Bayesian learning

Usman Roshan

CS 675

Machine Learning

Supervised learning for two classes

- We are given *n* training samples (x_i, y_i) for $i=1..n$ drawn i.i.d from a probability distribution $P(x,y)$.
- Each x_i is a d-dimensional vector ($x_i \in R^d$) and y_i is +1 or -1
- Our problem is to learn a function $f(x)$ for predicting the labels of test samples x_i' (in R^d) for $i=1..n'$ also drawn i.i.d from $P(x,y)$

Classification: Bayesian learning

- Bayes rule: $$P(M \mid x) = \frac{P(x \mid M)P(M)}{P(x)} = \frac{P(x \mid M)P(M)}{\sum_M P(x \mid M)P(M)}$$
- To classify a given datapoint *x* we select the model (class) M_i with the highest $P(M_i|x)$
- The denominator is a normalizing term and does not affect the classification of a given datapoint. Therefore

$$P(M \mid x) \propto P(x \mid M)P(M)$$

- *P(x|M)* is called the likelihood and *P(M)* is the prior probability. To classify a given datapoint *x* we need to know the likelihood and the prior.
- If priors *P(M)* are uniform (the same) then finding the model that maximizes *P(M|D)* is the same as finding *M* that maximizes the likelihood *P(D|M)*.

Maximum likelihood

- We can classify by simply selecting the model *M* that has the highest *P(M|D)* where *D*=data, *M*=model. Thus classification can also be framed as the problem of finding *M* that maximizes *P(M|D)*

- By Bayes rule:

$$P(M \mid D) = \frac{P(D \mid M)P(M)}{P(D)} = \frac{P(D \mid M)P(M)}{\sum_{M} P(D \mid M)P(M)}$$

Maximum likelihood

- Suppose we have *k* models to consider and each has the same probability. In other words we have a uniform prior distribution *P(M)=1/k*. Then

$$P(M \mid D) = P(D \mid M)\, {}^{1}\!/_{k} \Big/ \sum_{M} P(D \mid M) P(M) \propto P(D \mid M)$$

- In this case we can solve the classification problem by finding the model that maximizes *P(D|M)*. This is called the maximum likelihood optimization criterion.

Maximum likelihood

- Suppose we have *n* i.i.d. samples (*xi,yi*) drawn from *M*. The likelihood *P(D|M)* is

$$P(D \mid M) = P((x_1, y_1), \ldots, (x_n, y_n) \mid M) = P(x_1, y_1 \mid M) \ldots P(x_n, y_n \mid M)$$

$$= \prod_{i=1}^{n} P(x_i, y_i \mid M) = \prod_{i=1}^{n} P(y_i \mid x_i, M) P(x_i)$$

- Consequently the log likelihood is

$$-\log P(D \mid M) = -\sum_{i=1}^{n} \log P(y_i \mid x_i, M) - \sum_{i=1}^{n} P(x_i)$$

Maximum likelihood and empirical risk

- Maximizing the likelihood *P(D|M)* is the same as maximizing *log(P(D|M))* which is the same as minimizing *-log(P(D|M))*
- Set the loss function to

$$c(x_i, y_i, f(x_i)) = -\log(P(y_i \mid x_i, f))$$

- Now minimizing the empirical risk is the same as maximizing the likelihood (return to this later again)

Maximum likelihood example

- Consider a set of coin tosses produced by a coin with $P(H)=p$ ($P(T)=1-p$)
- We want to determine the probability $P(H)$ of the coin that produces k heads and $n-k$ tails?
- We are given some tosses (training data): HTHHHTHHHTHTH.
- Solution:
  - Form the log likelihood
  - Differentiate w.r.t. p
  - Set to the derivative to 0 and solve for p

Maximum likelihood example

- Likelihood is probability of data given model
- Data are the set of coin tosses and model is given by one parameter p
- $P(data|p) = p^k(1-p)^{n-k}$
- $$Log\left(P(data|p)\right) = \log\left(p^k\right) + \log(1-p)^{n-k}$$
  $$= k\log(p) + (n-k)\log(1-p)$$
- Take derivative with respect to p, set to 0, and solve for p

Classification by likelihood

- Suppose we have two classes C_1 and C_2.
- Compute the likelihoods $P(D|C_1)$ and $P(D|C_2)$.
- To classify test data D' assign it to class C_1 if $P(D|C_1)$ is greater than $P(D|C_2)$ and C_2 otherwise.

Gaussian models

- Assume that class likelihood is represented by a Gaussian distribution with parameters μ (mean) and σ (standard deviation)

$$P(x \mid C_1) = \frac{1}{\sqrt{2\pi}\sigma_1} e^{-\frac{(x-\mu_1)^2}{2\sigma_1^2}} \qquad P(x \mid C_2) = \frac{1}{\sqrt{2\pi}\sigma_2} e^{-\frac{(x-\mu_2)^2}{2\sigma_2^2}}$$

- We find the model (in other words mean and variance) that maximize the likelihood (or equivalently the log likelihood). Suppose we are given training points $x_1, x_2, \ldots, x_{n1}$ from class C_1. Assuming that each datapoint is drawn independently from C_1 the sample log likelihood is

$$P(x_1, x_2 \ldots, x_{n1} \mid C_1) = P(x_1 \mid C_1) P(x_2 \mid C_1) \ldots P(x_{n1} \mid C_1) = \frac{1}{\sqrt[n1]{2\pi}\sigma_1} e^{-\frac{\sum_{i=1}^{n1}(x_i-\mu_1)^2}{2\sigma_1^2}}$$

Gaussian models

- The log likelihood is given by

$$\log(P(x_1, x_2 \ldots, x_{n1} \mid C_1)) = -\frac{n1}{2}\log(2\pi) - n1\log(\sigma_1) - \frac{\sum_{i=1}^{n1}(x_i - \mu_1)^2}{2\sigma_1^2}$$

- By setting the first derivatives $dP/d\mu_1$ and $dP/d\sigma_1$ to 0. This gives us the maximum likelihood estimate of μ_1 and σ_1 (denoted as m_1 and s_1 respectively)

$$m_1 = \frac{\sum_{i=1}^{n1} x_i}{n_1} \qquad s_1^2 = \frac{\sum_{i=1}^{n1}(x_i - m_1)^2}{n_1}$$

- Similarly we determine m_2 and s_2 for class C_2.

