mathcal{F}$ is

$$\mathcal{Y}(\mathcal{F}) = \{\{x \in \mathbb{X} : f_1(x) \geq f_2(x)\} : f_1, f_2 \in \mathcal{F}\}.$$

Exercise: $d_{TV}(f_1, f_2) = \|f_1 - f_2\|_{\mathcal{Y}(\mathcal{F})}$

(1) For any class $\mathcal{F}$, the sample complexity of learning $\mathcal{F}$ is $O(\frac{\mathrm{VCdim}(\mathcal{Y}(\mathcal{F}) + \log(1/\delta))}{\epsilon^2})$.
(2) Let G be a vector space of real-valued functions. Then $\mathrm{VCdim}(\{\{x : f(x) > 0\} : f \in G\}) \leq \dim(G)$.

Proof: $\mathcal{Y}(\mathcal{N}_d) = \{\{x : \mathcal{N}(\mu_1, \Sigma_1)(x) \geq \mathcal{N}(\mu_2, \Sigma_2)(x)\} : \mu_i, \Sigma_i\}$ and so is contained in the space $\{\{x^\top Ax + b^\top x + c > 0\} : A, b, c\}$ whose dimension is $O(d^2)$.

>>> Permutations

We assume that there is a hidden central ranking $\pi_0 \in \mathbb{S}_n$ and we define a notion of distance between permutations:

>>> Permutations

We assume that there is a hidden central ranking $\pi_0 \in \mathbb{S}_n$ and we define a notion of distance between permutations:

$$d_{KT}(\pi, \sigma) = \sum_{i \succ_\pi j} 1\{j \succ_\sigma i\} = \texttt{Bubblesort}(\pi, \sigma)$$

>>> Permutations

We assume that there is a hidden central ranking $\pi_0 \in \mathbb{S}_n$ and we define a notion of distance between permutations:

$$d_{KT}(\pi, \sigma) = \sum_{i \succ_\pi j} 1\{j \succ_\sigma i\} = \texttt{Bubblesort}(\pi, \sigma)$$

$d_{KT}(123, 213) = 1$
$d_{KT}(123, 312) = 2$
$d_{KT}(\pi, \pi^{-1}) = \binom{n}{2}$

>>> Permutations

We assume that there is a hidden central ranking $\pi_0 \in \mathbb{S}_n$ and we define a notion of distance between permutations:

$$d_{KT}(\pi, \sigma) = \sum_{i \succ_\pi j} 1\{j \succ_\sigma i\} = \texttt{Bubblesort}(\pi, \sigma)$$

$d_{KT}(123, 213) = 1$
$d_{KT}(123, 312) = 2$
$d_{KT}(\pi, \pi^{-1}) = \binom{n}{2}$
Mallows Model $\mathbf{M}(\pi, \beta)$

$$\Pr[\pi | \pi_0, \beta] \propto \exp(-\beta \cdot d_{KT}(\pi, \pi_0))$$

>>> Permutations

We assume that there is a hidden central ranking $\pi_0 \in \mathbb{S}_n$ and we define a notion of distance between permutations:

$$d_{KT}(\pi, \sigma) = \sum_{i \succ_\pi j} 1\{j \succ_\sigma i\} = \texttt{Bubblesort}(\pi, \sigma)$$

$d_{KT}(123, 213) = 1$
$d_{KT}(123, 312) = 2$
$d_{KT}(\pi, \pi^{-1}) = \binom{n}{2}$
Mallows Model $\mathbf{M}(\pi, \beta)$

$$\Pr[\pi | \pi_0, \beta] \propto \exp(-\beta \cdot d_{KT}(\pi, \pi_0))$$

Sampling from a Mallows model, can we learn the true target ranking π_0?

Learning with probability at least $1 - \epsilon$ using

Learning with probability at least $1 - \epsilon$ using $\Theta(\log(n/\epsilon))$ samples.

