Midterm Cheatsheet

Tuesday, October 19, 2021 8:49 PM

Bayes' Rule

- Using Bayes, posterior prob of class ω_j given measurement x is given by:

$$P(\omega_j|x) = \frac{p(x|\omega_j)P(\omega_j)}{p(x)} = \frac{\text{likelihood} \times \text{prior}}{\text{evidence}}$$

Where $p(x) = \sum_{j=1}^{2} p(x|\omega_j)P(\omega_j)$ ★scale factor

Decide ω_1 if $P(\omega_1|x) > P(\omega_2|x)$; otherwise ω_2

Decide ω_1 if $p(x|\omega_1)P(\omega_1) > p(x|\omega_2)P(\omega_2)$;...$\omega_2$

Likelihood Ratio Test

$$\frac{p(x|\omega_1)}{p(x|\omega_2)} > \frac{\lambda_{12}-\lambda_{22}}{\lambda_{21}-\lambda_{11}} \frac{P(\omega_2)}{P(\omega_1)}$$

Let $\theta_\lambda = \frac{\lambda_{12}-\lambda_{22}}{\lambda_{21}-\lambda_{11}} \frac{P(\omega_2)}{P(\omega_1)}$, then

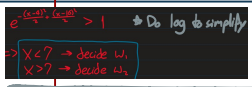
Decide ω_1 if $\frac{p(x|\omega_1)}{p(x|\omega_2)} > \theta_\lambda$

For zero-one loss: $\lambda = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$, $\theta_\lambda = \frac{P(\omega_2)}{P(\omega_1)} = \theta_a$

Penalize more on misclassifying ω_2 as ω_1,

e.g. $\lambda = \begin{bmatrix} 0 & 5 \\ 1 & 0 \end{bmatrix}$ (λ_{12}, λ_{21}), $\theta_\lambda = \frac{5P(\omega_2)}{P(\omega_1)} = \theta_b$

Gaussian: $f(x) = \frac{1}{\sigma\sqrt{2\pi}} \cdot e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$



Minimum Risk Decision

Minimum Risk Decision: choose the action that minimize the conditional risk. $(R(\alpha_i|x) = 1 - P(\omega_i|x))$

$$\min_{\text{risk}} R(\alpha_i|x) \equiv \max_{\text{prob}} P(\omega_i|x)$$

Decide ω_i if $P(\omega_i|x) > P(\omega_j|x)$ for $j \neq i$.

★Minimize prob error, minimize error rate, minimize risk, etc...

Logistic Regression

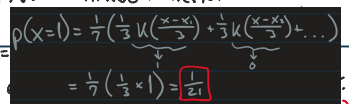
Consider problem of 2-class classification. Using discriminative approach, we tend to maximize a likelihood function defined thru the posterior.

$$p(C_1|\phi) = y(\phi) = \sigma(w^T\phi) \quad \text{★}\sigma = \text{sigmoid}$$

$$p(C_1|\phi) = 1 - p(C_1|\phi) \quad \sigma(x) = \frac{1}{1+e^{-x}}$$

MLE for Logistic Regression

for dataset $\{\phi_n, t_n\}$, where $t_n \in \{0,1\}$, $\phi_n = \phi(x_n)$, $n = 1, \ldots, N$. Likelihood Function:



where: $t =$

Get cross

$E(w) = -\ln p(t|w) = -\sum_{n=1}^{N} \{t_n \ln y_n + (1-t_n)\ln(1-y_n)\}$

Taking the gradient (using derivative of sigmoid):

$\nabla E(w) = \sum_{n=1}^{N} (y_n - t_n)\phi_n$ ★t_n=labels ★ϕ_n=inputs

Error Function w/ penalty:

Sigmoid Derivative

$\frac{\partial \sigma(x)}{}$

Perceptron

1) Loop over training data: $x_1, x_2, \ldots x_N$

2) In τ step, if x_n is correctly classified, then $w^{(\tau+1)} = w^{(\tau)}$, otherwise:

$$w^{(\tau+1)} = \begin{cases} w^{(\tau)} + x_n & \text{if } x_n \text{ is in + class } (+1) \\ w^{(\tau)} - x_n & \text{if } x_n \text{ is in - class } (-1) \end{cases}$$