Gaussian models

- After having determined class parameters for C1 and C2 we can classify a given datapoint by evaluating P(x|C1) and P(x|C2) and assigning it to the class with the higher likelihood (or log likelihood).

$$\log(P(x \mid C_1)) = -\frac{1}{2}\log(2\pi) - \log(s_1) - \frac{(x_i - m_1)^2}{2s_1^2}$$

$$\log(P(x \mid C_2)) = -\frac{1}{2}\log(2\pi) - \log(s_2) - \frac{(x_i - m_2)^2}{2s_2^2}$$

- The likelihood can also be used as a loss function and has an equivalent representation in empirical risk minimization (return to this later).

Gaussian classification example

- Consider one dimensional data for two classes (SNP genotypes for case and control subjects).
  - Case (class C_1): 1, 1, 2, 1, 0, 2
  - Control (class C_2): 0, 1, 0, 0, 1, 1
- Under the Gaussian assumption case and control classes are represented by Gaussian distributions with parameters (μ_1, σ_1) and (μ_2, σ_2) respectively. The maximum likelihood estimates of means are

$$m_1 = \frac{\sum_{i=1}^{n1} x_i}{n_1} = \frac{1+1+2+1+0+2}{6} = 7/6 \qquad m_2 = \frac{0+1+0+0+1+1}{6} = 3/6$$

Gaussian classification example

- The estimates of class standard deviations are

$$s_1 = \frac{\sum_{i=1}^{n1}(x_i - m_1)^2}{n_1} = \frac{(1-7/6)^2 + (1-7/6)^2 + (2-7/6)^2 + (1-7/6)^2 + (0-7/6)^2 + (2-7/6)^2}{6} = .47$$

- Similarly s_2=.25
- Which class does x=1 belong to? What about x=0 and x=2?

$$\log(P(x \mid C_1)) = -\frac{1}{2}\log(2\pi) - \log(s_1) - \frac{(x_i - m_1)^2}{2s_1^2}$$

$$\log(P(x \mid C_2)) = -\frac{1}{2}\log(2\pi) - \log(s_2) - \frac{(x_i - m_2)^2}{2s_2^2}$$

- What happens if class variances are equal?

Multivariate Gaussian classification

- Suppose each datapoint is an m-dimensional vector. In the previous example we would have *m* SNP genotypes instead of one. The class likelihood is given by

$$P(x \mid C_1) = \frac{1}{(2\pi)^{d/2} \left|\Sigma_1\right|^{1/2}} e^{-\frac{1}{2}(x-\mu_1)^T \Sigma_1^{-1} (x-\mu_1)}$$

- Where Σ_1 is the class covariance matrix. Σ_1 is of dimensiona *d* x *d*. The (i,j)th entry of Σ_1 is the covariance of the ith and jth variable.

Multivariate Gaussian classification

- The maximum likelihood estimates of η$_1$ and Σ$_1$ are

$$m_1 = \frac{\sum_{i=1}^{n1} x_i}{n_1} \qquad S_1 = \frac{\sum_{i=1}^{n1} (x_i - m_1)(x_i - m_1)^T}{n_1}$$

- The class log likelihoods with estimated parameters (ignoring constant terms) are

$$\log(P(x \mid C_1)) = -\frac{1}{2}\log(|S_1|) - \frac{1}{2}(x - m_1)^T S_1^{-1}(x - m_1)$$

$$\log(P(x \mid C_2)) = -\frac{1}{2}\log(|S_2|) - \frac{1}{2}(x - m_2)^T S_2^{-1}(x - m_2)$$

Multivariate Gaussian classification

- If $S_1=S_2$ then the class log likelihoods with estimated parameters (ignoring constant terms) are

$$\log(P(x \mid C_1)) = -\frac{1}{2}(x-m_1)^T S^{-1}(x-m_1)$$

- Depends on distance to means.

Naïve Bayes algorithm

- If we assume that variables are independent (no interaction between SNPs) then the off-diagonal terms of S are zero and the log likelihood becomes (ignoring constant terms)

$$\log(P(x \mid C_1)) = -\frac{1}{2}\sum_{j=1}^{m}\left(\frac{x_j - m_{1j}}{s_j}\right)^2$$

Nearest means classifier

- If we assume all variances s_j to be equal then (ignoring constant terms) we get

$$\log(P(x \mid C_1)) = -\frac{1}{2s^2}\sum_{j=1}^{m}(x_j - m_{1j})^2$$

Gaussian classification example

- Consider three SNP genotype for case and control subjects.
  - Case (class C_1): (1,2,0), (2,2,0), (2,2,0), (2,1,1), (0,2,1), (2,1,0)
  - Control (class C_2): (0,1,2), (1,1,1), (1,0,2), (1,0,0), (0,0,2), (0,1,0)
- Classify (1,2,1) and (0,0,1) with the nearest means classifier