Linear Models for Classification

Regression: last week we talked about linear regression that we try to fit.

Classification Goal:

Take a D-dimensional vector $\boldsymbol{x}$ and assign it one of K discrete classes C_k $(k = 1, \dots, K)$.

The input space is divided into decision regions bounded by decision boundaries.

Linear model for classification: decision surfaces define $(D-1)$-dimensional hyperplanes.

1-of-K coding: $t = (0,1,0,0,0)^T$ – meaning the class chosen is 2.

Activation function:

$$y(\boldsymbol{x}) = f(\boldsymbol{w}^T\boldsymbol{x} + w_0)$$

And this function will give us the class of $\boldsymbol{x}$.

Discriminant function

Directly model the activation function. E.g. for binary classification, the function will be the hyperplane separating 0 and 1.

Say we have the input space with points indicating inputs.

A line $y(\boldsymbol{x})$ is the decision line, and in the discriminant case:

$$y(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + w_0$$

Assume there are $\boldsymbol{x}_a, \boldsymbol{x}_b$ that lay on the line, then they satisfy a set of linear equations:

$$y(\boldsymbol{x}_a) = \boldsymbol{w}^T\boldsymbol{x}_a + w_0 = 0$$

$$y(\boldsymbol{x}_b) = \boldsymbol{w}^T\boldsymbol{x}_b + w_0 = 0$$

$$\Rightarrow\ y(\boldsymbol{x}_a) - y(\boldsymbol{x}_b) = \boldsymbol{w}^T(\boldsymbol{x}_a - \boldsymbol{x}_b)$$

Therefore $\boldsymbol{w}$ – the vector of weights – is going to be perpendicular to the decision line.

Now assume a single point $\boldsymbol{x}$ lays on the line, then:

$$y(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + w_0 = 0 \Rightarrow \boldsymbol{w}^T\boldsymbol{x} = -w_0$$

Then:

$$\frac{\boldsymbol{w}^T\boldsymbol{x}}{|\boldsymbol{w}|} = -\frac{w_0}{|\boldsymbol{w}|}$$

Say we have some point $\boldsymbol{x}$ not on the line, then it can be written as:

$$\boldsymbol{x} = \boldsymbol{x}_\perp + r\frac{\boldsymbol{w}}{|\boldsymbol{w}|}$$

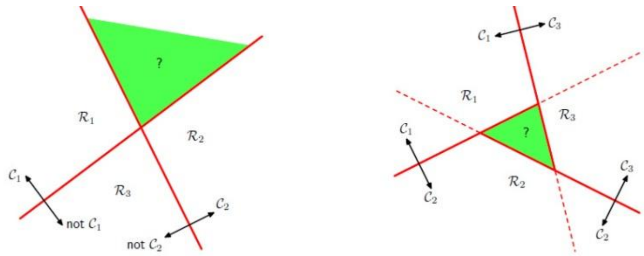
Where $\boldsymbol{x}_\perp$ is the projection of the point onto the line in a perpendicular direction (the shortest projection).

And now:

$$y(\boldsymbol{x}) = y\left(\boldsymbol{x}_\perp + r\frac{\boldsymbol{w}}{|\boldsymbol{w}|}\right) = \boldsymbol{w}^T\left(\boldsymbol{x}_\perp + r\frac{\boldsymbol{w}}{|\boldsymbol{w}|}\right) = \underbrace{\boldsymbol{w}^T\boldsymbol{x}_\perp}_{=0} + r\frac{\boldsymbol{w}^T\boldsymbol{w}}{|\boldsymbol{w}|} + w_0 = r\frac{\boldsymbol{w}^T\boldsymbol{w}}{|\boldsymbol{w}|} \Rightarrow r = \frac{y(\boldsymbol{x})}{|\boldsymbol{w}|}$$

With a single line it's a binary classification.

Multiple classes:



(a) One-versus-the-rest (b) one-versus-one

a. Find the line that best separates one class vs. the others. The obvious problem with that is that we will have regions that we don't really know how to assign a class in them (the green zone in the figure).

b. All-pairs lines, a total of $\binom{K}{2}$ lines. Then we again get a green "unknown" region.