3) Convergence theorem: regardless of initial choice of weights, if 2 classes are linearly separable, there exists w such that:

$$\begin{cases} w^T x > 0 & \text{if } x_n \text{ in + class} \\ w^T x \leq 0 & \text{if } x_n \text{ in - class} \end{cases}$$

Note that: ← ground truth (+1 or -1)

$$t_i(w^T x_i) > 0 \iff x_i \text{ correctly classified}$$

weights ↑ ↑ inputs

Parzen Window vs. KNN

Density Estimation: $p_n(x) = \frac{k_n/n}{V_n}$

Parzen Window: Fix window size (e.g. $V_n = 1/\sqrt{n}$)★

$k_n = k_n(x)$ is a function of x:

$$p_n(x) = \frac{k_n/n}{V_n} = \frac{k_n(x)}{V_1\sqrt{n}}$$

KNN: Fix # of samples inside window (e.g. $k_n = \sqrt{n}$)

★ $V_n = V_n(x)$ is a function of x: ★k_n = #points in R_n ★V_n = volume of R_n ★n = total n

$$p_n(x) = \frac{k_n/n}{V_n} = \frac{1}{V_n(x)\sqrt{n}}$$

★$p_n(x) = n^{th}$ estimate for $p(x)$ = density estimate

Parzen Window

Hypercube of volume $V_n = h_n^d$

Kernel Function: $k(u) = \begin{cases} 1; & |u_i| \leq 1/2, \ i = 1, \ldots, d \\ 0; & \text{otherwise} \end{cases}$

$k(u)$ = unit hypercube centered @ origin

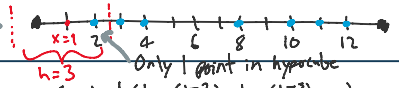
$k\left(\frac{x-x_i}{h_n}\right)$ will be 1 if x_i lies in $c(x, \frac{h_n}{2})$, 0 otherwise

Points in hypercube: $k_n = \sum_{i=1}^{n} k\left(\frac{x-x_i}{h_n}\right)$ ★ if ≥ 0.5, $k(u) = 1$

Plug in $p_n(x)$ w/ k_n:

$$p_n(x) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{V_n} k\left(\frac{x-x_i}{h_n}\right)$$

Ex) 7 samples $D = \{2,3,4,8,10,11,12\}$, $h = 3$, $x = 1$



$$p(x=1) = \frac{1}{7}\left(\frac{1}{3}K\left(\frac{1-2}{3}\right) + \frac{1}{3}K\left(\frac{1-3}{3}\right) + \ldots\right) = \frac{1}{7}\left(\frac{1}{3} \times 1\right) = V_{21}$$

KNN Classification

$p(C_j|x) = k_j/k$ ★Choose class w/ highest k

★Higher k = smoother, higher bias

MLE vs. Bayesian Estimation

MLE: Trying to find value for params θ that maximize the likelihood. Not looking for a distribution.

Bayesian: Fully calculate posterior to get distribution for the params. Bayesian is more complex but

$\nabla E(w) = \sum_{n=1}^{N} (y_n - t_n)\phi_n$ ★t_n=labels ★ϕ_n=inputs

Error Function w/ penalty: $\frac{1}{N}\sum_{n=1}^{N} -\{t_n \ln y(x_n) + (1-t_n)\ln(1-y(x_n))\} + \frac{\lambda}{2}\|w\|^2$

Sigmoid Derivative

$\frac{\partial \sigma(x)}{\partial x} = \sigma(x)(1-\sigma(x))$

$\nabla = \sum_{i=1}^{N} -\frac{\partial}{\partial w}(y_i \log(\sigma(w^T x_i)) + (1-y_i)\log(1-\sigma(w^T x_i))) + \frac{\partial}{\partial w}\frac{\lambda}{2}\|w\|^2$

$= -\left[\frac{y_i}{\sigma(w^T x_i)} \frac{\partial \sigma(w^T x_i)}{\partial w} + \frac{1-y_i}{1-\sigma(w^T x_i)}\left(-\frac{\partial \sigma(w^T x_i)}{\partial w}\right)\right] + \lambda w$

