Machine learning

Probabilistic Discriminative Classifiers

Hamid Beigy

Sharif University of Technology

December 4, 2021

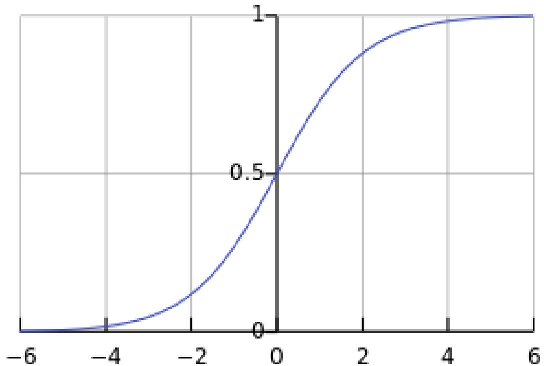
Table of contents

Introduction

- Bayes classifier for two classes C_1 and C_2

$$
\begin{aligned}
p(C_1|X) &= \frac{P(X|C_1)P(C_1)}{P(X)} = \frac{P(X|C_1)P(C_1)}{p(X|C_1)p(C_1) + p(X|C_2)p(C_2)} \\
&= \frac{1}{1 + \frac{p(X|C_2)p(C_2)}{P(X|C_1)P(C_1)}} = \frac{1}{1 + \exp(-a)} = \sigma(a) \\
a &= \ln \frac{P(X|C_1)P(C_1)}{P(X|C_2)P(C_2)}
\end{aligned}
$$

where $\sigma(x)$ refers to sigmoid function.



- Let the class conditional densities be $D-$dimensional Gaussian (for $k = 1, 2$)

$$p(x|C_k) = \mathcal{N}(\mu, \Sigma) = \frac{1}{|\Sigma|^{D/2}(2\pi)^{D/2}} exp\left(-\frac{1}{2}(x-\mu_k)^\top \Sigma^{-1}(x-\mu_k)\right)$$

- Hence a equals to

$$\begin{aligned} a &= \ln \frac{P(X|C_1)P(C_1)}{P(X|C_2)P(C_2)} \\ &= \ln \frac{exp\left(-\frac{1}{2}(x-\mu_1)^\top \Sigma^{-1}(x-\mu_1)\right)}{exp\left(-\frac{1}{2}(x-\mu_2)^\top \Sigma^{-1}(x-\mu_2)\right)} \frac{P(C_1)}{P(C_2)}. \end{aligned}$$

- Hence, we have

$$P(C_1|X) = \sigma\left(W^\top X + w_0\right)$$

where

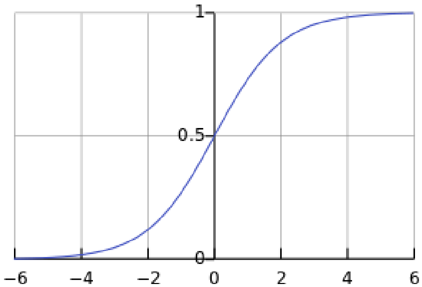
$$\begin{aligned} W &= \Sigma^{-1}(\mu_1 - \mu_2) \\ w_0 &= -\frac{1}{2}\mu_1^\top \Sigma^{-1}\mu_1 + \frac{1}{2}\mu_2^\top \Sigma^{-1}\mu_2 + \ln \frac{P(C_1)}{P(C_2)} \end{aligned}$$

or simply

$$P(C_1|X) = \sigma\left(W'^\top X\right)$$

- We compute a linear combination of the inputs but then we pass through a function that ensures $0 \leq y_n \leq 1$ by defining.

$$y_n = \sigma(w^\top x) \triangleq \frac{1}{1 + \exp(-w^\top x)}.$$



- If we find W directly, we need to find D parameters.
- If we find $P(C_k|X)$ via probabilistic modeling of data using Gaussian distribution and MLE, we need
  1. $2D$ parameters for mean
  2. $\frac{D(D+1)}{2}$ parameters for shared covariance matrix
  3. One parameter for $P(C_1)$

  resulting $\frac{D(D+5)}{2} + 1$ parameters.
- This results in Logistic regression classifier.