We will use K different discriminant functions:

$$y_k(\boldsymbol{x}) = \boldsymbol{w}_k^T \boldsymbol{x} + w_{k0}$$

And assign the class for the k that satisfies $k = argmax\ y_k(\boldsymbol{x})$.

Least Squares classification

Each class C_k is described by its own linear model: $y_l(x) = w_k^T x + w_{k0}$

$$y(x) = \tilde{W}^T \tilde{x}$$

Given a training data set $\{x_n, t_n\}_{n=1}^N$, sum-of-squares error function (slide 10):

$$E_D\left(\tilde{W}\right) = \frac{1}{2} Tr\left\{\left(\tilde{X}\tilde{W} - T\right)^T \left(\tilde{X}\tilde{W} - T\right)\right\}$$

Where

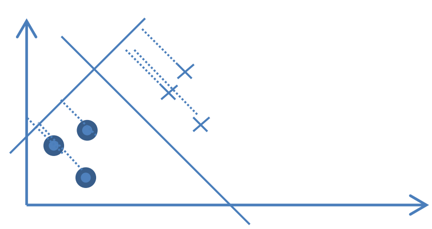
$$\tilde{W} = \left(\tilde{X}^T \tilde{X}\right)^{-1} \tilde{X}^T\, T = \tilde{X}^{\dagger} T, \qquad y(x) = \tilde{W}^T \tilde{x} = T^T \left(\tilde{X}^{\dagger}\right)^T \tilde{x}$$

This is very sensitive to outliers: fitting a line using this method might be messed up by outliers.

Fisher's Linear Discriminant

$y = w^T x$ is a dot product of w^T and x, i.e. it is a projection on some line.

x are points in the space, are projected on w which is a line perpendicular to y, and we are looking at the distribution of those projections, and we want those projections to be as separable as possible.



Binary classification with N_1 points of C_1 and N_2 points of C_2:

$$m_1 = \frac{1}{N_1}\sum_{n\in C_1} x_n \,, m_2 = \frac{1}{N_2}\sum_{n\in C_2} x_n$$

Between class distance: $m_2 - m_1 = w^T(m_2 - m_1)$

We want to maximize the between-class covariance – distance between the two means, and minimize the within-class covariance. That is the Fisher criterion:

$$J(w) = \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2} = \frac{w^T S_B w}{w^T S_W w}$$

$S_B = (m_2 - m_1)(m_2 - m_1)^T$ – the covariance of the means

$S_W = \sum_{n\in C_1}(x_n - m_1)(x_n - m_1)^T + \sum_{n\in C_2}(x_n - m_2)(x_n - m_2)^T$ - the in-class covariance

We want to maximize this:

$$\frac{\partial J(w)}{\partial w} = \frac{(w^T S_B w)'(w^T S_W w) - (w^T S_W w)'(w^T S_B w)}{(w^T S_W w)^2} = \frac{(w^T S_W w) S_B w - (w^T S_B w) S_W w}{(w^T S_W w)^2} = 0 \Leftrightarrow$$

$$\underbrace{(w^T S_W w)}_{scalar} S_B w = \underbrace{(w^T S_B w)}_{scalar} S_W w \Rightarrow$$

$$\boxed{w \propto S_W^{-1}(m_2 - m_1)}$$

Perceptron Algorithm

Perceptron function:

$$y(x) = f\left(w^T\phi(x)\right)$$

Where

$$f(a) = \begin{cases} +1, a \geq 0 \ (C_1) \\ -1, a < 0 \ (C_2) \end{cases}$$

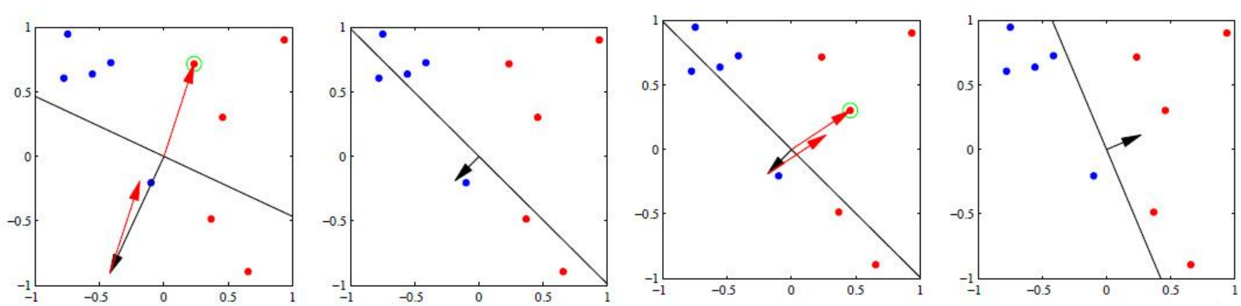
For each input point x we have a target value t – a binary target, and we want to have the function satisfy them: $w^T\phi(x_n)t_n > 0$. Therefore:

Perceptron criterion: minimize $E_P = -\sum_{n\in M} w^T\phi(x_n)t_n$

We can solve it with stochastic descent ($\phi_n \coloneqq \phi(x_n)$):

$$w^{(\tau+1)} = w^{(\tau)} - \eta\nabla E_P = w^{(\tau)} + \eta\phi_n t_n$$

Example for the descent (iterative process):



It can be proven that if such a line exists, this iterative process converges.

At each iteration a misclassified point is taken, and w is added the red vector to that misclassified point, resulting with a new w (and a new line – w is perpendicular to it).

Logistic Sigmoid Function

Probabilistic generative models for binary class problems:

$$p(C_1|x) = \frac{p(x|C_1)p(C_1)}{p(x|C_1)p(C_1) + p(x|C_2)p(C_2)} = \frac{1}{1 + \exp(-a)} = \sigma(a)$$

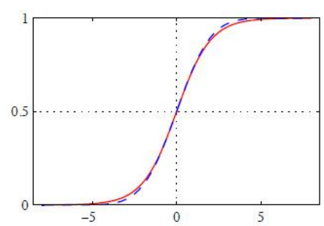
Where: $a = \ln \frac{p(x|C_1)p(C_1)}{p(x|C_2)p(C_2)}$

Development of $p(C_1|x)$ above:

$$\frac{p(x|C_1)p(C_1)}{p(x|C_1)p(C_1) + p(x|C_2)p(C_2)} = \frac{\frac{1}{p(x|C_1)p(C_1) + p(x|C_2)p(C_2)}}{p(x|C_1)p(C_1)} = \frac{1}{1 + \left(\frac{p(x|C_2)p(C_2)}{p(x|C_1)p(C_1)}\right)^{-1}} = \frac{1}{1 + \exp\left(-\, ln \frac{p(x|C_2)p(C_2)}{p(x|C_1)p(C_1)}\right)}$$

Logistic sigmoid function: $\boxed{\sigma(a) = \frac{1}{1+\exp(-a)}}$

and it looks like this:



<u>Probabilistic generative models</u> (slide 24):

In binary class problems:

$$p(x|C_k) = \frac{1}{(2\pi)^{\frac{D}{2}}} \frac{1}{|\Sigma|^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}(x - \mu_k)^T \Sigma^{-1} (x - \mu_k)\right\}$$

$$p(C_1|x) = \sigma(w^T x + w_0)$$

Where:

$$w = \Sigma^{-1}(\mu_1 - \mu_2)$$

$$w_0 = -\frac{1}{2}\mu_1^T \Sigma^{-1} \mu_1 + \frac{1}{2}\mu_2^T \Sigma^{-1} \mu_2 + \ln \frac{p(C_1)}{p(C_2)}$$

(This can be derived from σ)

Softmax Function

Sigmoid functions in multiclass problems:

$$p(C_k|x) = \frac{p(x|C_k)p(C_k)}{\sum_j p(x|C_k)p(C_k)} = \frac{\exp(a_k)}{\sum_j \exp(a_j)}, \quad a_k = \ln p(x|C_k)p(C_k)$$

Maximum Likelihood parameter estimation: (slide 31)

With $K = 2$:

$\{x_n, t_n\}_{n=1}^{N}, C_k \equiv (t_k = 1), p(C_1) = \pi$ (the prior that is unknown), $p(C_2) = 1 - \pi$ and Gaussian class conditional densities (likelihoods).

