

Samples

If we don't know we can estimate it from a given sample

Classification

Bayes' Rule

$$P(C_i | \vec{x}) = \frac{P(C_i) P(\vec{x} | C_i)}{P(\vec{x})}$$

$$\text{posterior} = \frac{\text{prior} \cdot \text{likelihood}}{\text{evidence}}$$

in general

if we have K classes (mutually exclusive, and exhaustive): $C_1, \dots, C_K$ we have

$$P(C_i) \geq 0 \text{ and } \sum_{i=1}^K P(C_i) = 1$$

$$P(C_i | \vec{x}) = \frac{P(\vec{x} | C_i) P(C_i)}{\sum_{k=1}^K P(\vec{x} | C_k) P(C_k)}$$

→ Bayes classifier: choose the class with the highest posterior probability

$$C_i \text{ if } P(C_i | \vec{x}) = \max_{k=1, \dots, K} P(C_k | \vec{x})$$

• action α is the decision to assign the input to class C_i

• $\lambda_{i,k}$ is the loss incurred for taking α when the input actually belongs to C_k .

• then the expected risk for taking α_i is

$$R(\alpha_i | \vec{x}) = \sum_{k=1}^K \lambda_{i,k} P(C_k | \vec{x})$$

in the case where

$$\lambda_{i,k} = \begin{cases} 0 & \text{if } i=k \\ 1 & \text{if } i \neq k \end{cases}$$

$$R(\alpha_i | \vec{x}) = \sum_{k \neq i} P(C_k | \vec{x}) = 1 - P(C_i | \vec{x})$$

i.e. minimizing the expected Risk is maximizing the expected probability.

reject by modeling the 0/1 loss, a possible loss function is

$$\lambda_{i,k} = \begin{cases} 0 & \text{if } i=k \\ 1 & \text{if } i \in [K] - \{k\} \\ 2 & \text{if } i = K+1 \end{cases}$$

for K classes $C_1, C_2, \dots, C_K$

the risk of reject is

$$R(\alpha_{K+1} | \vec{x}) = \sum_{k=1}^K \lambda_{K+1,k} P(C_k | \vec{x}) = 2$$

the risk of choosing C_i is $1 - P(C_i | \vec{x})$

the optimal decision rule?

• choose C_i if

$$1) R(\alpha_i | \vec{x}) < R(\alpha_k | \vec{x}) \text{ for all } k \neq i$$

$$2) R(\alpha_i | \vec{x}) < R(\alpha_{K+1} | \vec{x})$$

• reject if

$$R(\alpha_{K+1} | \vec{x}) < R(\alpha_i | \vec{x}) \text{ for all } i$$

Alternatively

• choose C_i if

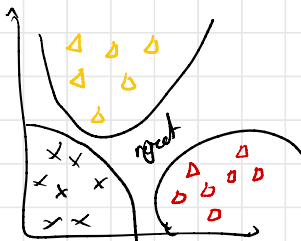
$$1) P(C_i | \vec{x}) > P(C_k | \vec{x}) \text{ for all } k \neq i$$

$$2) P(C_i | \vec{x}) > 1 - \lambda$$

• reject if

$$P(C_i | \vec{x}) < 1 - \lambda \text{ for all } i$$

Decision Region and Decision Boundary



Classification can be viewed as implementing a set of discriminant functions

$$g_i(\vec{x}), i = 1, \dots, K \text{ s.t.}$$

$$\text{choose } C_i \text{ if } g_i(\vec{x}) = \max_{k=1, \dots, K} g_k(\vec{x})$$

- α_i is the action chosen to assign to class C_i is indecisive reject action

- we can choose $g_i(\vec{x}) = -P(\alpha_i | \vec{x}) = P(C_i | \vec{x})$
- we can also put $g_i(\vec{x}) = p(\vec{x} | C_i) P(C_i)$

↳ since we assume $p(\vec{x})$ will be the same for all.

Maximum likelihood Estimator

Parametric Approach

- a sample is drawn from some distrib that obeys a known model.
- this model is defined up to a small number of parameters
- e.g. $N(\mu, \sigma^2)$ is a parametric approach that depends on two parameters: μ and σ

Statistic

- any value that is calculated from a given sample
- a statistic is said to be **sufficient** (for the parametric model) if:
 - no further information can be inferred from other statistics calculated from the same sample.
- e.g. $x_1, x_2, \dots, x_n \sim \text{iid } N(\mu, \sigma^2)$
 - the sample mean $m = \frac{1}{n} \sum_{i=1}^n x_i$
 - the sample variance $s^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - m)^2$
 - then (m, s^2) is a sufficient statistic for (μ, σ^2)

using sufficient statistics, we can get statistical models with only few parameters

Parametric Approach

density estimation

- $\mathcal{X} = \{\vec{x}_n\}_{n=1}^N$, where $\vec{x}_n \sim \text{iid } p(\vec{x} | \theta)$
- want $\hat{\theta}$ s.t. $\vec{x}_n$ is sampled from $p(\vec{x} | \hat{\theta})$ as likely as possible

likelihood

- maximize the likelihood of θ given $\mathcal{X}$:

$$L(\theta | \mathcal{X}) := P(\mathcal{X} | \theta) = \prod_{n=1}^N p(\vec{x}_n | \theta)$$

- e.g. $\mathcal{X}$ maximize the log-likelihood of θ given

$$L(\theta | \mathcal{X}) := \log L(\theta | \mathcal{X}) = \sum_{n=1}^N \log p(\vec{x}_n | \theta)$$

Multinomial density

K states $\{1, 2, \dots, K\}$

$\vec{x}$ takes state i with a prob q_i , $\sum_{i=1}^K q_i = 1$

define $\vec{x}_i = \begin{cases} 1 & \text{if state } i \text{ is chosen} \\ 0 & \text{else} \end{cases}$

we can represent $\vec{x}$ as a vector $(x_1, x_2, \dots, x_K)^T$ "one-hot" vector

$\Rightarrow \vec{x}_i$ is a vector where $x_i = 1$ and $x_k = 0$ for all $k \neq i$

$$\Rightarrow \vec{x} = (x_1, \dots, x_K)^T$$

$$P[\vec{x} = \vec{x}] = q_1^{x_1} q_2^{x_2} \dots q_K^{x_K}$$

we only care $x_i = 1$

$$P[\vec{x} = \vec{x}] = q_1^{x_1} \dots q_K^{x_K} = q_1^{x_1} q_2^{x_2} \dots q_K^{x_K}$$

parameters: $q_1, q_2, \dots, q_K$

$$i.e. P[\vec{x} | q_1, q_2, \dots, q_K]$$

given a sample $\vec{x} = \{\vec{x}_n\}_{n=1}^N$ each $\vec{x}_n$ is a K -dimensional one-hot vector

likelihood is

$$L(q_1, \dots, q_K | \mathcal{X}) = P(\vec{x} | q_1, \dots, q_K) = \prod_{n=1}^N p(\vec{x}_n | q_1, \dots, q_K) = \prod_{n=1}^N q_1^{x_{n1}} q_2^{x_{n2}} \dots q_K^{x_{nK}} = \prod_{n=1}^N \prod_{k=1}^K q_k^{x_{nk}}$$

the log-likelihood

$$L(q_1, \dots, q_K | \mathcal{X}) = \log \left(\prod_{n=1}^N \prod_{k=1}^K q_k^{x_{nk}} \right) = \sum_{n=1}^N \sum_{k=1}^K x_{nk} \log q_k$$

Following **Maximum likelihood**, we need to

$$\begin{cases} \max_{q_1, \dots, q_K} \sum_{n=1}^N \sum_{k=1}^K x_{nk} \log q_k \\ \text{s.t.} \quad \sum_{k=1}^K q_k = 1 \end{cases}$$

$$L(q, \lambda, \nu) = \sum_{n=1}^N \sum_{k=1}^K x_{nk} \log q_k - \lambda \left(\sum_{k=1}^K q_k - 1 \right)$$

$$\frac{\partial L}{\partial q_k} = \sum_{n=1}^N \left(\frac{x_{nk}}{q_k} \right) - \lambda = 0$$

$$\frac{\partial L}{\partial \lambda} = \sum_{k=1}^K q_k - 1 = 0$$

$$\sum_{k=1}^K q_k = 1$$

$$q_k = \frac{\sum_{n=1}^N x_{nk}}{\lambda}$$

we q_k does not depend on λ then fine

$$\sum_{n=1}^N \sum_{k=1}^K x_{nk} = \lambda \Rightarrow \sum_{k=1}^K \sum_{n=1}^N x_{nk} = \lambda = 1$$

$$\Rightarrow \frac{\sum_{i=1}^K \tilde{\sum_{n=1}^N x_{ni}}}{\lambda} = 1$$

$$\lambda = \sum_{i=1}^K \tilde{\sum_{n=1}^N x_{ni}} = \sum_{n=1}^N (1) = N$$

$$\Rightarrow q_i = \frac{\sum_{n=1}^N x_{ni}}{\lambda} = \frac{1}{N} \sum_{n=1}^N x_{ni}$$

$$\uparrow$$

$$(\hat{q}_{MHF})_i$$

2.1 Bias and variance = $\sigma^2 + \mu - E[m^2]$

• $\mathcal{X}$: a sample from a population specified up to a single parameter θ .

• $d = d(\mathcal{X})$: an estimator of θ

• to evaluate the quality of d , we measure $(d(\mathcal{X}) - \theta)^2$

• the Mean Squared Error (MSE) of d for the parameter θ is

$$r(d, \theta) = E[(d(\mathcal{X}) - \theta)^2]$$

what is the source of randomness? $\mathcal{X}$ is a set of N i.i.d. samples

bias

the bias of d is defined to be

$$b_d(\theta) = E(d(\mathcal{X})) - \theta$$

• if $b_d(\theta) = 0$ for any θ , the estimator is said to be unbiased
• otherwise it is biased.

Example: bias in MLE of Gaussian density

$$\hat{\mu}_{MLE} = m = \frac{\sum_{n=1}^N x_n}{N}$$

$$\hat{\sigma}^2_{MLE} = \frac{N-1}{N} S^2 = \frac{\sum_{n=1}^N (x_n - m)^2}{N}$$

$$E[\hat{\mu}_{MLE}] = E\left[\frac{\sum_{n=1}^N x_n}{N}\right] = \frac{1}{N} E\left[\sum_{n=1}^N x_n\right] = \frac{1}{N} \sum_{n=1}^N E[x_n] = \frac{1}{N} \sum_{n=1}^N \mu = \frac{N}{N} \mu = \mu$$

$\Rightarrow b_\mu(\mu_{MLE}) = E[\hat{\mu}_{MLE}] - \mu = \mu - \mu = 0$
we say that $\hat{\mu}_{MLE}$ is an unbiased estimator of μ .

On the other hand

$$\begin{aligned} E[\hat{\sigma}^2_{MLE}] &= E\left[\frac{\sum_{n=1}^N (x_n - m)^2}{N}\right] = \frac{1}{N} E[\dots] \\ &= \frac{1}{N} \sum_{n=1}^N E[(x_n - m)^2] = \frac{1}{N} \sum_{n=1}^N E[x_n^2 - 2x_n m + m^2] \\ &= \frac{1}{N} \sum_{n=1}^N E[x_n^2] - \frac{1}{N} 2 \sum_{n=1}^N E[x_n m] + \frac{1}{N} \sum_{n=1}^N E[m^2] \\ &= \frac{1}{N} \sum_{n=1}^N E[x_n^2] - \frac{1}{N} 2 E\left[\sum_{n=1}^N x_n m\right] + \frac{1}{N} \sum_{n=1}^N E[m^2] \\ &= \frac{1}{N} \sum_{n=1}^N E[x_n^2] - \frac{1}{N} 2 E[N m^2] + \frac{1}{N} \sum_{n=1}^N E[m^2] \\ &= \frac{1}{N} \sum_{n=1}^N E[x_n^2] - 2 E[m^2] + \frac{1}{N} N E[m^2] = \sigma^2 + \mu - E[m^2] \end{aligned}$$

$$\begin{aligned} E[m^2] &= E\left[\frac{\sum_{n=1}^N x_n \sum_{l=1}^N x_l}{N^2}\right]^{1/2} \\ &= E\left[\frac{\sum_{n=1}^N x_n (x_n + \sum_{l \neq n}^N x_l)}{N^2}\right]^{1/2} \\ &= E\left[\frac{\sum_{n=1}^N x_n^2 + \sum_{n=1}^N x_n \sum_{l \neq n}^N x_l}{N^2}\right]^{1/2} \\ &= E\left[\frac{\sum_{n=1}^N (x_n^2 + \sum_{l \neq n}^N x_n x_l)}{N^2}\right]^{1/2} \\ &= \left(\frac{1}{N^2} E\left[\sum_{n=1}^N (x_n^2 + \sum_{l \neq n}^N x_n x_l)\right]\right)^{1/2} \\ &= \frac{1}{N^2} \left[\sum_{n=1}^N (E[x_n^2] + E[x_n] E[x_l]) \right] \\ &= \frac{1}{N^2} N (E[x_n^2] + E[x_n]^2) + \frac{(N-1)N\mu^2}{N^2} \\ &= \frac{1}{N^2} N (\sigma^2 + \mu^2) + \frac{(N-1)\mu^2}{N} = \frac{\sigma^2 + N\mu^2}{N} \\ &= \sigma^2 + \mu - \frac{\sigma^2 + N\mu^2}{N} \end{aligned}$$

$$\begin{aligned} &= \sigma^2 + \mu - \frac{\sigma^2 + N\mu^2}{N} = \frac{N\sigma^2 + N\mu - \sigma^2 - N\mu^2}{N} \\ &= \frac{\sigma^2(N-1)}{N} = \frac{N-1}{N} \sigma^2 \end{aligned}$$

Now, $b_\sigma(\hat{\sigma}^2_{MLE}) = E[\hat{\sigma}^2_{MLE}] - \sigma^2 = \frac{N-1}{N} \sigma^2 - \sigma^2 = -\frac{1}{N} \sigma^2 \neq 0$

here $\hat{\sigma}^2_{MLE}$ is a biased estimator of σ

Nevertheless, $E\left[\frac{N}{N-1} \hat{\sigma}^2_{MLE}\right] = \frac{N}{N-1} \cdot \frac{N-1}{N} \sigma^2 = \sigma^2$

$\frac{N}{N-1} \cdot \frac{N-1}{N} \sigma^2 = \sigma^2$ is the sample variance

bias-variance formula

$$\begin{aligned} r(d, \theta) &= E[(d - \theta)^2] \quad \text{* note } d \text{ is a function of } \mathcal{X} \\ &= E[(d - E[d] + E[d] - \theta)^2] \\ &= E[(d - E[d])^2 + (E[d] - \theta)^2 + 2(d - E[d])(E[d] - \theta)] \\ &= \underbrace{E[(d - E[d])^2]}_{\sigma_d^2} + E[(E[d] - \theta)^2] + 2E[(d - E[d])(E[d] - \theta)] \\ &= \sigma_d^2 + (E[d] - \theta)^2 + 2(E[d] - \theta)E[d - E[d]] \\ &= \sigma_d^2 + (E[d] - \theta)^2 + 2(E[d] - \theta)(E[d] - E[d]) \\ &= \sigma_d^2 + (E[d] - \theta)^2 \end{aligned}$$

$$r(d, \theta) = \text{Var}(d) + (E[d] - \theta)^2$$

$$r(d, \theta) = \text{Var}(d) + (\text{bias}(d))^2$$

Score Function

given a log-likelihood function

$$\mathcal{L}(\theta | \mathcal{X}) = \sum_{n=1}^N \log(p(\tilde{x}_n | \theta))$$

we define the Fisher score function to be

$$S(\theta | \mathcal{X}) := \frac{\partial \mathcal{L}(\theta | \mathcal{X})}{\partial \theta}$$

- for MLE, $S(\hat{\theta}_{MLE} | \mathcal{X}) = 0$
- fact $\mathbb{E}[S(\theta | \mathcal{X})] = 0$
- the Fisher information is defined to be the variance of the score function:

$$\gamma(\theta) := \text{Var}(S(\theta | \mathcal{X})) \\ = -\mathbb{E}\left[\frac{\partial^2 \mathcal{L}(\theta | \mathcal{X})}{\partial \theta^2}\right]$$

$$\text{Var}(S(\theta | \mathcal{X})) = \mathbb{E}[S(\theta | \mathcal{X})^2] - \mathbb{E}[S(\theta | \mathcal{X})]^2 \\ = \mathbb{E}\left[\left(\frac{\partial \mathcal{L}(\theta | \mathcal{X})}{\partial \theta}\right)^2\right] - \mathbb{E}\left[\frac{\partial \mathcal{L}(\theta | \mathcal{X})}{\partial \theta}\right]^2$$

• remark: it does not make sense to talk about the score function and the Fisher information if a single observation $\tilde{x}_n$ in the sample.
just set $\mathcal{X} = \tilde{x}_n$

$$\Rightarrow S(\theta | \mathcal{X} = \tilde{x}_n) = \frac{\partial \log p(\tilde{x}_n | \theta)}{\partial \theta}$$

$$\text{and} \\ \gamma(\theta | \mathcal{X} = \tilde{x}_n) = \text{Var}(S(\theta | \tilde{x}_n)) = -\mathbb{E}\left[\frac{\partial^2 \log p(\tilde{x}_n | \theta)}{\partial \theta^2}\right]$$

The Bayesian estimator

$$\hat{p}_{MLE}(\tilde{x}) = p(\tilde{x} | \hat{\theta}_{MLE})$$

where $\hat{\theta}_{MLE}$ is solved by maximizing the (log-)likelihood.