Logistic regression

- Logistic regression is a model for probabilistic classification.
- It predicts label probabilities rather than a hard value of the label.
- Let

$$y_n = P(C_1|x_n)$$
$$1 - y_n = P(C_2|x_n)$$

- The output of Logistic regression is a probability defined using the sigmoid function

$$P(C_1|x_n) = y_n = \sigma\left(W^\top x_n\right) = \frac{1}{1 + exp(-W^\top x_n)}$$

- The log of the ratio of probabilities $\ln \frac{P(C_1|x_n)}{P(C_2|x_n)}$ for the two classes, also known as the log odds equals to

$$\ln \frac{P(C_1|x_n)}{P(C_2|x_n)} = \ln \exp\left(W^\top x_n\right) = W^\top x_n$$

- Thus if $W^\top x_n > 0$, the probable class is C_1.

- One loss function may be

$$\ell(t_n, h(x_n)) = (t_n - h(x_n))^2$$

This loss function is not a convex function and is not easy to optimize.

- The likelihood function can be written

$$p(t|w) = \begin{cases} y_n & t_n = 1 \\ (1 - y_n) & t_n = 0 \end{cases}$$

- If $t_n = 1$ but y_n is close to 0 then loss will be high.
- If $t_n = 0$ but y_n is close to 1 then loss will be high.
- The likelihood function can also be written

$$p(t|w) = y_n^{t_n}(1 - y_n)^{(1-t_n)}$$

- We can define a loss function by taking the negative logarithm of the likelihood.

$$\mathcal{L}(W) = -\ln \prod_{n=1}^{N} \ell(t_n, h(x_n)) = -\sum_{n=1}^{N} [t_n \ln y_n + (1 - t_n) \ln(1 - y_n)]$$

- This loss function is called the cross-entropy loss.

- Let $t_n \in \{-1, +1\}$. Another way to write the log-likelihood of data is.

$$
\begin{aligned}
p(+1|x) &= \frac{1}{1+\exp(-w^\top x)} \\
p(-1|x) &= \frac{1}{1+\exp(+w^\top x)}
\end{aligned}
$$

- By combining the above equations and computing negative log-likelihood of data, we obtain

$$
\begin{aligned}
\mathcal{L}(w) &= -\sum_{n=1}^{N} \ln \frac{1}{1+\exp(-t_n w^\top x_n)} \\
&= \sum_{n=1}^{N} \ln \left[1+\exp(-t_n w^\top x_n)\right]
\end{aligned}
$$

- Unlike linear regression, we can no longer write down the minimum of negative log-likelihood in the closed form. Instead, we need to use an optimization algorithm for computing it.

- Computing the gradients of $L(w)$ with respect to w, we obtain

$$\nabla\mathcal{L}(w) \quad = \quad \sum_{n=1}^{N} t_n x_n (y_n - t_n)$$

- Updating the weight vector using the gradient descent rule will result in

$$W^{(k+1)} \quad = \quad W^{(k)} - \eta \sum_{n=1}^{N} t_n x_n (y_n - t_n)$$

  η is the learning rate.

- In order to have a good trade-off between the training error and the generalization error, we can add the regularization term.

$$\mathcal{L}(w) \quad = \quad \sum_{n=1}^{N} \log\left[1 + \exp(-t_n w^\top x_n)\right] + \frac{\lambda}{2}\|w\|^2$$

- Using the gradient descent rule, will result in the following updating rule.

$$W^{(k+1)} \quad = \quad W^{(k)} - \eta \sum_{n=1}^{N} t_n x_n (y_n - t_n) - \lambda W^{(k)}$$

MLE formulation of Logistic regression

- In linear regression, we often assume that the noise has a Gaussian distribution.