Likelihood:

$$p(t|\pi, \mu_1, \mu_2, \Sigma) = \prod_{n=1}^{N} [\pi \mathcal{N}(x_n|\mu_1, \Sigma)]^{t_n} [(1-\pi)\mathcal{N}(x_n|\mu_2, \Sigma)]^{1-t_n}$$

Where $t = (t_1, \ldots, t_N)^T$

Estimation (class exercise):

$$\ln p(t|\pi, \mu_1, \mu_2, \Sigma) = \sum_{n=1}^{N} \ln([\pi \mathcal{N}(x_n|\mu_1, \Sigma)]^{t_n} [(1-\pi)\mathcal{N}(x_n|\mu_2, \Sigma)]^{1-t_n}) =$$

$$\sum_{n=1}^{N} [\ln[\pi \mathcal{N}(x_n|\mu_1, \Sigma)]^{t_n} + \ln[(1-\pi)\mathcal{N}(x_n|\mu_2, \Sigma)]^{1-t_n}] =$$

$$\sum_{n=1}^{N} [t_n(\ln \pi + \ln \mathcal{N}(x_n|\mu_1, \Sigma)) + (1-t_n)(\ln(1-\pi) + \ln \mathcal{N}(x_n|\mu_2, \Sigma))] =$$

$$\frac{\partial \ln p(t|\pi, \mu_1, \mu_2, \Sigma)}{\partial \pi} = \sum_{n=1}^{N} \left[\frac{t_n}{\pi} + (1-t_n) \cdot \frac{1}{1-\pi} \cdot -1\right] = \sum_{n=1}^{N} \frac{t_n}{\pi} + \frac{t_n - 1}{1-\pi} = \cdots$$

Class solution:

The Log of the likelihood with π in it:

$$A \coloneqq \sum_{n=1}^{N} [t_n \ln \pi + (1-t_n) \ln(1-\pi)]$$

$$\frac{\partial A}{\partial \pi} = \sum \left(\frac{t_n}{\pi} - \frac{1-t_n}{1-\pi}\right) = 0 \Leftrightarrow \sum (1-\pi)t_n - \pi(1-t_n) = \sum (t_n - \pi t_n - \pi + \pi t_n) = 0 \Leftrightarrow \boxed{\pi = \frac{1}{N}\sum_{n=1}^{N} t_n}$$

And since $t_n = 1$ for C_1 then $\pi = \frac{1}{N} N_1$(and of course $1 - \pi = \frac{1}{N} N_2$)

For μ_1: the same, take log likelihood only for terms with μ_1:

$$B \coloneqq \sum_{n=1}^{N} t_n \ln \mathcal{N}(x_n|\mu_1, \Sigma) = -\frac{1}{2} \sum t_n (x_n - \mu_1)^T \Sigma^{-1} (x_n - \mu_1)$$

$$\frac{\partial B}{\partial \mu_1} = 0 \Leftrightarrow \cdots \Leftrightarrow \boxed{\mu_1 = \frac{1}{N_1} \sum_{n=1}^{N} t_n x_n}$$

The trick to solve the above … is compare linear and quadratic terms.

For μ_2: $\boxed{\mu_2 = \frac{1}{N_2} \sum_{(n=1)}^{N} (1-t_n) x_n}$

Estimating Σ: slide 35.

Probabilistic Discriminative Models

Logistic regression (for classification):

$p(C_1|\phi) = y(\phi) = \sigma(w^T\phi)$

For data set $\{\phi_n, t_n\}$ where $t_n \in \{0,1\}$, $\phi_n = \phi(x_n)$

Likelihood to estimate the parameters of the logistic regression model:

$$p(t|w) = \prod_{n=1}^{N} y_n^{t_n}\{1-y_n\}^{1-t_n}, \;\; t = (t_1, \dots, t_N)^T, y_n = p(C_1|\phi_n)$$

Cross-entropy error function:

$$E[w] = -\ln p(t|w) = -\sum_{n=1}^{N}\{t_n \ln y_n + (1-t_n)\ln(1-y_n)\}, \qquad y_n = \sigma(a_n), \qquad a_n = w^T\phi_n$$

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

Newton-Raphson method: (slide 44+).