The Bayesian view

- before looking at a sample, we may have some prior knowledge on the parameter θ
- that is, we have a prior density $p(\theta)$
- if the parameter of our model, θ , is regarded as a random variable, then we can determine
 - the prior $p(\theta)$
 - the likelihood $p(\mathcal{X} | \theta)$
 - the joint probability $p(\mathcal{X}, \theta) = p(\mathcal{X} | \theta)p(\theta)$

Bayes' Density Estimation

combining $p(\theta)$ with the likelihood density $p(\mathcal{X} | \theta)$, we have, by Bayes rule

$$p(\theta | \mathcal{X}) = \frac{p(\mathcal{X} | \theta)p(\theta)}{p(\mathcal{X})} = \frac{p(\mathcal{X} | \theta)p(\theta)}{\int p(\mathcal{X} | \theta')p(\theta')d\theta'}$$

$$\begin{aligned} \hat{p}_{Bayes}(\tilde{x}) &= p(\tilde{x} | \mathcal{X}) = \int p(\tilde{x}, \theta | \mathcal{X}) d\theta \\ &= \int p(\tilde{x} | \theta, \mathcal{X}) p(\theta | \mathcal{X}) d\theta \\ &= \int p(\tilde{x} | \theta) p(\theta | \mathcal{X}) d\theta \end{aligned}$$

if only I had a way to integrate...

$$\hat{p}_{Bayes}(\tilde{x}) = \int p(\tilde{x} | \theta) \frac{p(\mathcal{X} | \theta)p(\theta)}{\int p(\mathcal{X} | \theta')p(\theta')d\theta'} d\theta$$

- numerical integration methods do not work if we are in high dimension
- we will need to use sampling methods, which we will discuss later in the course.

Bayes density estimation

subjective

objective

$$\hat{p}_{Bayes}(\tilde{x}) = \int p(\tilde{x} | \theta) p(\theta | \mathcal{X}) d\theta \quad \left\{ \begin{array}{l} \text{subjective} \\ \text{objective} \end{array} \right. \quad \begin{array}{l} \hat{p}_{MLE}(\tilde{x}) \\ = p(\tilde{x} | \hat{\theta}_{MLE}) \end{array}$$

$$\text{we have } \hat{p}_{Bayes}(\tilde{x}) = \int p(\tilde{x} | \theta) p(\theta | \mathcal{X}) d\theta$$

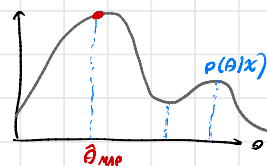
we suppose $p(\theta | \mathcal{X})$ is concentrated around a single point $\hat{\theta}$, then

$$\begin{aligned} \int p(\tilde{x} | \theta) p(\theta | \mathcal{X}) d\theta &= \int p(\tilde{x} | \theta) \delta(\theta - \hat{\theta}) d\theta \\ &= p(\tilde{x} | \hat{\theta}) \end{aligned}$$

• we recover the MLE density estimator if, in the above, $\hat{\theta} = \hat{\theta}_{MLE}$

• another choice: $\hat{\theta} = \hat{\theta}_{MAP}$ be defined by

$$\hat{\theta}_{MAP} = \arg\max_{\theta} p(\theta | \mathcal{X})$$



Gaussian MAP

Consider $N(X|A, \sigma^2)$ where σ^2 is known and fixed. Also

$$p(\theta) = N(\theta|0, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{\theta^2}{2\sigma^2}}$$

given a sample $X = \{X_n\}_{n=1}^N \sim^{i.i.d} N(X|A, \sigma^2)$

$$\begin{aligned} p(X|\theta) &= p(X_1, X_2, \dots, X_N|\theta) \\ &= p(X_1|\theta) p(X_2|\theta) \dots p(X_N|\theta) \\ &= \prod_{n=1}^N N(X_n|\theta, \sigma^2) \\ &= \prod_{n=1}^N \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(X_n-\theta)^2}{2\sigma^2}} \\ &= \frac{1}{\sigma^N (2\pi)^{N/2}} e^{-\sum_{n=1}^N \frac{(X_n-\theta)^2}{2\sigma^2}} \\ &= \frac{1}{\sigma^N (2\pi)^{N/2}} e^{-\frac{1}{2\sigma^2} \sum_{n=1}^N (X_n-\theta)^2} \end{aligned}$$

$$\begin{aligned} \hat{\theta}_{MAP} &= \arg \max_{\theta} p(\theta|X) = \arg \max_{\theta} \log(p(\theta|X)) \\ &= \arg \max_{\theta} \log(p(X|\theta) p(\theta)) \\ &= \arg \max_{\theta} \log(p(X|\theta)) + \log p(\theta) \end{aligned}$$

$$\log \frac{1}{\sigma^N (2\pi)^{N/2}} e^{-\frac{1}{2\sigma^2} \sum_{n=1}^N (X_n-\theta)^2} = \log \frac{1}{\sigma^N (2\pi)^{N/2}} - \frac{1}{2\sigma^2} \sum_{n=1}^N (X_n-\theta)^2$$

$$\log \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{\theta^2}{2\sigma^2}} = \log \frac{1}{\sigma \sqrt{2\pi}} - \frac{\theta^2}{2\sigma^2}$$

$$= \arg \max_{\theta} \left(-\log \sigma^N (2\pi)^{N/2} - \frac{1}{2\sigma^2} \sum_{n=1}^N (X_n-\theta)^2 - \log \sigma \sqrt{2\pi} - \frac{\theta^2}{2\sigma^2} \right)$$

$$= \arg \max_{\theta} \left(-N \log \sigma \sqrt{2\pi} - \frac{1}{2\sigma^2} \sum_{n=1}^N (X_n-\theta)^2 - \log \sigma \sqrt{2\pi} - \frac{\theta^2}{2\sigma^2} \right)$$

$$\frac{d}{d\theta} \left(-N \log \sigma \sqrt{2\pi} - \frac{1}{2\sigma^2} \sum_{n=1}^N (X_n-\theta)^2 - \log \sigma \sqrt{2\pi} - \frac{\theta^2}{2\sigma^2} \right)$$

$$= 0 + \frac{1}{\sigma^2} \sum_{n=1}^N (X_n-\theta) - 0 - \frac{2\theta}{2\sigma^2} = 0$$

$$\frac{1}{\sigma^2} \sum_{n=1}^N (X_n-\theta) = \frac{\theta}{\sigma^2}$$

$$\sigma^2 \sum_{n=1}^N (X_n-\theta) = \theta \sigma^2$$

$$\sigma^2 \sum_{n=1}^N X_n - \sigma^2 \sum_{n=1}^N \theta = \theta \sigma^2$$

$$\sigma^2 \sum_{n=1}^N X_n = \theta \sigma^2 + \sigma^2 N \theta$$

$$\theta (\sigma^2 + N \sigma^2) = \left(\sum_{n=1}^N X_n \right) \sigma^2$$

$$\theta = \frac{\sigma^2}{\sigma^2 + N \sigma^2} \sum_{n=1}^N X_n = \frac{1}{N + \frac{1}{\sigma^2}} \sum_{n=1}^N X_n$$

$$\hat{\theta}_{MAP} = \theta^* = \frac{\sum_{n=1}^N X_n}{N + \frac{1}{\sigma^2}}$$

yet another choice for the MAP:

$$\hat{\theta}_{Bayes} = E[\theta|X] = \int A p(\theta|X) d\theta$$

how do we choose the prior $p(\theta)$?

if $p(\theta)$ is chosen so that

$p(\theta|X)$ is of the same parametric form as $p(\theta)$, then $p(\theta)$ is said

to be a **conjugate prior** for the likelihood $p(X|\theta)$

for a gaussian likelihood, we can choose a Gaussian conjugate prior

Conjugate prior for multinomial likelihood

Recall: X takes state i with probability $q_i, i=1, \dots, K$. Each example $X_n \in \mathbb{R}^K$ is a one-hot vector

$$\text{Pembels: } \vec{q} = (q_1, \dots, q_K)^T$$

$$\text{Likelihood: } p(X|\vec{q}) = \prod_{n=1}^N \prod_{i=1}^K q_i^{X_{ni}}$$

Now define the prior to be a Dirichlet distribution:

$$\begin{aligned} p(\vec{q}) &= \text{Dir}(\vec{q}|\vec{\alpha}) \\ &= \frac{1}{\prod_{i=1}^K \Gamma(\alpha_i)} \prod_{i=1}^K q_i^{\alpha_i-1} \end{aligned}$$

normalization factor

$$\text{where } \vec{\alpha} = (\alpha_1, \dots, \alpha_K)^T$$

note: $\vec{\alpha}$

$$I(X) = \int_0^1 \dots \int_0^1 \vec{u}^X e^{-X} d\vec{u} \quad I(n) = (n-1)! \quad n \in \mathbb{N}$$

hence, the posterior $p(\vec{q}|X)$

$$p(\vec{q}|X) \propto p(X|\vec{q}) p(\vec{q})$$

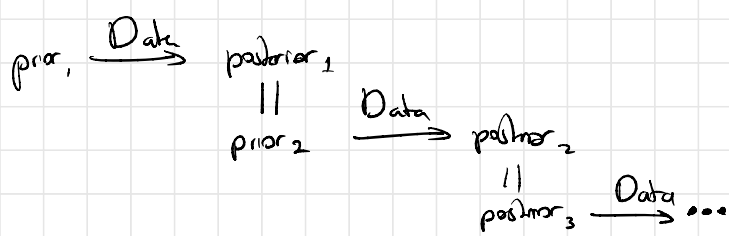
$$\begin{aligned} \prod_{n=1}^N \prod_{i=1}^K q_i^{X_{ni}} &= \prod_{i=1}^K q_i^{X_{i1} + X_{i2} + \dots + X_{iN}} \\ &= \prod_{i=1}^K q_i^{\sum_{n=1}^N X_{ni}} \end{aligned}$$

$$\prod_{i=1}^K q_i^{\sum_{n=1}^N X_{ni}} = \prod_{i=1}^K q_i^{N_i} \quad \text{det } N_i = \sum_{n=1}^N X_{ni} \text{ be the number of datapoints with state } i$$

$$\text{we write } p(\vec{q}|X) \propto \prod_{i=1}^K q_i^{(N_i + \alpha_i) - 1}$$

$$\text{so be explicit: } p(\vec{q}|X) = \frac{1}{\prod_{i=1}^K \Gamma(N_i + \alpha_i)} \prod_{i=1}^K q_i^{(N_i + \alpha_i) - 1}$$

we conclude that $\text{Dir}(\vec{q}|\vec{\alpha})$ is a conjugate prior for $p(X|\vec{q})$



Parametric Classification and Regression

K classes $(C_1, \dots, C_K)$

recall: Bayes decision rule

$$K = \arg \max P(C_i | x)$$

E.g., we can define a discriminant function

$$\bar{g}_i(x) = p(x | C_i) p(C_i) \leftarrow \text{discriminant}$$

$$K = \arg \max_i \bar{g}_i(x)$$

Equivalently we can choose to look at

$$g_i(x) = \log p(x | C_i) + \log p(C_i) \quad (1)$$

Consider a parametric model for $p(x | C_i)$

$$\text{Assume } p(x | C_i) = \mathcal{N}(x | \mu_i, \sigma_i^2) \quad i=1, \dots, K$$

$$= \frac{1}{\sqrt{2\pi} \sigma_i} e^{-\frac{1}{2} \left(\frac{x-\mu_i}{\sigma_i}\right)^2}$$

Then (1) becomes

$$g_i(x) = -\frac{1}{2} \log(2\pi) - \log \sigma_i - \frac{(x-\mu_i)^2}{2\sigma_i^2} + \log p(C_i) \quad (2)$$

Assume that we have a sample $X = \{\bar{x}_n\}_{n=1}^N$

where each $\bar{x}_n = (x_n, \bar{t}_n)$

where $x_n \in \mathbb{R}$, $\bar{t}_n \in \{0, 1\}^K$ is one-hot vector
s.t. $t_n = \begin{cases} 1 & \text{if } \bar{x}_n \in C_i \\ 0 & \text{if } \bar{x}_n \notin C_i \end{cases}$

For each class i

$$\hat{\mu}_i = \frac{\sum_{n=1}^N x_n t_{ni}}{\sum_{n=1}^N t_{ni}}$$

$$\hat{\sigma}_i^2 = \frac{\sum_{n=1}^N (x_n - \hat{\mu}_i)^2 t_{ni}}{\sum_{n=1}^N t_{ni}}$$

$$\hat{p}(C_i) = \frac{\sum_{n=1}^N t_{ni}}{N}$$

plug into (2)

$$g_i(x) = -\frac{1}{2} \log(2\pi) - \log \hat{\sigma}_i - \frac{(x - \hat{\mu}_i)^2}{2\hat{\sigma}_i^2} + \log \hat{p}(C_i)$$

if we further assume

1. the priors are all equal $\hat{p}(C_1) = \hat{p}(C_2) = \dots = \hat{p}(C_K)$

2. the variances are all equal $\hat{\sigma}_1^2 = \hat{\sigma}_2^2 = \dots = \hat{\sigma}_K^2$

$$K = \arg \max_i g_i(x)$$

$$= \arg \max_i - (x - \hat{\mu}_i)^2$$

$$= \arg \min_i (x - \hat{\mu}_i)^2$$

Regression

$$y = f(x) + \epsilon$$

f is unknown, estimated by $g(x | \theta)$

Assume $p(\epsilon) = \mathcal{N}(\epsilon | 0, \sigma^2)$

then $p(y | x) = \mathcal{N}(y | g(x, \theta), \sigma^2)$

Assume we have a sample $X = \{\bar{x}_n\}_{n=1}^N$

where each $\bar{x}_n = (x_n, y_n)^T \stackrel{i.i.d.}{\sim} p(x, y)$

note that $p(x, y) = p(y | x) p(x)$

Therefore, the log-likelihood

$$\mathcal{L}(\theta | X) = \log \prod_{n=1}^N p(x_n, y_n)$$

$$= \underbrace{\log \prod_{n=1}^N p(y_n | x_n)}_{\text{only this depends on } \theta} + \log \prod_{n=1}^N p(x_n)$$

$$\theta_{\text{MLE}} = \arg \max_{\theta} \log \prod_{n=1}^N p(y_n | x_n)$$

$$= \arg \max_{\theta} \sum_{n=1}^N \log \left(\frac{1}{\sqrt{2\pi} \sigma} e^{-\frac{1}{2} \left(\frac{y_n - g(x_n, \theta)}{\sigma}\right)^2} \right)$$

$$= \arg \max_{\theta} -\frac{N}{2} \log(2\pi) - N \log(\sigma) - \frac{1}{2\sigma^2} \sum_{n=1}^N (y_n - g(x_n, \theta))^2$$

$$= \arg \min_{\theta} \frac{1}{2} \sum_{n=1}^N (y_n - g(x_n, \theta))^2$$

$E(\theta | X)$ error function
a.k.a. "least squares estimation"

e.g. linear regression

$$g(x | w_0, w_1) = w_0 + w_1 x$$

the corresponding error function is

$$E(w_0, w_1 | X) = \frac{1}{2} \sum_{n=1}^N (y_n - w_0 - w_1 x_n)^2$$

Now get w_0 and w_1

$$\begin{cases} \frac{\partial E}{\partial w_0} = -\sum_{n=1}^N (y_n - w_0 - w_1 x_n) = 0 \\ \frac{\partial E}{\partial w_1} = -\sum_{n=1}^N (y_n - w_0 - w_1 x_n) x_n = 0 \end{cases}$$

That is,
$$\begin{cases} N w_0 + (\sum_{i=1}^N x_i) w_1 = \sum_{i=1}^N y_i \\ (\sum_{i=1}^N x_i) w_0 + (\sum_{i=1}^N x_i^2) w_1 = \sum_{i=1}^N y_i x_i \end{cases}$$

write

$$A = \begin{pmatrix} N & \sum_{i=1}^N x_i \\ \sum_{i=1}^N x_i & \sum_{i=1}^N x_i^2 \end{pmatrix}, \tilde{w} = \begin{pmatrix} w_0 \\ w_1 \end{pmatrix}, \tilde{r} = \begin{pmatrix} \sum_{i=1}^N y_i \\ \sum_{i=1}^N y_i x_i \end{pmatrix}$$

