Bayesian Classification: Parametric and nonparametric methods

Jayanta Mukhopadhyay
Dept. of Computer Science and Engg.

Books

- Chapters 4 and 8 of "Introduction to Machine Learning" by Ethem Alpaydin.
- Chapter 8 of "Machine Learning" by Tom M. Mitchell

Parametric methods

- Probability density functions of known form and described by a set of parameters.
  - E.g. Gaussian distribution.
  - $P(x|\Theta)$, Θ is a set of parameters.
  - Estimation of Θ is enough to provide probability measures.
- Subsequently use them for Bayesian inference.

Maximum likelihood estimation

- Data: $\mathrm{X}=\{x^t\}$, t=1,2,..,N
  - x^t : an independent and identically distributed (iid) sample.
- Likelihood: $l(\Theta|\mathrm{X})=\mathrm{P}(\mathrm{X}|\,\Theta) = \prod_{t=1}^{N} P(x^t|\theta)$
- MLE of Θ : $\theta^* = \underset{\theta}{\operatorname{argmax}}\, l(\theta|X)$
- Log-likelihood: log($l(\Theta|\mathrm{X}))=\mathrm{L}(\Theta|\mathrm{X})=\sum_{t=1}^{N} \log P(x^t|\theta)$

Bernoulli distribution

- Two possible outcomes of X:
  - 1 with p, 0 with (1-p).
  - $P(x)=p^x (1-p)^{(1-x)}$, x=0, and 1.
  - $E(X)=p$, $var(X)=p(1-p)$
- $L(p|X)= \log \prod_{t=1}^{N} p^{x^t}(1-p)^{1-x^t}$

$$= \log p \left(\sum_{t=1}^{N} x^t\right) + \log(1-p)\left(N - \sum_{t=1}^{N} x^t\right)$$

Bernoulli distribution

- $\mathrm{L(p|X)} = \log p \left(\sum_{t=1}^{N} x^t \right) + \log(1-p) \left(N - \sum_{t=1}^{N} x^t \right)$

$$\frac{\partial L(p|X)}{\partial p} = 0 \;\Rightarrow\; \hat{p} = \frac{\sum_{t=1}^{N} x^t}{N}$$

Note:

$\mathrm{E(X)} = p$ and the estimate of p is the sample average.

Multinomial density function

- Generalization of Bernoulli process.
  - One of K mutually exclusive states occurs at every trial.
  - p_i prob. of occurrence of ith state.
  - x_i i-th indicator variable; 1 if it occurs else 0.
  - P(x)= $\prod_{i=1}^{K} p_i^{x_i}$
  - Data: $\{\mathbf{x}^t \mid \mathbf{x}^t = (x_1^t, x_2^t, \ldots, x_K^t)\}$, t=1,2,…,N
  - MLE of p_i: $\hat{p_i} = \frac{\sum_{t=1}^{N} x_i^t}{N}$

Gaussian (Normal) density function

- Univariate distribution:

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}, -\propto < x < \propto$$

$$E(x) = \mu, \qquad var(x) = \sigma^2$$

- MLE of parameters: $\hat{\mu} = m = \frac{1}{N}\sum_{t=1}^{N} x^t$

$$\widehat{\sigma^2} = s^2 = \frac{\sum_{t=1}^{N}(x^t - m)^2}{N}$$

Bias, MSE, and variance of estimators

- Let $d(X)=d$ be estimator of a parameter Θ.
  - As X randomly vary, d is also a random variable, with E(d) and var(d).
  - Bias: $b_\Theta(d)=E(d\,(X))-\Theta$
    - Unbiased estimator: $b_\Theta(d)=0$
  - MSE of estimator: $E((d(X)-\Theta)^2)$
- Sample mean m: $$m=\frac{1}{N}\sum_{t=1}^{N}x^t$$

$$E(m)=\frac{1}{N}\sum_{t=1}^{N}E(x^t)=\frac{N\mu}{N}=\mu$$

Sample mean

- Sample mean m: $$m = \frac{1}{N}\sum_{t=1}^{N} x^t$$

$$E(m) = \frac{1}{N}\sum_{t=1}^{N} E(x^t) = \frac{N\mu}{N} = \mu$$

Sample mean is an unbiased estimator of μ.

$$var(m) = \frac{1}{N^2}\sum_{t=1}^{N} var(x^t) = \frac{N\sigma^2}{N^2} = \frac{\sigma^2}{N}$$

As N → ∝, var(m) → 0. Hence, sample mean is a consistent estimator of μ.