$= -\left[\frac{y_i}{\sigma(w^T x_i)} - \frac{1-y_i}{1-\sigma(w^T x_i)}\right] \frac{\partial \sigma(w^T x_i)}{\partial w} + \lambda w$

$= -\left[\frac{y_i}{\sigma(w^T x_i)} - \frac{1-y_i}{1-\sigma(w^T x_i)}\right] \sigma(w^T x_i)(1-\sigma(w^T x_i)) x_i + \lambda w$

$= -[y_i(1-\sigma(w^T x_i)) - (1-y_i)(\sigma(w^T x_i))] x_i + \lambda w$

$= [y_i - \sigma(w^T x_i)] x_i + \lambda w$

$= (\sigma(w^T x_i) - y_i) x_i + \lambda w$

$\Rightarrow \sum_{i=1}^{N} [\sigma(w^T x_i) - y_i] x_i + \lambda w$

Discriminant: Directly assign x to a specific class.

Probabilistic Discriminative: Directly model posterior $p(C_k|x)$

Probabilistic Generative: Model class-conditional * prior.

Likelihood * Prior

Consider $t = w^T \phi(x) + \epsilon$, * $\epsilon \sim N(0, \beta^{-1})$. Assume noise precision β is known constant. Likelihood equals:

$p(t|w) = \prod_{n=1}^{N} N(t_n | w^T \phi(x_n), \beta^{-1})$

Define prior distribution on w:

$p(w) = N(w | m_0, S_0)$

Consider simplified Gaussian prior:

assuming $p(w|\alpha) = N(w|0, \alpha^{-1} I)$ ★ only unknown $= w$

MLE: Trying to find value for params θ that maximize the likelihood. Not looking for a distribution.

Bayesian: Fully calculate posterior to get distribution for the params. Bayesian is more complex but more accurate since it uses more information.

FLD vs. PCA

FLD: Focusing on maximizing separability among known classes. It is supervised * reduces to line (1D).

PCA: Focusing on reducing dimensions to find feature w/ highest variation. Unsupervised (no labels).

3 Types of Gradient Descent

Batch: Uses all data each iteration. Accurate * unbiased, but slow * easy to get trapped in local modes.

Mini-Batch: Use small batches each iteration. Faster * adds noise to prevent getting trapped. Noise may cause learning to go back & forth.

Stochastic: 1 piece of data each iteration. Most efficient but can be slow to converge.

Maximum Likelihood Estimator

$f(x|\theta) = \prod_{i=1}^{n} 2\theta x_i e^{-\theta x_i^2}$

$\ln(f(x|\theta)) = \sum_{i=1}^{n} \ln[2\theta x_i e^{-\theta x_i^2}]$

$LL(\theta) = \sum_{i=1}^{n} (\ln(2\theta x_i) * \ln[e^{-\theta x_i^2}])$

$LL(\theta) = \sum_{i=1}^{n} [\ln(2\theta x_i) - \theta x_i^2]$

$\nabla_\theta LL(\theta) = \sum_{i=1}^{n} \left(\frac{2x_i}{2\theta x_i} - x_i^2\right) = \sum_{i=1}^{n} \left(\frac{1}{\theta} - x_i^2\right)$

$\nabla_\theta LL(\theta) = \frac{n}{\theta} - \sum_{i=1}^{n} x_i^2$

Set $\nabla_\theta LL(\theta) = 0 \rightarrow \frac{n}{\theta} - \sum_{i=1}^{n} x_i^2 = 0$

$\frac{1}{\theta} = \frac{1}{n}\sum_{i=1}^{n} x_i^2 \rightarrow \theta = \frac{n}{\sum_{i=1}^{n} x_i^2}$ MLE

Likelihood for Uniform Distribution

$\prod_{i=1}^{n} f(x_i; a, b) = \prod_{i=1}^{n} \frac{1}{(b-a)} = \frac{1}{(b-a)^n}$

$\log \prod_{i=1}^{n} f(x_i; a, b) = \log((b-a)^{-n}) = -n \log(b-a)$

★Log Likelihood ↑

Derivative wrt a: $\frac{n}{(b-a)}$

Derivative wrt b: $\frac{-n}{(b-a)}$

MLE: Largest possible a, smallest b

$a = \min\{x\}$

$b = \max\{x\}$