Need to solve: $A \tilde{w} = \tilde{r}$

$$|A| = N \sum_{i=1}^N x_i^2 - (\sum_{i=1}^N x_i)^2 \geq 0$$

$|A| = 0$ if and only if $x_1 = x_2 = \dots = x_N$
all x_i 's are equal

As long as $|A| > 0$, $\tilde{w} = A^{-1} \tilde{r}$

Ex 1

$$f(\tilde{x}) = \tilde{x}^T \tilde{w} = \sum a_i x_i$$

$$\frac{\partial f}{\partial x_i} = a_i$$

then, $\frac{\partial f}{\partial \tilde{x}}$ is a vector whose i -th entry $\frac{\partial f}{\partial x_i} = a_i$

$$\Rightarrow \frac{\partial f}{\partial \tilde{x}} = \tilde{a}$$

Ex 2

$$f(\tilde{x}) = \frac{1}{2} \tilde{x}^T A \tilde{x} \text{ where } A = A^T$$

$$\Rightarrow \frac{\partial f}{\partial \tilde{x}} = A \tilde{x}$$

Ex 3

$$f(A) = \ln(A^{-1})$$

$$\Rightarrow \frac{\partial f}{\partial A} = -(A^{-1})^T$$

Gaussian density in high dimension

give a marginal Gaussian distb for x and a conditional Gaussian distribution for y given x in the form

$$p(\tilde{x}) = N(\tilde{x} | \tilde{\mu}, \Lambda^{-1})$$

$$p(\tilde{y} | \tilde{x}) = N(\tilde{y} | A \tilde{x} + b, L^{-1})$$

$$p(\tilde{y}) = N(\tilde{y} | A \tilde{\mu} + b, L^{-1} + A \Lambda^{-1} A^T)$$

$$p(\tilde{x} | \tilde{y}) = N(\tilde{x} | \Lambda \{A^T L (A \tilde{x} + b) + A \tilde{\mu}\}, \tilde{Z})$$

$$\text{where } \tilde{Z} = (\Lambda + A^T L A)^{-1}$$

Multidimensional Classification

$$\text{Recall: } g_i(\tilde{x}) = \log p(\tilde{x} | C_i) + \log p(C_i)$$

Now $\tilde{x} \in \mathbb{R}^p$

Assume

$$p(\tilde{x} | C_i) = N(\tilde{x} | \tilde{\mu}_i, \Sigma_i)$$

$$= \frac{1}{(2\pi)^{p/2} |\Sigma_i|^{1/2}} \exp\left(-\frac{1}{2}(\tilde{x} - \tilde{\mu}_i)^T \Sigma_i^{-1} (\tilde{x} - \tilde{\mu}_i)\right) + \log p(C_i)$$

consider a sample $\tilde{X} = \{\tilde{x}_n\}_{n=1}^N = \{(\tilde{x}_{n1}, \tilde{x}_{n2})\}_{n=1}^N$
the MLE estimator

$$\hat{\tilde{\mu}}_i = \frac{\sum_{n=1}^N t_{ni} \tilde{x}_n}{\sum_{n=1}^N t_{ni}} =: \tilde{\mu}_i$$

$$\hat{\Sigma}_i = \frac{\sum_{n=1}^N t_{ni} (\tilde{x}_n - \tilde{\mu}_i)(\tilde{x}_n - \tilde{\mu}_i)^T}{\sum_{n=1}^N t_{ni}} =: \hat{\Sigma}_i$$

$$\hat{p}(C_i) = \frac{\sum_{n=1}^N t_{ni}}{N}$$

Multi-dimensional regression

$$y = f(\tilde{x}) + \epsilon, \quad \tilde{x} \in \mathbb{R}^p, f: \mathbb{R}^p \rightarrow \mathbb{R}$$

Estimate $f(\tilde{x})$ with $g(\tilde{x} | \theta)$

given a sample $\tilde{X} = \{\tilde{x}_n\}_{n=1}^N = \{(\tilde{x}_{n1}, \tilde{x}_{n2})\}_{n=1}^N$

Following the same assumption $\tilde{X} \sim N(0, \Sigma)$

smthg to be 1D case, we need to minimize the error function

$$\text{Assume the linear } g(\tilde{x} | \tilde{w}) = w_0 + w_1 x_1 + \dots + w_p x_p$$

Now $\tilde{w}(\tilde{w} | \tilde{X})$

$$= \frac{1}{2} \sum_{n=1}^N (y_n - (w_0 + w_1 x_{n1} + \dots + w_p x_{np}))^2$$

$$\text{write } \tilde{X} = \begin{bmatrix} 1 & x_{11} & x_{12} & \dots & x_{1p} \\ 1 & x_{21} & x_{22} & \dots & x_{2p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{N1} & x_{N2} & \dots & x_{Np} \end{bmatrix} \in \mathbb{R}^{N \times (p+1)}$$

$$\tilde{w} = \begin{pmatrix} w_0 \\ w_1 \\ \vdots \\ w_p \end{pmatrix} \in \mathbb{R}^{p+1}, \quad \tilde{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix} \in \mathbb{R}^N$$

then is

$$\tilde{w}(\tilde{w} | \tilde{X}) = \frac{1}{2} \|\tilde{y} - \tilde{X} \tilde{w}\|^2$$

$$= \frac{1}{2} \|\tilde{y}\|^2 + \frac{1}{2} \tilde{w}^T \tilde{X}^T \tilde{X} \tilde{w} - \tilde{y}^T \tilde{X} \tilde{w}$$

$$\text{setting } \frac{\partial \tilde{L}}{\partial \tilde{w}} = \tilde{X}^T \tilde{X} \tilde{w} - \tilde{X}^T \tilde{y} \stackrel{!}{=} 0$$

$$\text{yields } \tilde{w} = (\tilde{X}^T \tilde{X})^{-1} \tilde{X}^T \tilde{y} =: \hat{\tilde{w}}$$

$$N(\tilde{x} | \tilde{\mu}, \tilde{\Sigma}) = \frac{1}{(2\pi)^{p/2} |\tilde{\Sigma}|^{1/2}} e^{-\frac{1}{2}(\tilde{x} - \tilde{\mu})^T \tilde{\Sigma}^{-1} (\tilde{x} - \tilde{\mu})}$$

it is clear that the Gaussian density depends on $\tilde{x}$ through

$$A^2 = (\tilde{x} - \tilde{\mu})^T \tilde{\Sigma}^{-1} (\tilde{x} - \tilde{\mu})$$

A is called the Mahalanobis distance from $\tilde{\mu}$ to $\tilde{x}$.

this reduces the Euclidean distance $\Sigma = I$

Joint Gaussian implies conditional gaussian

given a joint Gaussian distb $N(\tilde{x}, \tilde{\rho}, \tilde{Z})$
with $\Lambda = \tilde{\Sigma}^{-1}$ and

$$\tilde{x} = \begin{pmatrix} x_a \\ x_b \end{pmatrix}, \quad \tilde{\mu} = \begin{pmatrix} \mu_a \\ \mu_b \end{pmatrix}$$

$$\tilde{\Sigma} = \begin{pmatrix} \Sigma_{aa} & \Sigma_{ab} \\ \Sigma_{ba} & \Sigma_{bb} \end{pmatrix}, \quad \Lambda = \begin{pmatrix} \Lambda_{aa} & \Lambda_{ab} \\ \Lambda_{ba} & \Lambda_{bb} \end{pmatrix}$$

conditional distb

marginal distb

$$p(\tilde{x}_a | \tilde{x}_b) = N(x_a | \mu_a, \Lambda_{aa}^{-1})$$

$$p(\tilde{x}_b) = N(x_b | \mu_b, \Lambda_{bb})$$

From parametric to non-parametric: LASSO regression

Ridge regression

$$\hat{\omega}_{\text{MLE}} = \arg \max_{\omega} \log(p(\omega | \mathcal{D}))$$

$$= \arg \max_{\omega} \log(p(x | \omega)) + \log(p(\omega))$$

Assuming $p(\omega) = \mathcal{N}(\omega | \bar{\omega}, \bar{\sigma}^2 \mathbf{I})$

$$\text{for } p(\omega) = \frac{1}{(\bar{\sigma}^2)^{p/2} (\pi)^{p/2}} e^{-\frac{1}{2\bar{\sigma}^2} \|\omega - \bar{\omega}\|^2}$$

$$\Rightarrow \log p(\omega) = -\frac{1}{2\bar{\sigma}^2} \|\omega - \bar{\omega}\|^2 + \text{const}$$

$$\hat{\omega}_{\text{MLE}} = \arg \max_{\omega} (\log p(x | \omega) + \log p(\omega))$$

$$= \arg \max_{\omega} (\|y - X\omega\|^2 + \frac{\lambda}{2} \|\omega\|^2)$$

6 ridge

Ridge regression yields

$$\frac{\partial \mathcal{L}_{\text{ridge}}}{\partial \omega} = X^T X \omega - X^T y + \lambda \omega = 0$$

$$\Rightarrow (X^T X + \lambda I) \omega = X^T y$$

And $\omega = (X^T X + \lambda I)^{-1} X^T y =: \hat{\omega}_{\text{ridge}}$

We can compare $\hat{\omega}_{\text{MLE}}$ and $\hat{\omega}_{\text{ridge}}$ via SVD decomposition to

$$X = U \Delta V^T \text{ where } U^T U = V^T V = I$$

And $\Delta = \text{diag}(\delta_0, \delta_1, \dots, \delta_p)$

$$\Rightarrow X^T X = V \Delta \Delta^T V^T = V \Delta^2 V^T$$

$$\Rightarrow \hat{\omega}_{\text{ridge}} = (X^T X + \lambda I)^{-1} X^T y$$

$$= (V \Delta^2 V^T + \lambda I)^{-1} V \Delta U^T y$$

$$= V \Delta (\Delta^2 + \lambda I)^{-1} V^T V \Delta U^T y$$

$$= V \Delta \frac{1}{\Delta^2 + \lambda} \Delta U^T y$$

$$= V \text{diag}\left(\frac{\delta_j^2}{\delta_j^2 + \lambda}\right)_{j=0}^p U^T y$$

$$= \sum_{j=0}^p \left(\frac{\delta_j^2}{\delta_j^2 + \lambda}\right) \tilde{u}_j \tilde{y}_j$$

$$= \sum_{j=0}^p \left(\frac{\delta_j^2}{\delta_j^2 + \lambda}\right) \tilde{u}_j \tilde{y}_j$$

On the other hand

$$X \hat{\omega}_{\text{MLE}} = X (X^T X)^{-1} X^T y$$

$$= U \Delta V^T V \Delta^+ V^T V \Delta U^T y$$

$$= U \Delta \Delta^+ U^T y$$

$$= U U^T y$$

$$= \sum_{j=0}^p \tilde{u}_j \tilde{y}_j$$

$\Rightarrow$ Ridge regression gives a shrunken version of estimator compared with regular linear regression.

$$\hat{\omega}_{\text{LASSO}} = \arg \min_{\omega} \frac{1}{2} \|y - X\omega\|^2 + \lambda \|\omega\|_1$$

where $\|\omega\|_1 = \sum_{j=0}^p |\omega_j|$

"no closed-form solution"

LAD estimator with a different prior

$$p(\omega) \propto e^{-\sum_{j=0}^p |\omega_j|}$$

"Laplace prior"

4.2 bias-variance decomposition for regression

In general, suppose we have a loss function $\mathcal{L}(t, y(\vec{x}))$. The overall expected loss is

$$\mathbb{E}[\mathcal{L}] = \int \int \mathcal{L}(t, y(\vec{x})) p(\vec{x}, t) d\vec{x} dt$$

In regression, the loss is given by

$$\mathbb{E}[\mathcal{L}] = \int \int (y(\vec{x}) - t)^2 p(\vec{x}, t) d\vec{x} dt$$

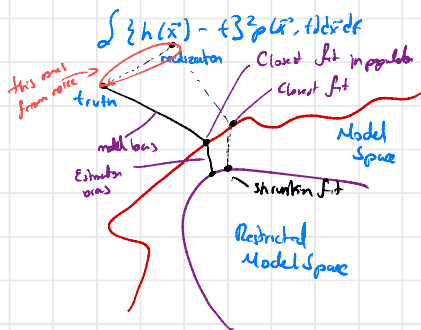
Expected loss

$$\mathbb{E}_0[\{y(\vec{x}; \mathcal{D}) - h(\vec{x})\}^2]$$

$$= \mathbb{E}_0[\underbrace{y(\vec{x}; \mathcal{D}) - h(\vec{x})}_{\text{bias}}^2]$$

$$+ \underbrace{\mathbb{E}_0[\{y(\vec{x}; \mathcal{D}) - \mathbb{E}_0[y(\vec{x}; \mathcal{D})]\}^2]}_{\text{variance}}$$

In regression we have noise as well



Non-parametric density estimation

in parametric methods...

we assumed that data are drawn from a probability distribution of a known form.

- e.g.
 - in classification we assumed that all samples of a class are drawn from the same density.
 - in regression we assumed a linear exponential model with Gaussian additive noise.
- assumptions in parametric methods are not always good.
 - a bad model can lead to large errors.

Non-parametric methods, on the other hand, are assuming that

similar inputs have similar outputs

Non-parametric methods are distribution free

Assumptions

- assume the sample $X = \{X_1, \dots, X_n\}$ in which the observations are drawn i.i.d. from some unknown density $p(\cdot)$.

$\hat{p}(\cdot)$ is our estimator of $p(\cdot)$

- consider the univariate case $x_i \in \mathbb{R}$

Kernel estimator

- a naïve estimator is a special form of the kernel estimator

$$\hat{p}(x) = \frac{1}{Nn} \sum_{i=1}^N K\left(\frac{x-x_i}{h}\right)$$

is called a kernel estimator

a.k.a. Parzen windows.

A kernel estimator is given by

$$\hat{p}(x) = \frac{1}{Nn} \sum_{i=1}^N K\left(\frac{x-x_i}{h}\right)$$

Informally, given $X = \{X_i\}_{i=1}^N$, define a "empirical distribution" as

$$f(x) dx = \frac{1}{N} \mathbb{1}_X(x) = \begin{cases} \frac{1}{N} & \text{if } x \in X \\ 0 & \text{if } x \notin X \end{cases}$$

on the other hand, define

$$\hat{K}(u) = \frac{1}{h} K\left(\frac{u}{h}\right)$$

the convolution of f and $\hat{K}$ is

$$(\hat{f} * \hat{K})(x) = \int_{\mathbb{R}} \hat{f}(u) \hat{K}(x-u) du$$

$$= \frac{1}{N} \sum_{i=1}^N \hat{K}(x-x_i) = \frac{1}{Nn} \sum_{i=1}^N K\left(\frac{x-x_i}{h}\right) = \hat{p}(x)$$

with the formal view, we can also validate that

$$\int_{\mathbb{R}} \hat{p}(x) dx = \int_{\mathbb{R}} (\hat{f} * \hat{K})(x) dx$$

$$= \int_{\mathbb{R}} \int_{\mathbb{R}} \hat{f}(u) \hat{K}(x-u) dx du$$

$$= \int_{\mathbb{R}} \left(\int_{\mathbb{R}} \hat{K}(x-u) dx \right) \hat{f}(u) du$$

$$= \int_{\mathbb{R}} 1 \cdot \hat{f}(u) du = 1$$

Naïve Estimator

Gaussian kernel

$$K(u) = \begin{cases} 1 & \text{if } |u| \leq 1/2 \\ 0 & \text{else} \end{cases} \quad k(u) = \frac{1}{\sqrt{2\pi}} e^{-\frac{u^2}{2}}$$

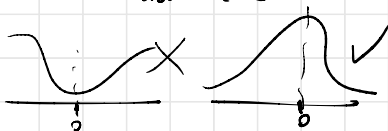
In general, a kernel function K satisfies

(1) $K(u) \geq 0$ for all u

(2) $\int_{\mathbb{R}} K(u) du = 1$

$$\int_{\mathbb{R}} \frac{1}{h} K\left(\frac{u}{h}\right) du = 1$$

(3) The mass of K is concentrated about $u=0$



S.1. Smoothing kernels

regression

• given training set $\mathcal{X} = \{ \tilde{x}_n, y_n \}_{n=1}^N$, where

$$\tilde{x}_n \in \mathbb{R}^d, y_n \in \mathbb{R}$$

• assume $y_n = g(\tilde{x}_n) + \epsilon$

non-parametric regression

• we assume there's a smooth function $g(x)$

running mean smoother

• no fixed bias

$$\hat{g}(x) = \frac{\sum_{i=1}^N w_i(x) y_i}{\sum_{i=1}^N w_i(x)}$$

$$\text{where } w_i(x) = \begin{cases} 1 & \text{if } |x - \tilde{x}_i| \leq 1/2 \\ 0 & \text{else} \end{cases}$$

general case

kernel smoother

$$\hat{g}(x) = \frac{\sum_{i=1}^N k\left(\frac{x - \tilde{x}_i}{h}\right) y_i}{\sum_{i=1}^N k\left(\frac{x - \tilde{x}_i}{h}\right)}$$

S.2. review of K-means

3 steps | introduction

- Expectation (E)

- Maximization (M)

• we can assume K "clusters" of the data is decided by $\mu_1, \dots, \mu_K$

• we would like that the "total distance" between datapoints is small.

$$J = \sum_{n=1}^N \sum_{k=1}^K r_{nk} \| \tilde{x}_n - \tilde{\mu}_k \|^2$$

indicator
• $r_{nk} = 1$ if $\tilde{x}_n$ is assigned to the k th class
• $r_{nk} = 0$ if $\tilde{x}_n$ is NOT assigned to the k th class

to minimize J , we need to deal with both $\{r_{nk}\}$ and $\{\tilde{\mu}_k\}$, which is difficult if we want to find the global minimum.

• instead, we iteratively update $\{r_{nk}\}$ and $\{\tilde{\mu}_k\}$

1. (initialization) randomly initialize $\tilde{\mu}_1, \dots, \tilde{\mu}_K$

2. iteratively do the following until converge:

E-step: for fixed $\tilde{\mu}_1, \dots, \tilde{\mu}_K$ find $\{r_{nk}\}$ that minimize J i.e. assign points to the closest center

M-step: for fixed $\{r_{nk}\}$, find $\tilde{\mu}_1, \dots, \tilde{\mu}_K$ that minimize J i.e. calculate the sample means

BUT - K-means is deterministic

Expectation Maximization for Gaussian Mixture Models

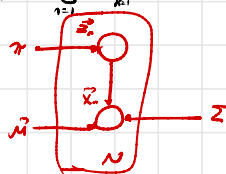
semiparametric approach

• in the parametric approach, we assumed that the sample comes from a known distribution
• in cases when an assumption is introduced and a non-parametric approach is NOT informative, we use a semiparametric approach

MLE for GMM

• if we are given a sample $\mathcal{X} = \{ \tilde{x}_n \}_{n=1}^N$, how do we derive the MLE on the underlying GMM

$$\ln p(\mathcal{X} | \pi, \tilde{\mu}, \Sigma) = \sum_{n=1}^N \ln \left\{ \sum_{k=1}^K \pi_k N(\tilde{x}_n | \tilde{\mu}_k, \Sigma_k) \right\}$$



Gaussian Mixture Models

• let $\tilde{z}$ be a random variable that denotes the clustering

• $\tilde{z}$ is one-hot and $\tilde{z}_n = 1$ implies choosing the k th cluster.