Sample variance

$$\widehat{\sigma^2} = s^2 = \frac{\sum_{t=1}^{N}(x^t - m)^2}{N} = \frac{\sum_t (x^t)^2 - Nm^2}{N}$$

$$E(s^2) = \frac{\sum_t E((x^t)^2) - NE(m^2)}{N}$$

$$var(X) = E(X^2) - (E(X))^2 \Rightarrow E(X^2) = (E(X))^2 + var(X)$$

$$E((x^t)^2) = \mu^2 + \sigma^2 \qquad E(m^2) = \mu^2 + \frac{\sigma^2}{N}$$

$$E(s^2) = \frac{N(\mu^2 + \sigma^2) - N\left(\mu^2 + \frac{\sigma^2}{N}\right)}{N} = \frac{N-1}{N}\sigma^2 \neq \sigma^2$$

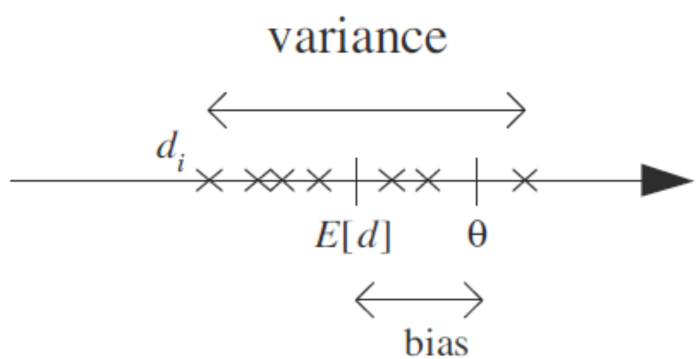
- Unbiased estimator: $\frac{N}{N-1}s^2 = \frac{\sum_{t=1}^{N}(x^t - m)^2}{N-1}$

s^2: Asymptotically unbiased.

MSE of an estimator d

- MSE: $E((d(X) - \Theta)^2)$

$$
\begin{aligned}
&= E((d(X) - E(d) + E(d) - \Theta)^2) \\
&= \underbrace{E((d(X)-E(d))^2}_{} + \underbrace{E((E(d)-\Theta)^2)}_{} + E(2(d(X)-E(d))\,(E(d)-\Theta)) \\
&= \quad \text{Variance} \quad + \quad \text{bias}^2 \quad + 2\,(E(d)-\Theta)\,\underbrace{E(d(X)-E(d))}_{0} \\
&= \text{Variance of d} + (\text{bias of d})^2
\end{aligned}
$$



Courtesy: “Introduction to Machine” Learning by Ethem Alpaydin (Chapter 4, Fig. 4.1)

The Bayes' estimator

- Estimation of Θ by considering posterior probability $P(\Theta|X)$.

$$P(\theta \mid X) = \frac{P(X \mid \theta)P(\theta)}{P(X)} = \frac{P(X \mid \theta)P(\theta)}{\int P(X \mid \theta)P(\theta)d\theta}$$

  - Estimation or assumption of prior $P(\Theta)$ required.
  - Difficult to compute the denominator.
  - If assumed a narrow posterior distribution $P(\Theta|X)$. use Maximum Posterior (MAP) estimate of Θ.

$$\theta_{MAP} = \arg\max_{\theta} P(\theta \mid X)$$

  - Another estimation method to take the expected value.

$$\theta_{Bayes} = \int \theta P(\theta \mid X) d\theta$$

    - The best estimate of a random variable is its mean.

Bayes' estimator of mean of a normal distribution

- $x^t \sim N(\mu, \sigma^2)$ and $\Theta \sim N(\mu_0, \sigma_0^2)$

$$P(X\mid\theta) = \frac{1}{(2\pi)^{\frac{N}{2}}\sigma^N} e^{-\frac{\sum_t (x^t-\mu)^2}{2\sigma^2}} \qquad p(\theta) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(\theta-\mu_0)^2}{2\sigma_0^2}}$$

N: Number of samples

- It can be shown that $P(\Theta|X)$ is normal.
  - Mean is given by weighted mean of μ and μ_0.