Learning with probability at least $1-\epsilon$ using $\Theta(\log(n/\epsilon))$ samples.
In each sample, either $i \succ j$ or $j \succ i$

Learning with probability at least $1 - \epsilon$ using $\Theta(\log(n/\epsilon))$ samples.
In each sample, either $i \succ j$ or $j \succ i$
Count for each ordered pair i, j, the votes n_{ij} and n_{ji}

Learning with probability at least $1-\epsilon$ using $\Theta(\log(n/\epsilon))$ samples.

In each sample, either $i \succ j$ or $j \succ i$

Count for each ordered pair i, j, the votes n_{ij} and n_{ji}

If $i \succ_{\pi_0} j$, we expect $n_{ij} - n_{ji} > 0$ due to the Mallows model

Learning with probability at least $1-\epsilon$ using $\Theta(\log(n/\epsilon))$ samples.

In each sample, either $i \succ j$ or $j \succ i$

Count for each ordered pair i, j, the votes n_{ij} and n_{ji}

If $i \succ_{\pi_0} j$, we expect $n_{ij} - n_{ji} > 0$ due to the Mallows model

Hoeffding and U.B. over $\binom{n}{2}$ pairs.

>>> Learning Coarse Gaussians

Consider a mixture of partitions π over $\mathbb{R}^d$ and an unknown target mean $\mu^\star$.

1. Draw a partition $\mathcal{S} \sim \pi$
2. Draw $x \sim \mathcal{N}(\mu^\star, I)$
3. Output the unique set $S \in \mathcal{S}$ that contains x (with distribution $\mathcal{N}_\pi$)

Can we learn the true mean from i.i.d. samples from $\mathcal{N}_\pi$?

>>> Efficient algorithm for Coarse Gaussians

Draw S from $\mathcal{N}_\pi(\mu^\star)$

>>> Efficient algorithm for Coarse Gaussians

Draw S from $\mathcal{N}_\pi(\mu^\star)$

$$\mathcal{L}(\mu) = \log(\mathcal{N}(\mu; S)) = \log\left(\int_S \frac{1}{\sqrt{(2\pi)^d}} \exp(-\|x-\mu\|_2^2/2)\right)$$

>>> Efficient algorithm for Coarse Gaussians

Draw S from $\mathcal{N}_\pi(\mu^\star)$

$$\mathcal{L}(\mu) = \log(\mathcal{N}(\mu; S)) = \log\left(\int_S \frac{1}{\sqrt{(2\pi)^d}} \exp(-\|x-\mu\|_2^2/2)\right)$$

$$\nabla\mathcal{L}(\mu) = \frac{\int_S (x-\mu)\cdot\exp(-\|x-\mu\|_2^2/2)dx}{\int_S \exp(-\|x-\mu\|_2^2/2)dx} = \mathbb{E}_{\mathcal{N}_S(\mu)}[x] - \mu$$

>>> Efficient algorithm for Coarse Gaussians

Draw S from $\mathcal{N}_\pi(\mu^\star)$

$$\mathcal{L}(\mu) = \log(\mathcal{N}(\mu; S)) = \log\left(\int_S \frac{1}{\sqrt{(2\pi)^d}} \exp(-\|x-\mu\|_2^2/2)\right)$$

$$\nabla\mathcal{L}(\mu) = \frac{\int_S (x-\mu)\cdot\exp(-\|x-\mu\|_2^2/2)dx}{\int_S \exp(-\|x-\mu\|_2^2/2)dx} = \mathbb{E}_{\mathcal{N}_S(\mu)}[x] - \mu$$

$$\nabla^2\mathcal{L}(\mu) = \mathrm{Cov}_{\mathcal{N}_S(\mu)}[x] - I$$

>>> Efficient algorithm for Coarse Gaussians

Draw S from $\mathcal{N}_\pi(\mu^\star)$

$$\mathcal{L}(\mu) = \log(\mathcal{N}(\mu; S)) = \log\left(\int_S \frac{1}{\sqrt{(2\pi)^d}} \exp(-\|x-\mu\|_2^2/2)\right)$$

$$\nabla\mathcal{L}(\mu) = \frac{\int_S (x-\mu)\cdot\exp(-\|x-\mu\|_2^2/2)dx}{\int_S \exp(-\|x-\mu\|_2^2/2)dx} = \mathbb{E}_{\mathcal{N}_S(\mu)}[x] - \mu$$

$$\nabla^2\mathcal{L}(\mu) = \mathrm{Cov}_{\mathcal{N}_S(\mu)}[x] - I$$

If S is convex then the Brascamp-Lieb Inequality implies that the negative log-likelihood is convex!