• the marginal distribution over $\tilde{z}$ is given by

$$p(\tilde{z}_k = 1) = \pi_k \Rightarrow p(\tilde{z} = \tilde{e}_k) = \pi_k$$

where $0 \leq \pi_k \leq 1$

$$\sum_{k=1}^K \pi_k = 1$$

similar to multi-class classification we can write

$$p(\tilde{z}) = \prod_{k=1}^K \pi_k^{\tilde{z}_k}$$

in a Gaussian Mixture Model (GMM), $\tilde{z}$ is the conditional distribution $p(\tilde{z} | \tilde{x})$ solution

that is, each cluster is a gaussian, and we can write

$$p(\tilde{x} | \tilde{z}) = \prod_{k=1}^K N(\tilde{x} | \tilde{\mu}_k, \Sigma_k)^{\tilde{z}_k}$$

$$\Rightarrow p(\tilde{x}) = \sum_{\tilde{z}} p(\tilde{z}) p(\tilde{x} | \tilde{z}) = \sum_{k=1}^K \pi_k N(\tilde{x} | \tilde{\mu}_k, \Sigma_k)$$

$\Rightarrow$ By Bayes theorem

$$y(\tilde{z}_n) = p(\tilde{z}_n = 1 | \tilde{x}) = \frac{p(\tilde{z}_n = 1) p(\tilde{x} | \tilde{z}_n = 1)}{\sum_{j=1}^K p(\tilde{z}_j = 1) p(\tilde{x} | \tilde{z}_j = 1)}$$

$$= \frac{\pi_k N(\tilde{x} | \tilde{\mu}_k, \Sigma_k)}{\sum_{j=1}^K \pi_j N(\tilde{x} | \tilde{\mu}_j, \Sigma_j)}$$

$$\text{setting } \frac{\partial J}{\partial \mu_k} = 0 \text{ we have } \sum_{n=1}^N y(\tilde{z}_n) \tilde{x}_n = \sum_{n=1}^N y(\tilde{z}_n) \tilde{\mu}_k$$

$$\text{Next } \frac{\partial J}{\partial \Sigma_k} = \sum_{n=1}^N \frac{\pi_k \tilde{z}_k N(\tilde{x}_n | \tilde{\mu}_k, \Sigma_k)}{\sum_{j=1}^K \pi_j N(\tilde{x}_n | \tilde{\mu}_j, \Sigma_j)}$$

$$= \frac{1}{N} \sum_{n=1}^N y(\tilde{z}_n) \tilde{x}_n \tilde{x}_n^T - \frac{1}{N} \sum_{n=1}^N y(\tilde{z}_n) \tilde{\mu}_k \tilde{\mu}_k^T$$

$$\text{then } \frac{\partial J}{\partial \Sigma_k} = \frac{1}{N} \sum_{n=1}^N y(\tilde{z}_n) \left(\tilde{x}_n \tilde{x}_n^T - \tilde{\mu}_k \tilde{\mu}_k^T \right)$$

$$\text{Solving } \frac{\partial J}{\partial \Sigma_k} = 0 \text{ yields } \sum_{n=1}^N y(\tilde{z}_n) \tilde{x}_n \tilde{x}_n^T = \sum_{n=1}^N y(\tilde{z}_n) (\tilde{x}_n \tilde{\mu}_k^T + \tilde{\mu}_k \tilde{x}_n^T)$$

hence

$$\Sigma_k = \frac{1}{N} \sum_{n=1}^N y(\tilde{z}_n) (\tilde{x}_n \tilde{\mu}_k^T + \tilde{\mu}_k \tilde{x}_n^T)$$

lastly,

$$\max_{\pi} \underbrace{\ln p(x | \pi, \mu, \Sigma) + \lambda \left(\sum_{k=1}^K \pi_k - 1 \right)}_{g(\pi)}$$

$$\frac{\partial g}{\partial \pi_k} = \frac{1}{N} \sum_{n=1}^N \frac{N(x_n | \mu_k, \Sigma_k)}{\sum_{j=1}^K \pi_j N(x_n | \mu_j, \Sigma_j)} = -\pi_k \lambda$$

$$\sum_{n=1}^N \underbrace{\sum_{k=1}^K \gamma(z_{nk})}_{=1} = - \left(\sum_{k=1}^K \pi_k \right) \lambda = -\lambda$$

$$\Rightarrow \lambda = -N$$

$$\sum_{n=1}^N \gamma(z_{nk}) = -\pi_k \lambda$$

$$\pi_k = \frac{N_k}{N}$$

Here MLE for GMM

$$\mu_k = \frac{1}{N_k} \sum_{n=1}^N \gamma(z_{nk}) x_n$$

$$\Sigma_k = \frac{1}{N_k} \sum_{n=1}^N \gamma(z_{nk}) (x_n - \mu_k)(x_n - \mu_k)^T$$

$$\pi_k = \frac{N_k}{N}$$

$$N_k = \sum_{n=1}^N \gamma(z_{nk})$$

$$\text{also } \gamma(z_{nk}) = \frac{\pi_k N(x_n | \mu_k, \Sigma_k)}{\sum_{j=1}^K \pi_j N(x_n | \mu_j, \Sigma_j)}$$

EM for GMM

E-step: expectation
for fixed parameters find the responsibilities

M-step: maximization
for find responsibilities find the company μ_k, Σ_k, π_k

general EM

suppose X is the data matrix
and Z is the corresponding
latent variables

$$\text{then } \ln p(X | \theta) = \ln \left\{ \sum_Z p(X, Z | \theta) \right\}$$

GMM revisited

the likelihood of the complete dataset is

$$p(X, Z | \mu, \Sigma, \pi) = \prod_{n=1}^N \prod_{k=1}^K \pi_k^{z_{nk}} N(x_n | \mu_k, \Sigma_k)^{z_{nk}}$$

taking log yields

$$\ln p(X, Z | \mu, \Sigma, \pi) = \sum_{n=1}^N \sum_{k=1}^K z_{nk} \{ \ln \pi_k + \ln N(x_n | \mu_k, \Sigma_k) \}$$

General EM

- the expectation-maximization algorithm, or EM algorithm, is a general technique for finding maximum likelihood solutions for probabilistic models having latent variables

consider $p(X|A) = \sum_Z p(X, Z|A)$

- assuming Z is discrete and dealing with $p(X, Z|A)$ is easier than $p(X|A)$

- introduce a distribution $q(Z)$ over the latent variables

- claim $\ln p(X|A) = \ln(q, \vec{A}) + \widehat{KL}(q||p)$

where $\ln(q, \vec{A}) = \sum_Z q(Z) \ln \left\{ \frac{p(X, Z|A)}{q(Z)} \right\}$

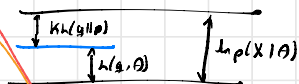
$\widehat{KL}(q||p) = - \sum_Z q(Z) \ln \left\{ \frac{p(Z|X, A)}{q(Z)} \right\}$

this holds since

$\ln p(X, Z|A)$
 $= \ln p(Z|X, A) \cdot p(X|A)$
 $= \ln p(Z|X, A) + \ln p(X|A)$

then $\ln(q, A) = \sum_Z q(Z) \ln \left\{ \frac{p(X, Z|A)}{q(Z)} \right\}$
 $= \sum_Z q(Z) (\ln p(Z|X, A) + \ln p(X|A) - \ln q(Z))$
 $= \sum_Z q(Z) \ln \left(\frac{p(Z|X, A)}{q(Z)} \right) + \sum_Z q(Z) \ln p(X|A)$
 $= -\widehat{KL}(q||p) + \ln p(X|A)$

$\Rightarrow$ since $\ln p(X|A) = \ln(q, A) + \widehat{KL}(q||p)$
 and $\widehat{KL}(q||p) \geq 0$
 therefore, $\ln(q, A)$ is a lower bound of $\ln p(X|A)$



E-step: expectation

maximize $\ln(q, A^{old})$ with respect to q

this q will be pushed to

$q(Z) = p(Z|X, A^{old})$

M-step: maximize $\ln(q, A)$ and maximize $\ln(q, A)$ with respect to A

- often the E-step $g(Z) = p(Z|X, A^{old})$, and therefore the lower bound takes the form

$\ln(q, A) = \sum_Z p(Z|X, A^{old}) \ln g(Z|A)$
 $= \sum_Z p(Z|X, A^{old}) \ln p(Z|X, A^{old})$
 $= Q(A, A^{old}) + \text{const}$

$E[p(Z|X, A^{old}) \ln p(X, Z|A)]$

Spectral Clustering

For both k -means and EM, we need the location of the data instances $\{\tilde{x}_n\}_{n=1}^N$

- However, in a distance-based method, we make decision only based on the relative locations.

- goal of this section we want to get low dimensional features disregarding the original locations

guide for projection

- given $\tilde{x}_n \in \mathbb{R}^d, n=1, \dots, N$
- suppose we want to look at low-dimension projection $\tilde{z}_n \in \mathbb{R}^k, n=1, \dots, N$
- suppose further that we are given a similarity score for each pair of instances
 - $w_{nm} = w_{mn}$ is the similarity between $\tilde{x}_n$ and $\tilde{x}_m$
 - $w_{nn} = 0$ for simplicity

- the goal is to minimize $\frac{1}{2} \sum_{n=1}^N \sum_{m=1}^N \|\tilde{z}_n - \tilde{z}_m\|_{w_{nm}}^2$

one-dimensional case ($k=1$): $z_n \in \mathbb{R}$

$\frac{1}{2} \sum_{n=1}^N \sum_{m=1}^N |z_n - z_m|^2 w_{nm}$
 $= \frac{1}{2} \sum_{n=1}^N \sum_{m=1}^N (z_n^2 w_{nm} + z_m^2 w_{nm} - 2z_n z_m w_{nm})$
 $= \frac{1}{2} \sum_{n=1}^N z_n^2 \sum_{m=1}^N w_{nm} + \frac{1}{2} \sum_{m=1}^N z_m^2 \sum_{n=1}^N w_{nm} - \sum_{n=1}^N \sum_{m=1}^N z_n z_m w_{nm}$
 since $d_n = \sum_{m=1}^N w_{nm} = \sum_{m=1}^N w_{mn}$
 $= \sum_{n=1}^N z_n d_n - \sum_{n=1}^N \sum_{m=1}^N z_n w_{nm} z_m = \vec{z}^T D \vec{z} - \vec{z}^T W \vec{z}$

$D = \text{diag}(d_1, \dots, d_N)$
 $W = (w_{nm})_{n,m=1}^N$

Now that $\frac{1}{2} \sum_{n=1}^N \sum_{m=1}^N |z_n - z_m|^2 w_{nm} = \vec{z}^T (D - W) \vec{z}$ we

- W is the graph Laplacian where (n,m) -th entry is w_{nm} .

- d $w_{nn} = 0$

- D is the degree matrix with

$d_n = \sum_{m=1}^N w_{nm}$

- $L = D - W$ is called the (graph) Laplacian of the similarity graph of our data.

Eigenvector of Laplacian

- now we need to minimize $\sum \lambda_i^2$
- $\vec{z} = (z_1, \dots, z_n)^T$ should be the eigenvector of L .
- note the trivial eigenvector $\vec{e} = \frac{1}{\sqrt{n}}(1, 1, \dots, 1)^T$
- $L\vec{e} = 0\vec{e} = \mathcal{U}\vec{e}$, where n -th entry is $\frac{1}{\sqrt{n}}[D_{nn} - \sum_{i=1}^n w_{ni}] = 0$
- but, this gives the same z_i for all x_i

Hence, we take $\vec{z}$ to be the eigenvector corresponding to the second smallest eigenvalue

Laplacian eigenmap

- the map that maps $\vec{x}_i \in \mathbb{R}^D$ to $\vec{z}_i \in \mathbb{R}^n$ is called the **Laplacian eigenmap**
- where $\vec{z}_i = (z_{i1}, \dots, z_{in})^T$ are the eigenvectors corresponding to the second smallest eigenvalue of L .
- in case $K > 1$, we need to take the next $K-1$ eigenvectors corresponding to the next smallest eigenvalues.
- specifically, we will get a $n \times K$ matrix Z of size n by K whose columns are given by the eigenvectors corresponding to the 2 nd to the $(K+1)$ th smallest eigenvalue of L .

- once we have the features $\{\vec{z}_i\}_{i=1}^N$, we can do clustering on these features instead of the original datapoints $\{\vec{x}_i\}_{i=1}^N$.

- for instance, we can apply k -means on $\{\vec{z}_i\}_{i=1}^N$

Choice of similarity score

- in practice, we can choose to only consider similarities locally (in a neighborhood):

1. edge (first radius) set $w_{nn} = 0$ for $\|\vec{x}_n - \vec{x}_m\| > \epsilon$ for some preset threshold ϵ .

2. or (fixed number of neighbors) use $w_{nn} = 0$ iff $\vec{x}_n$ is among the K nearest neighbors of $\vec{x}_m$ or vice versa.

- one commonly used similarity score for neighboring instances is for some fixed σ ,

$$w_{nm} = \exp\left(-\frac{\|\vec{x}_n - \vec{x}_m\|^2}{2\sigma^2}\right)$$

Entropy

- consider a discrete random variable X
- if we observe a specific value, how much information is received?
- if we have two independent events X and Y , then we hope the information should satisfy:

$$h(X, Y) = h(X) + h(Y)$$

- since $p(X, Y) = p(X)p(Y)$, one good candidate is

$$h(X) = -\log_2 p(X)$$

suppose that a sender wishes to transmit the value of a random variable to a receiver

- the average amount of information transmitted is obtained by taking the expectation with respect to the distribution $p_X(x)$:

$$\begin{aligned} H[X] &= \sum_x p_X(x) (-\log_2 p_X(x)) \\ &= -\sum_x p_X(x) \log_2 p_X(x) \end{aligned}$$

- we call $H[X]$ the entropy of the discrete random variable X
- correspondingly, we call $2^{H(X)}$ the perplexity of X .

- here we understand $p(x) \log_2 p(x) = 0$ if $p(x) = 0$

Shannon's Theorem: entropy is a lower bound on the number of bits needed to transmit the state of a random variable.

- We can use $\ln$ instead of $\log_2$ so we will have **nats** instead of **bits**
- a view from physics: consider a set of N identical objects that are divided amongst a set of bins, s.t. there are n_i objects in the i -th bin.
- we can do this in W ways (called the multiplicity).

$$W = \frac{N!}{\prod_i n_i!}$$

$\Rightarrow$ entropy would then be defined as the logarithm of the multiplicity.

$$H = \frac{1}{N} \ln W = \frac{1}{N} \ln N! - \frac{1}{N} \sum_i \ln n_i!$$

Consider large numbers n_i 's

- by Stirling's Formula,

$$n! \approx \left(\frac{n}{e}\right)^n$$

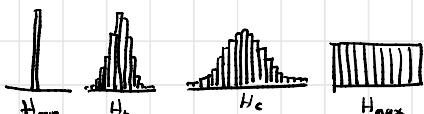
$$\text{thus } \ln(n!) \approx n \ln n - n$$

> the cases

$$a. C > 0 \text{ s.t.}$$

$$c \left(\frac{n}{e}\right)^a \leq n! \leq C \left(\frac{n}{e}\right)^a$$

$$\Rightarrow H = -\sum_{i=1}^N \sum_{j=1}^N \left(\frac{n_{ij}}{N}\right) \ln \left(\frac{n_{ij}}{N}\right) = -\sum p_{ij} \ln p_{ij}$$



$$0 = H_{\min} < H_b < H_c < H_{\max} = \ln(N)$$

- we can interpret the bins as states x_i of a discrete random variable X

where $p(x=x_i) = p_i$

- the entropy of the random var X is then

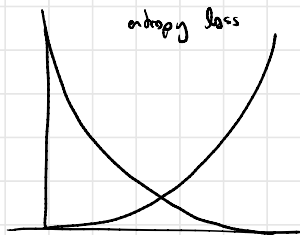
$$H[p] = -\sum_i p(x_i) \ln p(x_i)$$

- the entropy H always satisfies

$$H \geq 0. \text{ It is minimized at zero}$$

$$H=0 \text{ when one of } p_i=1 \text{ and all other } p_j \neq i = 0$$

entropy loss



Application of Entropy

- suppose a random var X takes two values z and y

what is

$$p(X=z) \text{ and } p(X=y)$$

$$p(X=z)$$

- the current state of knowledge about a system is true one with the largest entropy.

$$\text{so } \max_{x,y} -x \ln x - y \ln y \text{ s.t. } x+y=1 \Rightarrow x=y=\frac{1}{2}$$

E.g. Suppose the states are given by $\{x_i\}_{i \in \mathcal{S}}$ and no prior information is given. Following this principle, we need to

$$\text{maximize } -\sum_{i \in \mathcal{S}} p(x_i) \ln p(x_i)$$

$$\text{s.t. } \sum_{i \in \mathcal{S}} p(x_i) = 1$$

using the Lagrange multiplier.

$$\tilde{H} = -\sum_i p(x_i) \ln p(x_i) + \lambda \left(\sum_i p(x_i) - 1 \right)$$

suppose $|\mathcal{S}| = M$

the maximy $\tilde{H}$ yields $p(x_i) = \frac{1}{M}$ and the maximal entropy is

$$H = \ln M$$

that's the true one

$$\frac{d\tilde{H}}{dp_i} = -\left(\frac{p(x_i)}{p(x_i)} + \ln p(x_i)\right) + \lambda = 0$$

$$= -1 - \ln p(x_i) + \lambda = 0$$

$$\lambda - 1 = \ln p(x_i)$$

$$p(x_i) = e^{\lambda-1}$$

$\Rightarrow$ all $p(x_i)$ must be the same

$$\Rightarrow p(x_i) = \frac{1}{M}$$

Continuous case

- differential entropy

$$H[X] = -\int p(x) \ln p(x) dx$$

suppose we have some constraints on 1st and 2nd moments of $p(x)$.