$$E(\theta\mid X) = \frac{\frac{N}{\sigma^2}}{\frac{N}{\sigma^2}+\frac{1}{\sigma_0^2}}\mu + \frac{\frac{1}{\sigma_0^2}}{\frac{N}{\sigma^2}+\frac{1}{\sigma_0^2}}\mu_0$$

Parametric classification

- Compute posterior for all classes:

$$P(C_i \mid x) = \frac{P(x \mid C_i)P(C_i)}{P(x)} = \frac{P(x \mid C_i)P(C_i)}{\sum_{i=1}^{K} P(x \mid C_i)P(C_i)}$$

- Assign the class with maximum posterior.
  - May ignore $P(x)$ or the denominator.
- Discriminant functions:
  - $g_i(x)=P(x|C_i)P(C_i)$
  - $g_i(x)=log\,P(x|C_i) + \log P(C_i)$

Parametric classification

- Discriminant functions:
  - $g_i(x)=P(x|C_i)P(C_i)$
  - $g_i(x)=log\ P(x|C_i) + \log P(C_i)$

There exists pure discriminant function based approach not requiring parameter estimation.

- Assuming $P(x|C_i)$ is Gaussian: N(μ_i, σ_i^2)

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

Estimate μ_i and σ_i.

  - The first term can be dropped. The same for all classes.
  - If $P(C_i)$ is the same for all classes, the last term can also be dropped.
  - If $\sigma_i=\sigma$ for all classes, class with closest mean to x is assigned.
    - Nearest neighbor classification.

Multivariate representation

- $\boldsymbol{x}^t \in R^d$, t=1,2,..,N
- A data matrix:
  - Each row a data sample.

$$\mathbf{X} = \begin{bmatrix} X_1^{\ 1} & X_2^{\ 1} & \dots & X_d^{\ 1} \\ X_1^{\ 2} & X_2^{\ 2} & \dots & X_d^{\ 2} \\ | & | & | & | \\ X_1^{\ N} & X_2^{\ N} & \dots & X_d^{\ N} \end{bmatrix}$$

- Mean vector: $\boldsymbol{\mu}=[\mu_1\ \mu_2\ \dots\ \mu_d]$
  - Mean of each column.
- Covariance matrix: $\boldsymbol{\Sigma}=[\sigma_{ij}]$
  - $\sigma_{ij}=$ COV(X_i, X_j)
  - $\boldsymbol{\Sigma}= \mathrm{E}((\mathbf{X}-\boldsymbol{\mu})^T(\mathbf{X}-\boldsymbol{\mu}))=\mathrm{E}(\mathbf{X}^T\mathbf{X})-\boldsymbol{\mu}^T\boldsymbol{\mu}$
  - Correlation matrix: $[\rho_{ij}=\sigma_{ij}/\sigma_i\,\sigma_j]$

Multivariate normal distribution

$$P(\boldsymbol{x}) = \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \Sigma^{-1} (\boldsymbol{x}-\boldsymbol{\mu})}$$

- Mahalanabis distance: $(\boldsymbol{x} - \boldsymbol{\mu})^{\mathrm{T}} \Sigma^{-1} (\boldsymbol{x} - \boldsymbol{\mu})$
- $\mathrm{P}(\boldsymbol{x}|C_i) \sim \mathrm{N}(\boldsymbol{\mu}_i, \Sigma_i)$

$$g_i(\boldsymbol{x}) = -\frac{d}{2}\log 2\pi - \frac{1}{2}\log|\boldsymbol{\Sigma}_i| - \frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}_i^{-1} (\boldsymbol{x}-\boldsymbol{\mu}) + \log P(C_i)$$

- Estimate parameters by computing mean vector and cov. matrix from the samples of each class.
- Quadratic discriminant function
  - If the cov. matrices are the same, discriminant becomes linear.

Multivariate normal distribution

- If all the features of x are independent, cov. matrix is diagonal → only σ_i's of classes are nonzero.

$$g_i(\boldsymbol{x}) = -\frac{1}{2}\sum_{j=1}^{d}\left(\frac{x_j - \mu_{ij}}{\sigma_{ij}}\right)^2 + \log P(C_i)$$

Multivariate discrete features

- Discrete attributes taking one of n values.
- Let $\boldsymbol{x}=(x_1,x_2,\ldots,x_d)$, each x_j is a Bernoulli.
  - $p_{ij}=P(x_j=1|C_i)$
- Class likelihood:

$$P(\boldsymbol{x}|C_i) = \prod_{j=1}^{d} p_{ij}{}^{x_j}\left(1-p_{ij}\right)^{(1-x_j)}$$

- Discriminant function:

$$g_i(\boldsymbol{x}) = \sum_{j=1}^{d}\left(x_j \log p_{ij} + \left(1-x_j\right)\log\left(1-p_{ij}\right)\right) + \log P(C_i)$$

  - A linear function.