$$p(t|x, w) = \mathcal{N}(t|\mu(x), \sigma^2(x))$$

- We can generalize the linear regression to binary classification by making two changes:
  - First, replacing the Gaussian distribution for t with Bernoulli distribution, which is more appropriate for classification.

$$p(t_n|x_n, w) = Ber(t_n|y_n) = \begin{cases} y_n & \text{if } t_n = 1 \\ 1 - y_n & \text{if } t_n = 0 \end{cases}$$

    where $\mu(x_n) = \mathbb{E}[t_n|x_n] = p(t_n = 1|x_n)$.
  - This is equivalent to

$$p(t_n|x_n, w) = Ber(t_n|\mu(x_n)) = \mu(x_n)^{t_n}(1 - \mu(x_n))^{(1-t_n)}$$

  - Second, compute a linear combination of the inputs and then we pass this through a function that ensures $0 \leq \mu(x) \leq 1$ by defining

$$\mu(x) = \sigma(w^\top x)$$

- Putting these two steps together and dropping index n, we obtain

$$p(t|x,w) = Ber(t|\sigma(w^\top x)).$$

- This is called logistic regression due to its similarity to linear regression.
- If we threshold the output probability at $\frac{1}{2}$, we can introduce a decision rule of the form

$$\text{if } p(t=1|x) > 0.5 \Longleftrightarrow h(x) = 1.$$

- Logistic regression learns weights so as to maximize the (log-)likelihood of the data.
- Let $S = \{(x_1,t_1),(x_2,t_2),\ldots,(x_N,t_N)\}$ be the training set. The negative log-likelihood of data equals

$$\begin{aligned}\mathcal{L}(w) &= -\ln \prod_{n=1}^{N} y_n^{t_n}(1-y_n)^{(1-t_n)} \\ &= -\sum_{n=1}^{N} t_n \ln y_n + (1-t_n)\ln(1-y_n)\end{aligned}$$

This is called the cross-entropy error function.

MAP formulation of Logistic regression

MAP formulation for Logistic regression

- Maximum likelihood estimate of W can lead to overfitting when data set is linearly separable. A solution is to use a prior on w.
- This can be avoided by inclusion of a prior and finding a MAP solution or equivalently by adding a regularization term to the error function.
- Same as linear regression, we consider a Gaussian prior on w

$$p(W) = \mathcal{N}(0, \sigma_0^2 I_D).$$

- I_D denotes the $D \times D$ identity matrix. This is equivalent to assume that the prior selects each component of W independently from a $\mathcal{N}(0, \sigma_0^2)$. This prior can be written as

$$p(W) = \frac{1}{(2\pi)^{D/2}\sigma_0^D} exp\left\{-\frac{1}{2\sigma_0^2}||W||_2^2\right\}.$$

- Assume that noise precision is known, The posterior density of W given set S and solving the equation gives the form

$$\mathcal{L}(w) = \sum_{n=1}^{N} \log\left[1 + \exp(-t_n w^\top x_n)\right] + \frac{\lambda}{2}\|w\|^2$$

- Thus MAP estimation is equivalent to regularized logistic regression.
- Using the gradient descent rule, will result in the following updating rule.

$$W^{(k+1)} \quad = \quad W^{(k)} - \eta \sum_{n=1}^{N} t_n x_n (y_n - t_n) - \lambda W^{(k)}$$

Reading

Readings

1. Sections 4.3.2 of Pattern Recognition and Machine Learning Book (Bishop 2006).
2. Chapter 8 of Machine Learning: A probabilistic perspective (Murphy 2012).
3. Chapter 10 of Probabilistic Machine Learning: An introduction (Murphy 2022).

- Bishop, Christopher M. (2006). *Pattern Recognition and Machine Learning*. Springer-Verlag.
- Murphy, Kevin P. (2012). *Machine Learning: A Probabilistic Perspective*. The MIT Press.
- – (2022). *Probabilistic Machine Learning: An introduction*. The MIT Press.

Questions?