here we need to consider

$$\int_{-\infty}^{\infty} p(x) dx = 1$$

$$\int_{-\infty}^{\infty} x p(x) dx = \mu$$

$$\int_{-\infty}^{\infty} (x-\mu)^2 p(x) dx = \sigma^2$$

when we maximize $H(X)$.

the Lagrange multiplier method requires maximizing

$$-\int p(x) \ln p(x) dx + \lambda_1 \left(\int p(x) dx - 1 \right) + \lambda_2 \left(\int x p(x) dx - \mu \right) + \lambda_3 \left(\int (x-\mu)^2 p(x) dx - \sigma^2 \right)$$

treat $p(x)$ as a variable, w.r.t.

$$-\ln p(x) = -1 + \lambda_1 + \lambda_2 x + \lambda_3 (x-\mu)^2$$

the form of p is

$$p(x) = \exp(-1 + \lambda_1 + \lambda_2 x + \lambda_3 (x-\mu)^2)$$

$\Rightarrow$ it has to be a gaussian

let $X_1, \dots, X_n$ be a sequence of random variables drawn i.i.d. according to $p(X)$

by the law of large numbers we have

$$-\frac{1}{N} \ln p(X_1, \dots, X_n)$$

$$\rightarrow H[-\ln p(X)] = H[p]$$

for $\epsilon > 0$ and any N , the set

$$A_\epsilon^{(N)} = \{ (X_1, \dots, X_n) : |-\frac{1}{n} \ln p(X_1, \dots, X_n) - H[p]| \leq \epsilon \}$$

is said to be a typical set.

conditional entropy.

suppose we have a joint density $p(x, y)$.

if a value of X is already known, the additional information needed to specify the corresponding value of Y is given by $-\ln p(y|x)$

the average additional information called the conditional entropy is

$$H[Y|X] = - \int \int p(y, x) \ln p(y|x) dy dx$$

fact

$$H[X, Y] = H[Y|X] + H[X]$$

since

$$\begin{aligned} & - \int \int p(y, x) \ln p(y|x) dx dy - \int p(x) \ln p(x) dx \\ &= - \int \int p(y, x) \ln p(y|x) dx dy - \int \int p(y, x) dy \ln p(x) dx \\ &= - \int \int p(y, x) [\ln p(y|x) + \ln p(x)] dx dy \\ &= - \int \int p(y, x) \ln(p(y|x) \cdot p(x)) dx dy \\ &= - \int \int p(y, x) \ln p(y, x) dx dy \\ &= H[X, Y] \end{aligned}$$

Kullback-Leibler (KL) divergence

consider some unknown distribution $p(X)$
suppose we model using an approximating distribution $q(X)$.

the additional information required to specify the value of X as a result of using q instead of p is called the relative entropy or KL divergence given by

$$KL(p||q) = - \int p(x) \ln q(x) dx$$

$$= - \left(- \int p(x) \ln p(x) dx \right)$$

$$= - \int p(x) \ln \left(\frac{q(x)}{p(x)} \right) dx$$

$$= - E_p \left[\ln \left(\frac{q}{p} \right) \right] = E_p \left[\ln \left(\frac{p}{q} \right) \right]$$

in general though

$$KL(p||q) \neq KL(q||p)$$

$$\text{fact } KL(p||q) \geq 0$$

Note that $-\ln$ is a convex function therefore,

$$E_p \left[-\ln \left(\frac{q}{p} \right) \right] \geq -\ln \left(E_p \left[\frac{q}{p} \right] \right)$$

by Jensen's theorem that

$$f[E[X]] \leq E[f(X)]$$

$$\text{noting that } E_p \left(\frac{q}{p} \right) = \int p(x) \frac{q(x)}{p(x)} dx$$

$$= \int q(x) dx = 1$$

$$\Rightarrow -\ln(1) = 0, \text{ so}$$

$$KL(p||q) = E_p \left[-\ln \left(\frac{q}{p} \right) \right] \geq -\ln \left(E_p \left[\frac{q}{p} \right] \right) = 0$$

$$KL(p||q) \geq 0$$

Nutritional Information

if X and Y are independent,

$$\text{th } p(X, Y) = p(X)p(Y)$$

for a given $p(X, Y)$, how close is it to being independent,

we can use KL to measure.

$$\begin{aligned} I[X, Y] &= KL(p(X, Y) \| p(X)p(Y)) \\ &= - \int \int p(x, y) \ln \left(\frac{p(x)p(y)}{p(x, y)} \right) dx dy \end{aligned}$$

this is called the mutual information between X and Y

• fact

$$\begin{aligned} I[X, Y] &= H[X] - H[X|Y], \\ &= H[Y] - H[Y|X] \end{aligned}$$

$$\text{• fact: } H[Y|X] \leq H[Y]$$

$$\text{since } KL(p(X, Y) \| p(X)p(Y)) \geq 0$$

$$I[X, Y] \geq 0$$

$$H[Y] - H[Y|X] \geq 0$$

$$H[Y] \geq H[Y|X]$$

independence bound on entropy

• fact: let $X_1, \dots, X_n$ be drawn acedly to $p(X_1, \dots, X_n)$ th

$$H[X_1, \dots, X_n] \leq \sum_{i=1}^n H[X_i]$$

$$\text{since } H[X_1, \dots, X_n] = H[X_1] + H[X_2, \dots, X_n | X_1]$$

$$H[X_2, \dots, X_n | X_1] = H[X_2, \dots, X_n | X_1, X_2] + H[X_2 | X_1]$$

$$\Rightarrow H[X_1, \dots, X_n] \leq H[X_1] + H[X_2 | X_1] + \dots + H[X_n | X_1, \dots, X_{n-1}]$$

since

$$H[X_i | X_1] \leq H[X_i]$$

$$\Rightarrow H[X_1, \dots, X_n] \leq \sum_{i=1}^n H[X_i]$$

KL-divergence

- suppose that data is being generated from an unknown distribution $p(x)$ that we wish to model.
- we can try to approximate using some parametric distrib $q(x|\theta)$.

- one way to determine θ is to minimize KL divergence from $p(x)$ to $q(x|\theta)$ w.r.t. θ . but...

- since we don't know $p(x)$, we instead use our set of observed training points x_n drawn from $p(x)$.

- then

$$KL(p||q) \approx \sum_{n=1}^N \{-\ln q(x_n|\theta) + \ln p(x_n)\}$$

this would be the only term remaining
→ it is equivalent as to minimizing the Maximum likelihood

Mutual Information

- if x and y are independent, then $p(x, y) = p(x)p(y)$

- for a general $p(x, y)$, how close is it to being independent?

$$I[x, y] = KL(p(x, y) || p(x)p(y)) \\ = - \int \int p(x, y) \ln \left(\frac{p(x)p(y)}{p(x, y)} \right) dx dy$$

$$\text{Fact } I[x, y] = H[x] - H[x|y] \\ = H[y] - H[y|x]$$

$$\text{Fact } H[y|x] \leq H[y] \\ \text{since } H[y] - H[y|x] = I[x, y] \geq 0 \\ \Rightarrow H[y] \geq H[y|x]$$

- Fact: let $\bar{x}_1, \dots, \bar{x}_n$ be drawn according to $p(x_1, \dots, x_n)$;

$$H[\bar{x}_1, \dots, \bar{x}_n] \leq \sum_{i=1}^n H[\bar{x}_i]$$

$$H[\bar{x}, \bar{y}] = \underbrace{H[\bar{x}|\bar{y}] + H[\bar{y}]}_{H[\bar{x}, \bar{y}]} \leq \underbrace{H[\bar{x}] + H[\bar{y}]}_{H[\bar{x}, \bar{y}]}$$

KL divergence is Convex

- KL divergence $KL(p||q)$ is convex in (p, q) :
for any $(p_1, q_1), (p_2, q_2), 0 \leq \lambda \leq 1$,

$$KL(\lambda p_1 + (1-\lambda)p_2 || \lambda q_1 + (1-\lambda)q_2) \leq \lambda KL(p_1||q_1) + (1-\lambda) KL(p_2||q_2) \quad (1)$$

Useful 1 log-Sum Inequality

$$\sum_{n=1}^N a_n \ln \left(\frac{a_n}{b_n} \right) \geq \left(\sum_{n=1}^N a_n \right) \ln \left(\frac{\sum_{n=1}^N a_n}{\sum_{n=1}^N b_n} \right)$$

the inequality 1 yields the following calculation

$$\lambda p_1(x) + (1-\lambda)p_2(x) \ln \frac{\lambda p_1(x) + (1-\lambda)p_2(x)}{\lambda q_1(x) + (1-\lambda)q_2(x)}$$

$\leq$

$$\lambda p_1(x) \ln \frac{p_1(x)}{q_1(x)} + (1-\lambda)p_2(x) \ln \frac{p_2(x)}{q_2(x)}$$

which has the form of the log-sum inequality

$$\text{with } a_1 = \lambda p_1, \quad a_2 = (1-\lambda)p_2 \\ b_1 = \lambda q_1, \quad b_2 = (1-\lambda)q_2$$

KL of Gaussian

$$\text{let } p(x) = \mathcal{N}(x|\mu, \Sigma)$$

$$q(x) = \mathcal{N}(x|\bar{\mu}, \bar{\Sigma})$$

$$\text{find } KL(p||q)$$

$$KL(p||q) = - \int p \ln q - (-p \ln p)$$

$$= - \int p(x) \ln p(x) dx$$

$$= - \int p(x) \ln \mathcal{N}(x|\bar{\mu}, \bar{\Sigma}) dx$$

$$= - \int p(x) \ln \left(\frac{1}{(2\pi)^{\frac{D}{2}} |\bar{\Sigma}|^{\frac{1}{2}}} e^{-\frac{1}{2}(x-\bar{\mu})^T \bar{\Sigma}^{-1}(x-\bar{\mu})} \right) dx$$

$$= - \int p(x) \left(\frac{D}{2} \ln(2\pi) - \frac{1}{2} \ln |\bar{\Sigma}| - \frac{1}{2} (x-\bar{\mu})^T \bar{\Sigma}^{-1} (x-\bar{\mu}) \right) dx$$

$$= \underbrace{\frac{D}{2} \ln(2\pi) + \frac{1}{2} \ln |\bar{\Sigma}|}_{C} + \frac{1}{2} \int p(x) (x-\bar{\mu})^T \bar{\Sigma}^{-1} (x-\bar{\mu}) dx$$

$$= \underbrace{\frac{D}{2} \ln(2\pi) + \frac{1}{2} \ln |Z| + \frac{1}{2} \int p(\vec{x}) (\vec{x} - \vec{\mu})^T Z^{-1} (\vec{x} - \vec{\mu}) d\vec{x}}_C$$

$$= C + \frac{1}{2} \int \frac{1}{(2\pi)^D |Z|^{\frac{1}{2}}} e^{-\frac{1}{2} (\vec{x} - \vec{\mu})^T Z^{-1} (\vec{x} - \vec{\mu})} d\vec{x}$$

change of variable $\vec{y} = Z^{-1}(\vec{x} - \vec{\mu})$
 $\ln (\vec{x} - \vec{\mu})^T Z^{-1} (\vec{x} - \vec{\mu}) = \vec{y}^T \vec{y} = \|\vec{y}\|^2$
the jacobian is $|J| = |Z^{-1}| = |Z|^{-1/2}$
thus,

$$= C + \frac{1}{2} \int \frac{1}{(2\pi)^{D/2}} e^{-\frac{1}{2} \|\vec{y}\|^2} \|\vec{y}\|^2 d\vec{y}$$

$$= C + \frac{1}{2} \int N(\vec{y} | \vec{0}, I) \|\vec{y}\|^2 d\vec{y}$$

$$= C + \frac{1}{2} \sum_{j=1}^D \int N(\vec{y} | \vec{0}, I) y_j^2 d\vec{y}$$

$$= C + \frac{1}{2} \sum_{j=1}^D \int_{-\infty}^{\infty} N(y_j | 0, 1) y_j^2 dy_j \dots dy_D$$

$$= C + \frac{1}{2} \sum_{j=1}^D \left(\underbrace{\int_{-\infty}^{\infty} N(y_j | 0, 1) dy_j}_{=1} \cdot \underbrace{\int_{-\infty}^{\infty} N(y_j | 0, 1) y_j^2 dy_j}_{\sigma^2=1} \right)$$

$$= C + \frac{1}{2} \sum_{j=1}^D 1 = C + \frac{D}{2}$$

on the other hand

$$= \int p(\vec{x}) \ln q(\vec{x}) d\vec{x}$$

$$= - \int p(\vec{x}) \ln N(\vec{x} | \vec{0}, I) d\vec{x}$$

$$= - \int p(\vec{x}) \ln \left(\frac{1}{(2\pi)^{D/2}} e^{-\frac{1}{2} \|\vec{x}\|^2} \right) d\vec{x}$$

$$= - \int p(\vec{x}) \ln \left(\frac{1}{(2\pi)^{D/2}} \right) d\vec{x} - \int p(\vec{x}) \ln e^{-\frac{1}{2} \|\vec{x}\|^2} d\vec{x}$$

$$= \frac{D}{2} \ln 2\pi + \frac{1}{2} \int p(\vec{x}) \|\vec{x}\|^2 d\vec{x}$$

$$= \frac{D}{2} \ln 2\pi + \frac{1}{2} \int N(\vec{x} | \vec{0}, I) \|\vec{x}\|^2 d\vec{x}$$

$$= \frac{D}{2} \ln 2\pi + \frac{1}{2} \int N(\vec{x} | \vec{0}, \Sigma) \|\vec{x} + \vec{\mu}\|^2 d\vec{x}$$

$$= \frac{D}{2} \ln 2\pi + \frac{1}{2} \int N(\vec{x} | \vec{0}, \Sigma) \|\vec{x} + \vec{\mu}\|^2 d\vec{x}$$

$$= \frac{D}{2} \ln 2\pi + \frac{1}{2} \int N(\vec{x} | \vec{0}, \Sigma) (\|\vec{x}\|^2 + \|\vec{\mu}\|^2 + 2\vec{x}^T \vec{\mu}) d\vec{x}$$

"odd factor"

$$= \frac{D}{2} \ln 2\pi + \frac{1}{2} \|\vec{\mu}\|^2 + \frac{1}{2} \int N(\vec{x} | \vec{0}, \Sigma) \|\vec{x}\|^2 d\vec{x}$$

$\tilde{C}$

$$= \tilde{C} + \frac{1}{2} \int \frac{1}{(2\pi)^{D/2}} e^{-\frac{1}{2} \|\vec{y}\|^2} \vec{y}^T Z \vec{y} d\vec{y}$$

let $Z = U \Lambda U^T$ be the eigen decomposition

$$= \tilde{C} + \frac{1}{2} \int \frac{1}{(2\pi)^{D/2}} e^{-\frac{1}{2} \|\vec{y}\|^2} \vec{y}^T U \Lambda U^T \vec{y} d\vec{y}$$

$$= \tilde{C} + \frac{1}{2} \int \frac{1}{(2\pi)^D} \exp(-\frac{1}{2} \|\vec{z}\|^2) \vec{z}^T \Lambda \vec{z} d\vec{z}$$

$$= \tilde{C} + \frac{1}{2} \sum_{i=1}^D \int N(\vec{z} | \vec{0}, I) \lambda_i z_i^2 d\vec{z}$$

$$= \tilde{C} + \frac{1}{2} \sum_{i=1}^D \lambda_i$$

$$= \tilde{C} + \frac{1}{2} \text{tr}(Z)$$

therefore $\ln h(p||q) = (\tilde{C} + \frac{1}{2} \text{tr}(Z)) - C + \frac{D}{2}$

$$= \left(\frac{D}{2} \ln(2\pi) + \frac{1}{2} \|\vec{\mu}\|^2 + \frac{1}{2} \text{tr}(Z) \right) - \left(\frac{D}{2} \ln(2\pi) + \frac{1}{2} \ln |Z| + \frac{D}{2} \right)$$

$$= \frac{1}{2} (\|\vec{\mu}\|^2 + \text{tr}(Z) - \ln |Z| - D)$$

Linear SVM

Let $\{\vec{x}_n, t_n\}_{n=1}^N$ be the support - training data for the classification problem, where $t_n \in \{1, -1\}$ is the desired.