An application

- Document characterization (say news items).
  - Represent each document by a vector of bag of words.
    - x_j denotes whether j th word in the "BOW" occurs or not.
    - Estimate p_{ij} from training data set, and
    - Obtain discriminant functions for each class of documents.

Generalization to multinomial cases

- Each x_j can take one of n_j discrete values. $(v_1, v_2, \ldots v_{nj})$
- Define a dummy variable z_{jk}: $$z_{jk} = \begin{cases} 1 & if\ x_j = v_k \\ 0 & Otherwise \end{cases}$$
- Parameters: $$p_{ijk} = P(z_{jk} = 1|C_i)$$
- Class likelihood: $$P(\boldsymbol{x}|C_i) = \prod_{j=1}^{d} \prod_{k=1}^{n_j} p_{ijk}^{z_{jk}}$$
- Discriminant function: $$\sum_j \sum_k z_{jk} \log p_{ijk} + \log P(C_i)$$

Summary of parametric methods

- Estimation of parameters used for computing class likelihood and posterior.
- Discriminant functions are obtained from the analytical forms of posterior.
- Mean and s.d. of classes in feature space form the parameters of normal distribution.
- Probabilities of occurrences of Bernoulli features form the parameters.
- The same analysis could be extended for multivariate class distribution.

Nonparametric approaches

- Only assumption: similar inputs have similar outputs.
  - No assumption on form of density functions.
- Estimate probability density locally.
  - Parametric approach: estimation of density function using all samples together.
- Instance based or memory based learning.
  - Need to store samples in memory for look up.
  - Computation intensive: at least $O(N)$: $N=|X|$
    - Parametric approach: $O(d)$ or $O(d^2)$: d: dimension
  - Lazy learning compared to eager parametric models.

Univariate nonparametric density estimation

- $\{x^t\}, t=1,2,..N$
  - Estimated cumulative prob.:
    - $F(x)=\#(x^t<x)/N$
  - Estimated prob. density :
    - $P(x)= [(\#(x^t<x+h) - \#(x^t<x))/N]/h$
  - Naive estimator:
    - $P(x)= [(\#(x^t<x+h/2) - \#(x^t<x-h/2))/N]/h$
  - Using histogram of bin-width h
    - $P(x)=[\#(x^t$ in the same bin containing $x)/N]/h$
      - Once histogram is computed, training samples not required for estimating P(x).

Kernel estimator

- Kernel function: A function of distance used to determine the weight of each example.
- $\{x^t\}, t=1,2,..N$
  - $w^t = K(d(x, x^t))$
- Kernel estimator (Parzen window):

$$P(x) = \frac{1}{Nh}\sum_{t=1}^{N} K\left(\frac{x - x^t}{h}\right)$$

$$K(u) = \begin{cases} 1 & |u| < \frac{1}{2} \\ 0 & \text{Otherwise} \end{cases}$$

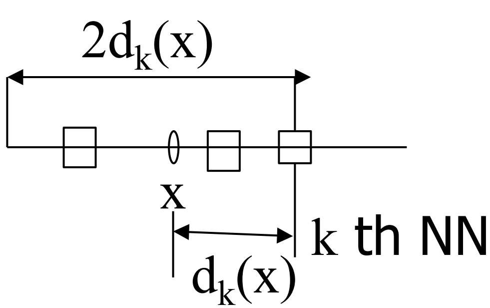
$$K(u) = \frac{1}{\sqrt{2\pi}} e^{-\frac{u^2}{2}}$$

Smooth estimator

k-NN estimator

- k-nearest neighbor (k-NN) estimator
  - Distance: $|a-b|$ (Univariate)
  - $d_i(x)$= distance of the i th NN from x.
  - k-NN density estimate:
    - $k/(N(2d_k(x)))$
- Adaptive kernel estimator:

$$P(x) = \frac{1}{N2d_k(x)}\sum_{t=1}^{N} K\left(\frac{x - x^t}{2d_k(x)}\right)$$