>>> Efficient algorithm for Coarse Gaussians

Draw S from $\mathcal{N}_\pi(\mu^\star)$

$$\mathcal{L}(\mu) = \log(\mathcal{N}(\mu; S)) = \log\left(\int_S \frac{1}{\sqrt{(2\pi)^d}} \exp(-\|x-\mu\|_2^2/2)\right)$$

$$\nabla \mathcal{L}(\mu) = \frac{\int_S (x-\mu)\cdot \exp(-\|x-\mu\|_2^2/2)dx}{\int_S \exp(-\|x-\mu\|_2^2/2)dx} = \mathbb{E}_{\mathcal{N}_S(\mu)}[x] - \mu$$

$$\nabla^2 \mathcal{L}(\mu) = \mathrm{Cov}_{\mathcal{N}_S(\mu)}[x] - I$$

If S is convex then the Brascamp-Lieb Inequality implies that the negative log-likelihood is convex!
Beyond convexity?

>>> Ising Model and RBMs

J symmetric matrix, h external field

$$\Pr[X = x] = \frac{1}{Z} \exp(\frac{1}{2} \sum_{i,j} J_{ij} x_i x_j + \sum_i h_i x_i)$$

Ising models with hidden variables Y

$$\Pr[X = x, Y = y] \frac{1}{Z} \exp(x^\top J y + \sum_{i \in [n]} h_i^1 x_i + \sum_{j \in [m]} h_j^2 y_j)$$

Ferromagnetic: $J_{ij} \geq 0, h_i^1, h_j^2 \geq 0$
How many samples from RBM to learn the structure of the bipartite graph?

>>> Influence in Ising models

The observed variables that exert the most influence on some variable X_i ought to be X_i' s two-hop neighbors.

>>> Influence in Ising models

The observed variables that exert the most influence on some variable X_i ought to be X_i' s two-hop neighbors.

$$I_i(S) = \mathbb{E}_{X \sim \mu(J,h)}[X_i | X_S = \{+1\}^{|S|}]$$

>>> Influence in Ising models

The observed variables that exert the most influence on some variable X_i ought to be X_i' s two-hop neighbors.

$$I_i(S) = \mathbb{E}_{X \sim \mu(J,h)}[X_i | X_S = \{+1\}^{|S|}]$$

If J, h are ferromagnetic, then $I_i(S)$ is a monotone submodular function for any i.

>>> Influence in Ising models

The observed variables that exert the most influence on some variable X_i ought to be X_i' s two-hop neighbors.

$$I_i(S) = \mathbb{E}_{X\sim\mu(J,h)}[X_i | X_S = \{+1\}^{|S|}]$$

If J, h are ferromagnetic, then $I_i(S)$ is a monotone submodular function for any i.

Submodular: For $S \subseteq T$

$$I_i(S \cup \{j\}) - I_i(S) \geq I_i(T \cup \{j\}) - I_i(T)$$

>>> The Algorithm

Greedy Neighborhood for i

>>> The Algorithm

Greedy Neighborhood for i

1. Set $S_0 = \emptyset$
2. For $t = 1, ..., d_2$:
   2.1 Let $j_{t+1} = \text{argmax} I_i(S_t \cup \{j\})$
   2.2 $S_{t+1} = S_t \cup \{j_{t+1}\}$
3. Find two-hop neighborhood $j \in S_k$

Number of samples: $\text{poly}(d_2) \cdot \log(n)$