- Linear SVM solves

$$\min_{\vec{w}, b} \frac{1}{2} \|\vec{w}\|^2$$

s.t.

$$t_n (\vec{w}^T \vec{x}_n + b) \geq 1 \text{ for any } n$$

- the dual problem is

$$\max_{\alpha} \sum_{n=1}^N \alpha_n - \frac{1}{2} \sum_{n=1}^N \sum_{m=1}^N \alpha_n \alpha_m t_n t_m \vec{x}_n^T \vec{x}_m$$

s.t.

$$\alpha_n \geq 0 \text{ for all } n; \text{ and } \sum_{n=1}^N \alpha_n t_n = 0$$

$$\vec{w} = \sum_{n=1}^N \alpha_n t_n \vec{x}_n$$

by replacing $\vec{x}_n$ by $\phi(\vec{x}_n)$ in the formulation of SVM:

$$\min_{\vec{w}, b} \frac{1}{2} \|\vec{w}\|^2$$

s.t. $t_n (\vec{w}^T \phi(\vec{x}_n) + b) \geq 1 \text{ for any } n$

also, the dual problem becomes

$$\max_{\alpha} \sum_{n=1}^N \alpha_n - \frac{1}{2} \sum_{n=1}^N \sum_{m=1}^N \alpha_n \alpha_m t_n t_m \phi(\vec{x}_n)^T \phi(\vec{x}_m)$$

s.t.

$$\alpha_n \geq 0 \text{ for all } n; \text{ and } \sum_{n=1}^N \alpha_n t_n = 0$$

but instead of finding a good mapping $\phi(\cdot)$, it is easier to define a "Kernel" K , such that

$$K(\vec{x}_n, \vec{x}_m) = \phi(\vec{x}_n)^T \phi(\vec{x}_m)$$

you that, the dual problem becomes

$$\max_{\alpha} \sum_{n=1}^N \alpha_n - \frac{1}{2} \sum_{n=1}^N \sum_{m=1}^N \alpha_n \alpha_m t_n t_m K(\vec{x}_n, \vec{x}_m)$$

s.t.

$$\alpha_n \geq 0 \text{ for all } n;$$

and $\sum_{n=1}^N \alpha_n t_n = 0$

$$y(\vec{x}) = \vec{w}^T \phi(\vec{x}) + b = \sum_{n=1}^N \alpha_n t_n K(\vec{x}, \vec{x}_n) + b$$

Example: let $\vec{x} = [x_1, x_2]^T$

$$\vec{y} = [y_1, y_2]^T$$

- let $\phi(\vec{x}) = [1, \sqrt{2} x_1, \sqrt{2} x_2, \sqrt{2} x_1 x_2, x_1^2, x_2^2]^T$

- then $\phi(\vec{x})^T \phi(\vec{y}) = 1 + 2x_1 y_1 + 2x_2 y_2 + 2x_1 x_2 y_1 y_2 + x_1^2 y_1^2 + x_2^2 y_2^2 = (x_1 y_1 + x_2 y_2 + 1)^2 = (\vec{x}^T \vec{y} + 1)^2 =: K(\vec{x}, \vec{y})$

in practice, we start with K .

popular choices of K :

- polynomial: $K(\vec{x}, \vec{y}) = (\vec{x}^T \vec{y} + 1)^q$

- Radial Basis Function (RBF):

$$K(\vec{x}, \vec{y}) = \exp\left(-\frac{\|\vec{x} - \vec{y}\|^2}{2\sigma^2}\right)$$

- sigmoidal: $K(\vec{x}, \vec{y}) = \tanh(2\vec{x}^T \vec{y} + 1)$

- in general, what K can be chosen?

- if given a K , can we figure out ϕ ?

Gaussian Processes

• a Gaussian process (GP) is a probability distribution over functions $y(\vec{x})$ s.t. the set of values of $y(\vec{x})$ evaluated at an arbitrary set of points $\vec{x}_1, \dots, \vec{x}_N$ jointly has a Gaussian distribution.

• in hydrology also called a Gaussian random field.

• example: Brownian motion

• fact: a Gaussian process is completely determined by

- expectation $E[y(\vec{x})] = \mu(\vec{x})$

- covariance $Cov(y(\vec{x}), y(\vec{x}')) = k(\vec{x}, \vec{x}')$

• we can denote it as $GP(\mu, K)$

• let's stick with the regression case with the model

$$y(\vec{x}) = \vec{w}^T \phi(\vec{x})$$

• consider a prior distribution

$$p(\vec{w}) = N(\vec{w} | \vec{0}, \alpha^{-1} I)$$

• then we get a joint distribution for any collection

$$y(\vec{x}_1), \dots, y(\vec{x}_N)$$

• let $\vec{y} = (y(\vec{x}_1), \dots, y(\vec{x}_N))^T$

$$\vec{y} = \begin{bmatrix} \vec{w}^T \phi(\vec{x}_1) \\ \vdots \\ \vec{w}^T \phi(\vec{x}_N) \end{bmatrix} = \Phi \vec{w}$$

$$E[\vec{y}] = \Phi E[\vec{w}] = \vec{0}$$

$$Cov[\vec{y}] = E[\vec{y} \vec{y}^T] = \Phi E[\vec{w} \vec{w}^T] \Phi^T = \frac{1}{2} \Phi \Phi^T = K$$

• on underlying kernel

$$k(\vec{x}_n, \vec{x}_m) = \theta_0 e^{-\frac{\theta_1}{2} \|\vec{x}_n - \vec{x}_m\|^2} + \theta_2 + \theta_3 \vec{x}_n^T \vec{x}_m$$

GP for regression

consider $t_n = y_n + \epsilon_n$ with $p(t_n | y_n)$

$$= N(t_n | y_n, \sigma^2)$$

• assume the random variable y follows the GP described above.

• the joint distribution of the target values, conditioned on $\vec{y} = (y_1, \dots, y_N)^T$

$$p(\vec{t} | \vec{y}) = N(\vec{t} | \vec{y}, \sigma^2 I_N)$$

the prior is $p(\vec{y}) = N(\vec{y} | \vec{0}, K_N)$

• by a fact of gaussian, the marginal distribution of t is

$$p(t) = \int p(t | \vec{y}) p(\vec{y}) d\vec{y} = N(t | 0, c_n)$$

now C_{N+1} is

$$C_N(\vec{x}_n, \vec{x}_m) = k(\vec{x}_n, \vec{x}_m) + \beta^{-1} \delta_{nm}$$

this also holds if N is replaced by $N+1$

$\int_0^1 \int_{n+1}^m = 0$

also, we can say that $C_N = K + \beta^{-1} I_N$

• consider a test point $\vec{x}_{N+1}$

• want to find $p(t_{N+1} | \vec{t})$

• first consider the joint distribution $p(\vec{t}_{N+1})$ where $\vec{t}_{N+1} = (t_1, \dots, t_N, t_{N+1})^T$

$$p(\vec{t}_{N+1}) = N(\vec{t}_{N+1} | \vec{0}, C_{N+1})$$

$$C_{N+1} = \begin{pmatrix} C_N & \vec{k} \\ \vec{k}^T & c \end{pmatrix} \quad \vec{k} \text{ has elements } k(\vec{x}_n, \vec{x}_{N+1}) \text{ for } n=1, \dots, N$$

$$c = k(\vec{x}_{N+1}, \vec{x}_{N+1}) + \beta^{-1}$$

• then from analysis of Gaussian, $p(t_{N+1} | \vec{t})$ is also Gaussian with mean and variance given by

$$m(\vec{x}_{N+1}) = \vec{k}^T C_N^{-1} \vec{t}$$

$$\sigma^2(\vec{x}_{N+1}) = c - \vec{k}^T C_N^{-1} \vec{k}$$

General Kernel Theory:

reproducing kernel Hilbert space (RKHS)

a function $K: X \times X \rightarrow \mathbb{R}$ is called a kernel on X .

we want to define kernels that can be written as $K(\vec{x}, \vec{x}') = \langle \phi(\vec{x}), \phi(\vec{x}') \rangle$ for any pair of $\vec{x}, \vec{x}'$ in X .

here $\phi: X \rightarrow \mathbb{H}$ is a mapping where $\mathbb{H}$ is a Hilbert space (inner-product space) called a feature space. e.g. $\mathbb{R}^n$

Mercer's Theorem

let $K: X \times X \rightarrow \mathbb{R}$ be a continuous symmetric function. Then K admits the following expansion

$$K(\vec{x}, \vec{x}') = \sum_{j=1}^{\infty} \lambda_j \psi_j(\vec{x}) \psi_j(\vec{x}')$$

with nonnegative numbers $\{\lambda_j\}_{j=1}^{\infty}$ and orthonormal functions $\psi_j(\cdot): X \rightarrow \mathbb{R}$ iff it satisfies the

Mercer condition:

$$\int \int \psi_j(\vec{x}) \psi_k(\vec{x}') f(\vec{x}) f(\vec{x}') d\vec{x} d\vec{x}' \geq 0 \quad \text{for any integrable function } f.$$

$$\text{let } \Phi_j(x) = [\sqrt{\lambda_1} \psi_1(x), \sqrt{\lambda_2} \psi_2(x), \dots]^T$$

$$\text{then } K(\vec{x}, \vec{x}') = \Phi_j(\vec{x})^T \Phi_j(\vec{x}')$$

positive definite symmetric kernel exercise.

let K be a positive definite symmetric kernel. Then for any $\vec{x}, \vec{x}' \in X$,

$$K(\vec{x}, \vec{x}')^2 \leq K(\vec{x}, \vec{x}) K(\vec{x}', \vec{x}')$$

say

$$K = \begin{bmatrix} K(\vec{x}, \vec{x}) & K(\vec{x}, \vec{x}') \\ K(\vec{x}', \vec{x}) & K(\vec{x}', \vec{x}') \end{bmatrix}$$

$$\det K = K(\vec{x}, \vec{x}) K(\vec{x}', \vec{x}') - K(\vec{x}, \vec{x}')^2 \geq 0$$

$$\Rightarrow K(\vec{x}, \vec{x}) K(\vec{x}', \vec{x}') \geq K(\vec{x}, \vec{x}')^2$$

Reproducing Kernel Hilbert Space (RKHS)

let $K: X \times X \rightarrow \mathbb{R}$ be a PDS kernel. Then there exists a Hilbert space (inner-product space) $\mathcal{H}$ and a mapping $\phi: X \rightarrow \mathcal{H}$, s.t. for any $\vec{x}, \vec{x}' \in X$, $K(\vec{x}, \vec{x}')$

$= \langle \phi(\vec{x}), \phi(\vec{x}') \rangle_{\mathcal{H}}$. Furthermore, this $\mathcal{H}$ satisfies the reproducing property: for any $f \in \mathcal{H}$, $f(\vec{x}) = \langle f, K(\vec{x}, \cdot) \rangle_{\mathcal{H}}$.

this $\mathcal{H}$ is called a reproducing kernel Hilbert space (RKHS)

in general, a Hilbert space $\mathcal{H}$ is said to be an RKHS if there exists a function $K(\cdot, \cdot): X \times X \rightarrow \mathbb{R}$ s.t.

1. $K(\vec{x}, \cdot) \in \mathcal{H}$ for any $\vec{x} \in X$

2. $f(\vec{x}) = \langle f(\cdot), K(\vec{x}, \cdot) \rangle_{\mathcal{H}}$ for any $f \in \mathcal{H}$.

nevertheless, if $\mathcal{H}$ is an RKHS, then $K(\cdot, \cdot)$ must be PDS, that is, there is no need to distinguish RKHS from PDS kernels.

uniqueness of RKHS

an RKHS contains a unique reproducing kernel

conversely, a reproducing kernel defines a unique RKHS.

Positive definite symmetric kernel

a function $K: X \times X \rightarrow \mathbb{R}$ is said to be a positive definite symmetric (PDS) kernel if for any $\{\vec{x}_n\}_{n=1}^N \subset X$, the matrix K , whose (n, m) -th entry is given by $K(\vec{x}_n, \vec{x}_m)$, is a symmetric positive semidefinite matrix.

that is to say, the eigenvalues of K are all non-negative. e.g. for any vector $\vec{z}$, $\vec{z}^T K \vec{z} \geq 0$

This K is called the kernel matrix, or the gram matrix associated with K

Suppose $\mathcal{H}$ is a RKHS with PDS kernels k, k' .

$$\text{Then } f(x) = \langle f, k(x, \cdot) \rangle_{\mathcal{H}}$$

$$\text{and } g(x) = \langle g, k'(x, \cdot) \rangle_{\mathcal{H}} \text{ for any } f, g \in \mathcal{H}, x \in \mathcal{X}.$$

Trivially $\langle f, \underbrace{k(x, \cdot) - k'(x, \cdot)}_{=0} \rangle_{\mathcal{H}} = 0$, hence, a RKHS cannot map to more than one reproducing kernel.

Feature Space

Given a RKHS with a PDS kernel k , any Hilbert space (inner-product space) $\mathbb{H}$ s.t. $\exists \phi: \mathcal{X} \rightarrow \mathbb{H}$ with $k(\vec{x}, \vec{x}') = \langle \phi(\vec{x}), \phi(\vec{x}') \rangle$ for any pair of $\vec{x}, \vec{x}'$ in $\mathcal{X}$ is called a **feature space** and this ϕ is called a **feature map**.

• in general, a PDS kernel k does NOT uniquely determine the feature space.

• it does not uniquely determine the dimensionality of feature either.

Mercer's Theorem

Let $K: \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ be a continuous symmetric function. Then K admits the following expansion

$$K(\vec{x}, \vec{x}') = \sum_{j=1}^{\infty} \lambda_j \psi_j(\vec{x}) \psi_j(\vec{x}')$$

with nonnegative numbers $\{\lambda_j\}_{j=1}^{\infty}$ and orthonormal functions $\psi_j(\cdot): \mathcal{X} \rightarrow \mathbb{R}$ iff it satisfies the

↓ Mercer condition:

$$\langle \psi_j, \psi_j \rangle = 0 \int K(\vec{x}, \vec{x}') f(\vec{x}) f(\vec{x}') d\vec{x} d\vec{x}' \geq 0 \quad \|\psi_j\| = 1$$

$$\int \psi_j(\vec{x}) \psi_j(\vec{x}') d\vec{x} d\vec{x}'$$

for any integrable function f .

also for all j

thus K is a reproducing kernel for

$$\mathcal{H} = \left\{ f \mid \sum_{j=1}^{\infty} \frac{\langle f, \psi_j \rangle^2}{\lambda_j} < \infty \right\} \text{ s.t. }$$

$$\langle f, g \rangle_{\mathcal{H}} = \sum_{j=1}^{\infty} \frac{\langle f, \psi_j \rangle \langle g, \psi_j \rangle}{\lambda_j}$$

understanding Mercer's theorem.

Let $\mathcal{X} = \{\vec{x}_n\}_{n=1}^N$, $K = [k(\vec{x}_n, \vec{x}_m)]_{n,m=1}^N$

and $\vec{f}: \mathcal{X} \rightarrow \mathbb{R}^N$ with $f_n = f(\vec{x}_n)$.

• then $\vec{f}^T K \vec{f} \geq 0$ and $K = \sum_{j=1}^N \lambda_j \vec{v}_j \vec{v}_j^T$

• $k(\vec{x}_n, \vec{x}_m) = K_{nm} = (V \Lambda V^T)_{nm} = \sum_j \lambda_j v_{jn} v_{jm}$

general regularization problem

$$\min_{f \in \mathcal{H}} H[f] = \frac{1}{N} \sum_{n=1}^N \ell(y_n, f(\vec{x}_n)) + \lambda \|f\|_{\mathcal{H}}^2$$

search for the best map f from a RKHS $\mathcal{H}$ which contains all possible solutions.

$$\min_{f \in \mathcal{H}} H[f] = \frac{1}{N} \sum_{n=1}^N \ell(y_n, \sum_{j=1}^{\infty} c_j \psi_j(\vec{x}_n)) + \lambda \sum_{j=1}^{\infty} \frac{c_j^2}{\lambda_j}$$

Representer Theorem

Let $K: \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ be a PDS kernel whose RKHS is $\mathcal{H}$. Then, for any non-decreasing function $G: \mathbb{R} \rightarrow \mathbb{R}$ and any loss function ℓ , the optimization problem

$$\min_{f \in \mathcal{H}} H[f] = \frac{1}{N} \sum_{n=1}^N \ell(y_n, f(\vec{x}_n)) + G(\|f\|_{\mathcal{H}})$$

admits a solution of the form

$$f^* = \sum_{n=1}^N c_n K(\vec{x}_n, \cdot) \quad \leftarrow \text{finite}$$

for instance in SVM the optimizer is given by

$$y(\vec{x}) = \sum_{n=1}^N a_n \ell_n K(\vec{x}_n, \vec{x}) + b$$

Sampling Methods: Why and How?

Simple sampling methods

in general, we can evaluate

$$s = \int f(\vec{x}) p(\vec{x}) d\vec{x}$$

if $\vec{x}$ is low dim, we can apply numerical integration:

evaluate the integrand at grid points and take the sum.

But if $\vec{x}$ is high dim.

• suppose we are able to draw i.i.d. samples $\vec{x}_1, \dots, \vec{x}_N$ from the density $p(\vec{x})$.

• then calculate

$$\hat{s} = \frac{1}{N} \sum_{n=1}^N f(\vec{x}_n)$$

as an estimator for $s = \int f(\vec{x}) p(\vec{x}) d\vec{x}$

• how to sample from a given density?

assume we can generate (pseudo-)random numbers uniformly distributed in $(0,1)$.

• suppose then $y = f(z)$ where $z \sim \text{Unif}(0,1)$ is uniformly distributed in $(0,1)$

• then z is "pushed forward" by f to produce y . The uniform density is also "pushed forward" to produce the corresponding density of y , so that the probability of any event A , measured by density p_y , should satisfy.

$$P_y(A) = P_z(f^{-1}(A))$$

$$\int_A p_y(y) dy = \int_{f^{-1}(A)} p_z(z) dz$$

$$\Rightarrow p_y(y) dy = p_z(z) dz$$

$$p_y(y) = \underbrace{p_z(z)}_{=1} \left| \frac{dz}{dy} \right|$$

$$\Rightarrow p_y(y) = \left| \frac{dz}{dy} \right|$$

• Now, we need to design f

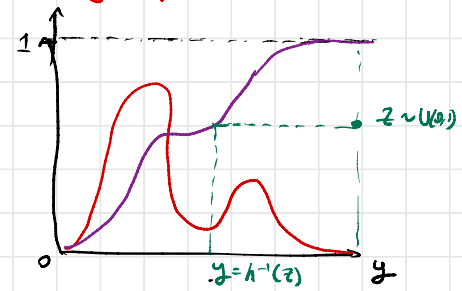
$$z = \int_0^z dz = \int_{-\infty}^y p_y(y) dy =: h(y)$$

(cdf of y)

$$y = h^{-1}(z)$$

• take $f = h^{-1}$, the inverse function of the CDF of y .

$$f = h^{-1}$$



Monte Carlo sampling

• the estimator $\hat{s}_N = \frac{1}{N} \sum_{n=1}^N f(\vec{x}_n)$ is unbiased:

$$\mathbb{E}[\hat{s}_N] = \frac{1}{N} \sum_{n=1}^N \mathbb{E}[f(\vec{x}_n)] = \frac{1}{N} \sum_{n=1}^N s = s$$

• also, by law of large numbers,

$$\lim_{N \rightarrow \infty} \hat{s}_N = s$$

• by the Central Limit Theorem (CLT), the estimator $\hat{s}_N = \frac{1}{N} \sum_{n=1}^N f(\vec{x}_n)$ converges in distribution to a normal distribution with mean s and variance $N^{-1} \text{Var}(f(\vec{x}))$

• by sampling a large number of datapoints, we will approach the true s with a small variance.