Multivariate density estimation

- $\{\mathbf{x}^t \mid \mathbf{x}^t \text{ in } R^d\}$,t=1,2,..N.

$$P(\mathbf{x}) = \frac{1}{Nh^d}\sum_{t=1}^{N} K\left(\frac{\mathbf{x}-\mathbf{x}^t}{h}\right) \qquad \int_{\mathbf{x}\in R^d} K(\mathbf{x})d\mathbf{x} = 1$$

- Gaussian kernel:

$$K(\mathbf{u}) = \frac{1}{(2\pi)^{\frac{d}{2}}} e^{-\frac{\|\mathbf{u}\|^2}{2}}$$

- Using S: cov. matrix of samples

$$K(\mathbf{u}) = \frac{1}{(2\pi)^{\frac{d}{2}}|S|^{\frac{1}{2}}} e^{-\frac{\mathbf{u}^T S^{-1}\mathbf{u}}{2}}$$

- For discrete input: Hamming distance may be used.

Nonparametric Classification

- $\{\mathbf{x}^t, \mathbf{r}^t \mid \mathbf{x}^t \text{ in } R^d\}$, t=1,2,..N.
  - r_i^t=1 if $\mathbf{x}^t$ belongs to C_i else 0. i=1,2,..,M
    - M: Number of classes
    - N_i=# of samples in C_i
    - $P(C_i) = N_i/N$

$$P(\mathbf{x}|C_i) = \frac{1}{N_i h^d} \sum_{t=1}^{N} K\left(\frac{\mathbf{x}-\mathbf{x}^t}{h}\right) r_i^t$$

    - $g_i(\mathbf{x}) = P(\mathbf{x}|C_i)P(C_i) = \frac{1}{Nh^d} \sum_{t=1}^{N} K\left(\frac{\mathbf{x}-\mathbf{x}^t}{h}\right) r_i^t$

Common factor to be ignored.

volume of hypersphere radius $d_k(\mathbf{x})$.

- k-nn estimator: $k_i/(N_i V^{(k)}(\mathbf{x}))$
  - $P(C_i|\mathbf{x}) = P(\mathbf{x}|C_i)P(C_i)/P(\mathbf{x}) = k_i/k$

Distance to kth nn.

Instance based learning

- Instance based learning
  - Training: Store Instances
  - Testing / Query processing:
    - Retrieve a set of similar related instances
    - Classify / Regress using them
- Significant advantage over complex target function
  - Instead of computing a global function, compute locally.
    - K-NN, Locally weighted regression
- Lazy learning
  - delayed until a new instance get classified
    - Instead of computing once for all, compute incrementally

k-NN Regression

- Target function: $f: R^d \rightarrow R$
- Training examples: $(\mathbf{x}^t, f(\mathbf{x}^t))$, $t=1,2,..N$
- i th Neighbor of $\mathbf{x}$: $\mathbf{x}_i$

$$\hat{f}(\mathbf{x}) = \frac{\sum_{i=1}^{k} f(\mathbf{x}_i)}{k}$$

- For weighted regression use weight
  - inversely proportional to square of the distance $d(\mathbf{x},\mathbf{x}_i)$
  - proportional to a kernel function

Locally weighted regression

- Target function linear on attribute variables.
  - $f(x)=w_0+w_1x_1+w_2x_2+ .. +w_dx_d$
- MSE in 3 scenarios:
  - Over unweighted k-NNs
  - Over all training examples with weights proportional to $K(\mathbf{x},\mathbf{x}^t)$.
  - Over k-NNs with weights proportional to $K(\mathbf{x},\mathbf{x}^t)$.

$$E = \frac{1}{2}\sum_{\mathbf{x}^t \in S} K(\mathbf{x}^t, \mathbf{x})\left(f(\mathbf{x}^t) - \hat{f}(\mathbf{x}^t)\right)^2$$

S: k-NNs or all instances

- Weight update rules for minimization

$$\Delta w_i = \eta \sum_{\mathbf{x}^t \in S} K(\mathbf{x}^t, \mathbf{x})(f(\mathbf{x}^t) - \hat{f}(\mathbf{x}^t))x_i^t$$

Summary of nonparametric approaches

- Nonparametric estimation using kernel function with a local support at **x** and bounded integral value of 1.
- Requires high computation and storage.
- Simple to implement and provides good performance on the average.
- Can be modeled as instance based learning
- Possible to use in regression

Thank you!