Sample from multivariate distrib.

$$p_{\mathbf{z}}(y_1, \dots, y_m) = p_{\mathbf{z}}(z_1, \dots, z_m) \left| \frac{2(z_1, \dots, z_m)}{2y_1, \dots, y_m} \right|$$

Box-Muller method for Gaussian

it starts by generating samples from a uniform distribution inside the unit circle.

1. uniformly over $(z_1, z_2)^T$ from unit disk



- generate $z_1 = 2\tilde{z}_1 - 1$, where $\tilde{z}_1 \sim U(0,1)$
- generate $z_2 = 2\tilde{z}_2 - 1$, where $\tilde{z}_2 \sim U(0,1)$
- reject $(z_1, z_2)^T$ if $z_1^2 + z_2^2 > 1$

2. apply the transform

$$y_1 = z_1 \left(\frac{-2 \ln r^2}{r^2} \right)^{1/2}, y_2 = z_2 \left(\frac{-2 \ln r^2}{r^2} \right)^{1/2} \text{ where } r^2 = z_1^2 + z_2^2$$

then it is easy to verify:

$$p(y_1, y_2) = p(z_1, z_2) \left| \frac{2(z_1, z_2)}{2(y_1, y_2)} \right|$$

$$= \left[\frac{1}{\sqrt{2\pi}} e^{-\frac{y_1^2}{2}} \right] \left[\frac{1}{\sqrt{2\pi}} e^{-\frac{y_2^2}{2}} \right]$$

Rejection Sampling

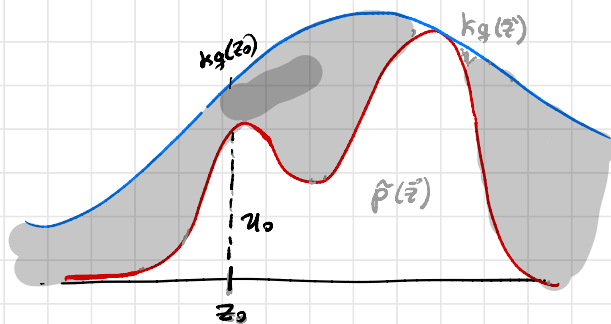
suppose it is easy to evaluate $p(\tilde{\mathbf{z}})$ up to a (possibly unknown) constant $\tilde{c}$.

$$p(\tilde{\mathbf{z}}) = \frac{1}{\tilde{c}} \tilde{p}(\tilde{\mathbf{z}})$$

let $q(\tilde{\mathbf{z}})$, called a proposal distribution, be simpler (we can draw samples from q).

let K be a constant s.t. $K q(\tilde{\mathbf{z}}) \geq \tilde{p}(\tilde{\mathbf{z}})$ for all $\tilde{\mathbf{z}}$.

1. generate $\tilde{\mathbf{z}}_0 \sim q(\tilde{\mathbf{z}})$
2. generate $u_0 \sim U(0, K q(\tilde{\mathbf{z}}_0))$
3. if $u_0 > \tilde{p}(\tilde{\mathbf{z}}_0)$ reject!, else, accept!



the resulting samples are distributed according to $p(\mathbf{z})$ which is the normalized version of $\tilde{p}(\mathbf{z})$.

Importance Sampling

- In $S = \int f(\tilde{\mathbf{x}}) p(\tilde{\mathbf{x}}) d\tilde{\mathbf{x}}$, the specific way of decomposing $p(\tilde{\mathbf{x}}) f(\tilde{\mathbf{x}})$ should not matter.

$$p(\tilde{\mathbf{x}}) f(\tilde{\mathbf{x}}) = q(\tilde{\mathbf{x}}) \frac{p(\tilde{\mathbf{x}}) f(\tilde{\mathbf{x}})}{q(\tilde{\mathbf{x}})}$$

- we can sample $\frac{p f}{q}$ from q instead of sampling f from p .

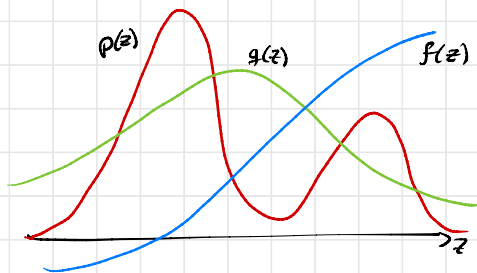
instead of calculating

$$\hat{S}[p] = \frac{1}{N} \sum_{n=1}^N f(\tilde{\mathbf{x}}_n), \quad \tilde{\mathbf{x}}_n \sim p$$

calculate

$$\hat{S}[q] = \frac{1}{N} \sum_{n=1}^N \frac{p(\tilde{\mathbf{x}}_n) f(\tilde{\mathbf{x}}_n)}{q(\tilde{\mathbf{x}}_n)}, \quad \tilde{\mathbf{x}}_n \sim q$$

Importance sampling addresses the problem of evaluating the expectation of a function $f(\tilde{\mathbf{z}})$ with respect to a distribution $p(\tilde{\mathbf{z}})$ from which it is difficult to draw samples directly. Instead, samples $\{\tilde{\mathbf{z}}^{(i)}\}$ are drawn from a simpler distribution $q(\tilde{\mathbf{z}})$ and the corresponding terms in the sum are weighted by the ratios $p(\tilde{\mathbf{z}}^{(i)})/q(\tilde{\mathbf{z}}^{(i)})$.



Markov Chain Monte Carlo

in practice it is often infeasible to sample from q or any good q , due to curse of dimensionality.

idea: build a Markov chain whose stationary distribution is p .

Markov Chain

assume that x has countably many states, say $x \in \mathcal{X}$.

we initialize some distribution $q^{(0)}$.

hope: construct a Markov chain $\{x^{(s)}\}_{s=0}^{\infty}$ so that $\{q^{(s)}(x)\}$ converges to $p(x)$.

for each probability distribution q we describe it as a vector $\vec{v}$ whose i -th entry is given by $v_i = q(x=i)$.

by the Markov property

$$q^{(s+1)}(x') = \sum_x q^{(s)}(x) T(x'|x)$$

transition probability from x to x'

we assume homogeneity: the transition probability does not change with s .

for the countable case, using the transition matrix A .

$$A_{x',j} = T(x'=i | x=j)$$

then $\vec{v}^{(s)} = A^s \vec{v}^{(0)}$

recursively $\vec{v}^{(s)} = A^s \vec{v}^{(0)}$

under some mild conditions, this process converges to a stationary distribution p represented by a vector $\vec{v}^*$:

$$A \vec{v}^* = \vec{v}^*$$

then almost surely, suppose $x_1, \dots, x_N$ are drawn from such a Markov chain.

$$\lim_{N \rightarrow \infty} \frac{1}{N} \sum_{n=1}^N f(x_n) = \mathbb{E}_{x \sim p(x)} f(x) = \int f(x) p(x) dx$$

Metropolis - Hastings (MH)

assume we can evaluate $\tilde{p}(\vec{x}) = p(\vec{x}) / Z$ for some (possibly unknown) Z .

at the beginning, choose a conditional density function, the proposal kernel q .

1. initialize $\vec{x}^{(0)}$

2. iterate: at step s ,

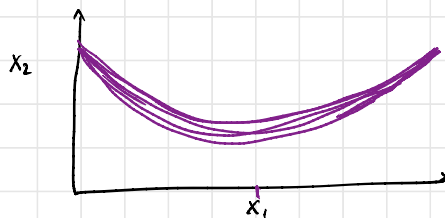
1) generate $\vec{y}^{(s+1)} \sim q(\vec{y} | \vec{x}^{(s)})$

2) take $\vec{x}^{(s+1)} = \begin{cases} \vec{y}^{(s+1)} & \text{with probability } p(\vec{x}^{(s)}, \vec{y}^{(s+1)}) \\ \vec{x}^{(s)} & \text{with probability } 1 - p(\vec{x}^{(s)}, \vec{y}^{(s+1)}) \end{cases}$

$$\text{where } p(x, y) = \min \left\{ \frac{\tilde{p}(y) q(x|y)}{\tilde{p}(x) q(y|x)}, 1 \right\}$$

Example: Rosenbrock density

$$p(x_1, x_2) \propto \tilde{p}(x_1, x_2) = \exp \left(- \frac{(1-x_1)^2 + 100(x_2-x_1)^2}{20} \right)$$



1. initialize $\vec{x}^{(0)} \sim U[-5, 5]^2$

2. take the proposal kernel to be $q(\vec{y} | \vec{x}) = \mathcal{N}(\vec{y} | \vec{x}, \sigma^2 I)$, where $\sigma^2 = 0.1$.

how does MH work?

a sufficient condition for a Markov chain to have a stationary distribution is that it is "reversible":

discrete case: there exists $\vec{v}^*$ s.t.

$$A_{i,j} v_j = A_{j,i} v_i$$

continuous case: there exists a distribution π over the state space s.t.

$$\pi(y) T(x|y) = \pi(x) T(y|x)$$

however, since our proposed q is arbitrary, in general

$$\pi(\tilde{y})q(\tilde{x}|\tilde{y}) \neq \pi(\tilde{x})q(\tilde{y}|\tilde{x}).$$

nonetheless, say for certain $\tilde{x}$ and $\tilde{y}$, w.l.o.g.

$$q(\tilde{y}|\tilde{x})\pi(\tilde{x}) \geq q(\tilde{x}|\tilde{y})\pi(\tilde{y})$$

that is, $\rho'(\tilde{x}, \tilde{y}) = \frac{q(\tilde{x}|\tilde{y})\pi(\tilde{y})}{q(\tilde{y}|\tilde{x})\pi(\tilde{x})}$

, let $\rho(x, y) = \min\{1, \rho'(\tilde{x}, \tilde{y})\}$

the above is equivalent to

$$q(\tilde{y}|\tilde{x})\pi(\tilde{x})\rho(\tilde{x}, \tilde{y}) = q(\tilde{x}|\tilde{y})\pi(\tilde{y})\rho(\tilde{y}, \tilde{x})$$

hence, we realize that the transition kernel is nothing but

$$q(\tilde{y}|\tilde{x})\rho(\tilde{x}, \tilde{y})$$

Gibbs sampling

• the MH algorithm does not leverage any structure of $p(\tilde{x})$.

• consider $p(\tilde{x}) = p(x_1, \dots, x_M)$

• suppose it is easy to sample from

$$p(x_i | \tilde{x}_{-i})$$

$$\uparrow$$

$$x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_M$$

• at each step s with $\tilde{x}^{(s)} = (x_1^{(s)}, \dots, x_M^{(s)})$

1. uniformly sample an index from $1, \dots, M$

2. draw a sample $z \sim p(x_i | \tilde{x}_{-i}^{(s)})$

3. set $\tilde{x}^{(s+1)} = (x_1^{(s)}, \dots, x_{i-1}^{(s)}, z, x_{i+1}^{(s)}, \dots, x_M^{(s)})$

it can be thought of as a special case of MH

where $q(\tilde{y}|\tilde{x}) = p(y_i | \tilde{x}_{-i})$

$$\rho(\tilde{x}, \tilde{y}) = \frac{\pi(\tilde{y})q(\tilde{x}|\tilde{y})}{\pi(\tilde{x})q(\tilde{y}|\tilde{x})} = \frac{p(y_i | \tilde{x}_{-i})\pi(\tilde{y}_{-i})p(x_i | \tilde{y}_{-i})}{p(x_i | \tilde{x}_{-i})\pi(\tilde{x}_{-i})p(y_i | \tilde{x}_{-i})} = 1$$

PAC learning

settings: basics

- domain set: $\mathcal{X}$ - the set of objects we wish to label
- label set: $\mathcal{Y}$ - the labels
 - for this moment, we consider binary classification, so $\mathcal{Y} = \{0, 1\}$

• learner's input: $S = ((x_1, y_1), \dots, (x_m, y_m)) \in (\mathcal{X} \times \mathcal{Y})^m$
 the sample of training data

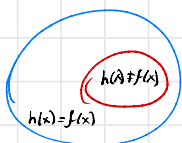
• learner's output: $h: \mathcal{X} \rightarrow \mathcal{Y}$ a rule for prediction
 - usually, we require this h is chosen from a hypothesis class $\mathcal{H}$.

• let f be the correct classifier: we want $h \approx f$

• given data distribution D , we can define the error of h w.r.t. f to be

$$L_{D,f}(h) = \mathbb{P}_{x \sim D}[h(x) \neq f(x)] =: D(\{x \in \mathcal{X} : h(x) \neq f(x)\})$$

the distribution D is the true data distribution, not the distribution of the training data we sample.



• on the other hand, the training error (empirical error) of h is defined to be

$$L_S(h) := \frac{|\{i \in [m] : h(x_i) \neq y_i\}|}{m}$$

where $[m] = \{1, \dots, m\}$

• by training we can achieve the empirical risk minimizer (ERM)

$$h_S \in \arg\min_{h \in \mathcal{H}} L_S(h)$$

• want: h_S is correct for examples not in S
 hope: perfect h .

• assume:

1. each x_i is sampled i.i.d. according to D
2. there exists $h^* \in \mathcal{H}$ s.t. $L_{D,f}(h^*) = 0$ (realizability assumption).

• remark: the realizability assumption automatically implies that $L_S(h^*) = 0$

• moreover, since $h_S \in \arg\min_{h \in \mathcal{H}} L_S(h)$, we also have $L_S(h_S) \leq 0$

• hope: find a perfect algorithm for finding h so that $L_{D,f}(h) = 0$

claim: there does NOT exist an algorithm h for which $L_{D,f}(h) = 0$.

• why?

• for every $\epsilon \in (0, 1)$, consider $\mathcal{X} = \{x_1, x_2\}$ where $D(\{x_1\}) = 1 - \epsilon$, $D(\{x_2\}) = \epsilon$

• the training set contains m i.i.d. samples
 • it is possible that we don't see x_2 at all in the training set.

• for any algorithm would not be able to determine the value of x_2

• we only hope for $L_{D,f}(h) \leq \epsilon$ for a given ϵ

• this ϵ is called the accuracy parameter.

• considering again that $\mathcal{X} = \{x_1, x_2\}$ and

$$D(\{x_1\}) = 1 - \epsilon, \quad D(\{x_2\}) = \epsilon, \quad \text{for}$$

there is a choice ϵ^m that we keep simply x_2 .

• we allow, however, our algorithm to fail with probability δ

• this δ is a confidence parameter.

can we be **probably, approximately** correct?

• let $S/x = (x_1, \dots, x_m)$ be the instances of the training set.

• we would like to upper bound the probability that h_S has large error.

• that is, we'd like to find an upper bound of

$$D^m(\{S/x : L_{D,f}(h_S) > \epsilon\})$$

• let $\mathcal{H}_\epsilon$ be the set of **bad hypotheses**

$$\mathcal{H}_\epsilon = \{h \in \mathcal{H} : L_{D,f}(h) > \epsilon\}$$

• let $\mathcal{M}$ be the set of **misleading samples**

$$\mathcal{M} = \{S/x : \text{there exists } h \in \mathcal{H} \text{ s.t. } L_S(h) = 0\}$$

• by realizability $L_S(h^*) = 0$

• therefore $\{S/x : L_{D,f}(h_S) > \epsilon\} \subset \mathcal{M}$

because if $S/x \in \mathcal{M}$, then $h_S \in \mathcal{H}_\epsilon$

we have $\{S|x: L_{(D,f)}(h_S) > \epsilon\} \subset M$, we

$$D^n(\{S|x: L_{(D,f)}(h_S) > \epsilon\})$$

$$\leq D^n(M)$$

$$= D^n(\{S|x: \exists h \in \mathcal{H}_0, h_S(h) = 0\})$$

$$= D^n(\bigcup_{h \in \mathcal{H}_0} \{S|x: h_S(h) = 0\})$$

$$\leq \sum_{h \in \mathcal{H}_0} D^n(\{S|x: h_S(h) = 0\})$$

$$\prod_{i=1}^n P(\{x_i: h(x_i) = f(x_i)\}) \leq |\mathcal{H}_0| e^{-\epsilon m}$$

$$= (1 - L_{(D,f)}(h))^m$$

$$\leq (1 - \epsilon)^m \leq e^{-\epsilon m}$$

$$\leq |\mathcal{H}| e^{-\epsilon m}$$

$$\leq \delta$$

$$-\epsilon m \leq \log\left(\frac{\delta}{|\mathcal{H}|}\right)$$

$$m \geq \frac{1}{\epsilon} \log\left(\frac{|\mathcal{H}|}{\delta}\right)$$

Probably Approximately Correct (PAC) learning

A hypothesis class $\mathcal{H}$ is said to be PAC learnable if there exists a function $m_{\mathcal{H}}: (0, 1)^2 \rightarrow \mathbb{N}$ and a learning algorithm with the following property:

- for every $\epsilon, \delta \in (0, 1)$, for any distribution D over $\mathcal{X}$, and for any labeling function $f: \mathcal{X} \rightarrow \{0, 1\}$ if the realizability assumption holds, w.r.t. $\mathcal{H}, D, f$, then,

- when running the learning algorithm on $m \geq m_{\mathcal{H}}(\epsilon, \delta)$ i.i.d. examples generated by D

- with probability of at least $1 - \delta$

$$L_{(D,f)}(h) \leq \epsilon$$

- $m_{\mathcal{H}}: (0, 1)^2 \rightarrow \mathbb{N}$ is called the sample complexity

- in the previous section, we have shown:

- every finite hypothesis class is PAC learnable with sample complexity

$$m_{\mathcal{H}}(\epsilon, \delta) \leq \left\lceil \frac{\log(|\mathcal{H}|/\delta)}{\epsilon} \right\rceil$$

agnostic PAC learning

in practice, the realizability assumption is too restrictive.

consider the data distribution D over $\mathcal{X} \times \mathcal{Y}$: pairs of a datapoint and a label

$$L_D(h) = P_{(x,y) \sim D}[h(x) \neq y]$$

$$:= D(\{(x,y): h(x) \neq y\})$$

(compare with what we had before:

$$L_{D,f}(h) = P_{x \sim D}[h(x) \neq f(x)]$$

$$:= D(\{x \in \mathcal{X}: h(x) \neq f(x)\})$$

A hypothesis class $\mathcal{H}$ is said to be **agnostic PAC learnable** if there exists a function $m_{\mathcal{H}}: (0, 1)^2 \rightarrow \mathbb{N}$ and a learning algorithm with the following property:

for every $\epsilon, \delta \in (0, 1)$, for every distribution D over $\mathcal{X} \times \mathcal{Y}$

- when running the learning algorithm on $m \geq m_{\mathcal{H}}(\epsilon, \delta)$ i.i.d. examples generated by D , the algorithm returns a hypothesis h s.t. with probability of at least $1 - \delta$

$$L_D(h) \leq \min_{h' \in \mathcal{H}} L_D(h') + \epsilon$$

example: Bayes optimal predictor

• given any D over $\mathcal{X} \times \{0,1\}$, the best label predicting function will be (the naive Bayes classifier in lecture 1).

$$f_0(x) = \begin{cases} 1 & \text{if } P(y=1|x) \geq 1/2 \\ 0 & \text{otherwise} \end{cases}$$

• we can prove that for this classifier, is the best possible in the sense that $L_D(f_0) \leq L_D(g)$ for any other classifier g , i.e.,

$$f_0 = \arg \min_{h' \in \mathcal{H}} L_D(h')$$

• however, since we don't know D , we don't know f_0 ; therefore, we can only hope to be "approximately optimal":

$$L_D(h) \leq \min_{h' \in \mathcal{H}} L_D(h') + \epsilon \\ = L_D(f_0) + \epsilon$$

beyond binary classification

• we can generalize the definition further by considering a general loss function: $\ell: \mathcal{Y} \times \mathcal{Z} \rightarrow \mathbb{R}_+$

• this $\mathcal{Z}$ is a general set:

– we used to take $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$

– but for instance, in supervised tasks, we could have $\mathcal{Z} = \mathcal{X}$.

• now we need to consider data distributions for this $\mathcal{Z}$.

• given this loss function ℓ , the error w.r.t. the data distribution D will be

$$L_D(h) := \mathbb{E}_{z \sim D} [\ell(h, z)]$$

A hypothesis class $\mathcal{H}$ is said to be **agnostic PAC learnable** w.r.t. a set $\mathcal{Z}$ and a loss function $\ell: \mathcal{H} \times \mathcal{Z} \rightarrow \mathbb{R}_+$ if there exists a function $m_{\mathcal{H}}: (0,1)^2 \rightarrow \mathbb{N}$ and a learning algorithm with the following property:

• for every $\epsilon, \delta \in (0,1)$, for every distribution D over $\mathcal{Z}$, when running the learning algorithm on $m \geq m_{\mathcal{H}}(\epsilon, \delta)$ i.i.d. examples generated by D , the algorithm returns a hypothesis h s.t.

• with probability of at least $1 - \delta$

$$L_D(h) \leq \min_{h' \in \mathcal{H}} L_D(h') + \epsilon$$

• where $L_D(h) := \mathbb{E}_{z \sim D} [\ell(h, z)]$

learning via uniform convergence.

ϵ -representative sample

• a training set S is called ϵ -representative if for any $h \in \mathcal{H}$,

$$|L_S(h) - L_D(h)| \leq \epsilon$$

• assume S is $\epsilon/2$ -representative, then any output $h_S \in \arg \min_{h \in \mathcal{H}} L_S(h)$ satisfies $L_D(h_S) \leq \min_{h \in \mathcal{H}} L_D(h) + \epsilon$

Uniform Convergence Property

we say that a hypothesis class $\mathcal{H}$ has the **uniform convergence property (UCP)** if there exists a function $m_{\mathcal{H}}^{\text{UC}}: (0,1)^2 \rightarrow \mathbb{N}$ s.t.

for every $\epsilon, \delta \in (0,1)$ and for every D over $\mathcal{Z}$, if S is a sample of $m \geq m_{\mathcal{H}}^{\text{UC}}(\epsilon, \delta)$ examples sampled i.i.d. according to D , then

with probability of at least $1 - \delta$, S is ϵ -representative

i.e. $\text{UCP} \Rightarrow \text{agnostic PAC}$

No Free lunch Theorem

universal learner

- in previous discussion, we assume that there is a hypothesis class $\mathcal{H}$ which serves as the search space for our model h .
- we then find the ERM $h_S \in \arg\min_{h \in \mathcal{H}} L_S(h)$
- $\mathcal{H}$ is a prior belief, determined by the task

the fundamental question of the NFL theorem is "is this prior belief necessary?"

in other words

is it possible to have a universal learner that works for any task?

more technically

is there an algorithm that outputs a low-risk h as long as it receives a large number of training data?

equivalently

does there exist a learning algorithm A and a training set size m , s.t.

for every distribution D , if A receives m i.i.d. examples from D , there is a high chance it outputs a predictor h with a low risk?

Theorem

Let A be any learning algorithm. For the task of binary classification w.r.t. the 0-1 loss over a domain $\mathcal{X}$. Let m , the size of the training set, be any number with $m < |\mathcal{X}|/2$. Then, there exists a distribution D over $\mathcal{X} \times \{0, 1\}$ s.t.:

1. There exists a function $f: \mathcal{X} \rightarrow \{0, 1\}$ with $L_D(f) = 0$.
2. With probability of at least $\frac{1}{4}$ over the choice of $S \sim D^m$, we have $L_D(h) \geq \frac{1}{8}$ where $h = A(S)$ is the output of the algorithm.

Proof

- let's proof the no-free-lunch theorem (NFL)
- consider $C \subseteq \mathcal{X}$ s.t. $|C| = 2m$
- there are $T = 2^{2m}$ possible functions that map from C to $\{0, 1\}$, which are named $f_1, \dots, f_T$.
- for each $i \in [T] = \{1, \dots, T\}$, define D_i to be the following distribution

$$D_i(x, y) = \begin{cases} \frac{1}{|C|} & \text{if } y = f_i(x) \\ 0 & \text{else} \end{cases}$$

claim: For every algorithm A that receives a training set of m examples from $C \times \{0, 1\}$,

$$\max_{i \in [T]} \mathbb{E}_{S \sim D_i} [L_D(A(S))] \geq \frac{1}{4}$$

this means that for every algorithm A' that receives a training set of m examples from $\mathcal{X} \times \{0, 1\}$, there exists a function $f: \mathcal{X} \rightarrow \{0, 1\}$ and a distribution D s.t. $L_D(f) = 0$ and

$$\mathbb{E}_{S \sim D^m} [L_D(A'(S))] \geq \frac{1}{4}$$

- there are $K = (2^m)^m$ possible sequences of m examples from C
- denote them by $S_1, \dots, S_K$
- say $S_j = (x_1, \dots, x_m)$ and denote $S_j^i = ((x_1, f_i(x_1)), \dots, (x_m, f_i(x_m)))$

if the distribution is D_i , then the possible training sets are $S_1^i, S_2^i, \dots, S_K^i$

$$\mathbb{E}_{S \sim D_i^m} [L_D(A(S))] = \frac{1}{K} \sum_{j=1}^K L_D(A(S_j^i))$$

we just need to show that the above is no less than $1/4$

$$\begin{aligned} \max_{S \in \mathcal{H}} \frac{1}{K} \sum_{j=1}^K L_{0,i}(A(S_j^i)) &\geq \frac{1}{T} \sum_{i=1}^T \frac{1}{K} \sum_{j=1}^K L_{0,i}(A(S_j^i)) \\ &= \frac{1}{K} \sum_{j=1}^K \frac{1}{T} \sum_{i=1}^T L_{0,i}(A(S_j^i)) \\ &\geq \min_{j \in [K]} \frac{1}{T} \sum_{i=1}^T L_{0,i}(A(S_j^i)) \end{aligned}$$

- fix some $j \in [K]$; denote $S_j = (x_1, \dots, x_m)$
- let $v_1, \dots, v_p$ be the examples in C not appearing in S_j
- since $|C| = 2m$, we have $p \geq m$
- therefore, for every $h: C \rightarrow \{0, 1\}$ and every i , we have

$$\begin{aligned} L_{0,i}(h) &= \frac{1}{2m} \sum_{x \in C} \mathbb{1}[h(x) \neq f_i(x)] \\ &\geq \frac{1}{m} \sum_{r=1}^p \mathbb{1}[h(v_r) \neq f_i(v_r)] \\ &\geq \frac{1}{2p} \sum_{r=1}^p \mathbb{1}[h(v_r) \neq f_i(v_r)] \\ &\geq \frac{1}{2} \min_{r \in [p]} \frac{1}{T} \sum_{i=1}^T \mathbb{1}[A(S_j^i)(v_r) \neq f_i(v_r)] = \frac{1}{4} \end{aligned}$$

$\Rightarrow$ this will be right half of the tree.

we have showed that

$$\begin{aligned} \max_{j \in [K]} \mathbb{E}_{S \sim D^m} [L_{0,i}(A(S))] &\geq \min_{j \in [K]} \frac{1}{T} \sum_{i=1}^T L_{0,i}(A(S_j^i)) \\ &\geq \min_{j \in [K]} \frac{1}{4} = \frac{1}{4} \end{aligned}$$

summary of the NFIH theorem.

Let A be any learning algorithm for the task of binary classification w.r.t. 0-1 loss over a domain X . Let m , the size of the training set, be any number with $m < |X|/2$. Then there exists a distribution D over $X \times \{0, 1\}$ s.t.

1. There exists a function $f: X \rightarrow \{0, 1\}$ with $L_0(f) = 0$
2. With probability of at least $\frac{1}{4}$ over the choice of $S \sim D^m$, we have $L_D(h) \geq \frac{1}{8}$ where $h = A(S)$ is the output of the algorithm.

no restriction on $\mathcal{H} \Rightarrow$ not PAC learnable

corollary. let X be an infinite domain set and let $\mathcal{H}$ be the set of all functions from X to $\{0, 1\}$. Then, $\mathcal{H}$ is not PAC learnable.

Vapnik-Chervonenkis (VC)

Dimension

- according to the No-Free-lunch theorem, if there is no restriction on the hypothesis class $\mathcal{H}$, then for any learning algorithm, there exists a distribution on which it performs poorly.
- is it because that $|\mathcal{H}| = \infty$???

finite-size classes may be learnable

Let $\mathcal{H}$ be the set of threshold functions over the real line:

- $\mathcal{H} = \{h_a : a \in \mathbb{R}\}$ where $h_a(x) = \mathbb{1}[x < a](x)$
- the $|\mathcal{H}| = \infty$ since a can be any real number.

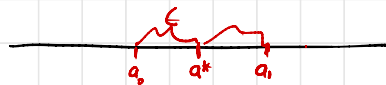
Claim: $\mathcal{H} = \{h_a : a \in \mathbb{R}\}$ is PAC-learnable

Proof: Let a^* be a threshold s.t. the hypothesis

$$h^*(x) := h_{a^*}(x) = \mathbb{1}[x < a^*](x)$$

$$\text{across} \\ L_D(h^*) = 0$$

a^* exists because of the realizability condition



Let $a_0 < a^* < a_1$ be s.t.

$$\mathbb{P}_D(a_0 < x < a^*) = \mathbb{P}(a^* < x < a_1) = \epsilon$$

$$\text{"} \quad \text{"} \\ D(a_0 < x < a^*) \quad D(a^* < x < a_1)$$

if a_0 does not exist, then set $a_0 = -\infty$

if a_1 does not exist, then set $a_1 = \infty$

given a target set S .

x x x x x x x

o o o o o o

$$\text{Let } b_0 = \max_{x \in \mathbb{R}} \{ (x, 1) \in S \}$$

"the maximum x in $S|_x$ whose label is 1"

$$b_1 = \min_{x \in \mathbb{R}} \{ (x, 0) \in S \}$$

"the minimum x in $S|_x$ whose label is 0"

if no b_0 exists, set $b_0 = -\infty$

if not b_1 exists, set $b_1 = \infty$

Let b_S be the threshold that minimizes the training loss.

$$\text{That is, } L_S(h_{b_S}) = 0$$

say $h_S = h_{b_S}$

b_S is a threshold that is correct for all data in S .

$$\text{Also, } b_0 < b_S \leq b_1$$

Therefore, a sufficient condition for

$$L_D(h_S) \leq \epsilon \quad (\text{approximately correct})$$

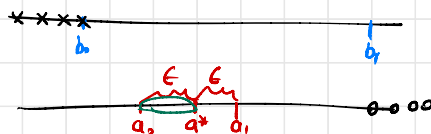
is that both $b_0 \geq a_0$ and $b_1 \leq a_1$



in other words, let $m \in |S|$

$$\mathbb{P}_D(h_S(h_S) > \epsilon) \leq \mathbb{P}_D(b_0 < a_0 \text{ or } b_1 > a_1) \\ \leq \mathbb{P}_D(b_0 < a_0) + \mathbb{P}_D(b_1 > a_1)$$

Note that $b_0 < a_0$ if and only if all data in S are not between a_0 and a^* , namely,



$$\mathbb{P}_D(b_0 < a_0) = \mathbb{P}_D(\text{for any } x \in S|_x, x \notin (a_0, a^*)) \\ \leq (1 - \epsilon)^m$$

Similarly, $P_{\mathcal{D}}(b_i > a_i) \leq (1 - \epsilon)^m$
together.

$$P_{\mathcal{D}}(h_0(h_i) > \epsilon) \leq 2(1 - \epsilon)^m \leq 2e^{-\epsilon m}$$

setting $2e^{-\epsilon m} \leq \delta$

yields

$$m \geq \frac{1}{\epsilon} \log\left(\frac{2}{\delta}\right)$$

that is, $\mathcal{H}$ is PAC-learnable with sample complexity

$$m_{\mathcal{H}} \leq \left\lceil \frac{1}{\epsilon} \log\left(\frac{2}{\delta}\right) \right\rceil$$

shattering

definition (restriction of $\mathcal{H}$ to C)

Let $\mathcal{H}$ be a class of functions from X to $\{0, 1\}$, and let $C = \{c_1, \dots, c_m\} \subseteq X$. The restriction of $\mathcal{H}$ to C is the set of function from C to $\{0, 1\}$ that can be derived from $\mathcal{H}$. That is

$$\mathcal{H}_C = \{h(c_1), \dots, h(c_m)\} : h \in \mathcal{H}\}$$

definition (shattering)

A hypothesis class $\mathcal{H}$ **shatters** a finite set $C \subseteq X$ if the restriction of $\mathcal{H}$ to C is the set of all functions from C to $\{0, 1\}$. That is $|\mathcal{H}_C| = 2^{|C|}$

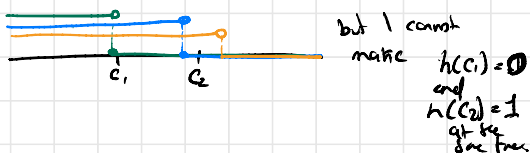
example of shattering

Let $\mathcal{H} = \{h_a : a \in \mathbb{R}\}$ where $h_a(x) = \mathbb{1}[x < a](x)$

Let $C = \{c_1\}$. Does $\mathcal{H}$ shatter C ? **Yes**



what abt. $C = \{c_1, c_2\}$ ($c_1 < c_2$) **No**



if $\mathcal{H}$ shatters some set C of size $2m$,

- a corollary of NFI: Let $\mathcal{H}$ be a hypothesis class of functions from X to $\{0, 1\}$. Let m be a training set size. Assume that there exists a set $C \subseteq X$ of size $2m$ that is shattered by $\mathcal{H}$.

Then for any learning algorithm A , there exists a distribution $\mathcal{D}$ over $X \times \{0, 1\}$ and a predictor $h \in \mathcal{H}$ such that $L_{\mathcal{D}}(h) = 0$ but with probability at least $\frac{1}{2}$ over the choice of $S \sim \mathcal{D}^m$ we have that $L_S(A(S)) \geq \frac{1}{8}$.

VC (Vapnik-Chervonnkis) dimension

• definition (VC-dimension): the VC-dimension of a hypothesis class $\mathcal{H}$, denoted by $\text{VCdim}(\mathcal{H})$, is the maximal size of a set $C \subset \mathcal{X}$ that can be shattered by $\mathcal{H}$.

• if $\mathcal{H}$ can shatter sets of arbitrary large size we say $\mathcal{H}$ has infinite VC dimension.

• if $\text{VCdim}(\mathcal{H}) = \infty$, then $\mathcal{H}$ is not PAC learnable

• to show that $\text{VCdim}(\mathcal{H}) = d$ we need to show

1. There exists a set C of size d that is shattered by $\mathcal{H}$.

2. Every set C of size $d+1$ cannot be shattered by $\mathcal{H}$.