6 Probabilistic Models for Classification

The main idea is to compute posterior probabilities distribution, so we want to estimate the posterior probability: $P(C_i|x)$.

Two possible ways:

1. GENERATIVE MODELS (estimate $P(C_i|X)$ through $P(x|C_i)$ and Bayes theorem)

2. DISCRIMINATIVE MODELS (estimate $P(C_i|x)$ directly)

Both methods TRY TO MAXIMIZE THE LIKELIHOOD.

1 Probabilistic generative model

Compute the probability by using the Bayes theorem.

$$\boxed{P(C_1|x) = \frac{p(x|C_1)\,P(C_1)}{\underbrace{p(x|C_1)P(C_1) + p(x|C_2)p(C_2)}_{\text{TOTAL PROB. APPLICATION}}}}$$

Bayes theorem.

All definitions can be extended also to more classes. We can reformulate the formula with the sigmoid function of a term,

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

We have first to define a model and we assume that the distribution of the input is given by a Gaussian function. (This is just an assumption).

$$p(x|C_i) = \underbrace{\mathcal{N}}_{\text{Gaussian function}}(x|\mu_i, \Sigma)$$

μ_i = mean

Σ = covariance matrix.

We assume that covariance matrix is the same for all the classes and the means are different.

$$P(C_1|x) = \sigma(\vec{w}^T\vec{x} + w_0)$$

where $\boxed{\vec{w} = \Sigma^{-1}(\vec{\mu}_1 - \vec{\mu}_2)}$ (*)

How can we estimate the parameters of our model? We have to learn the parameters of this model that are μ_1 and μ_2.

$$(*) \quad \boxed{w_0 = -\frac{1}{2}\vec{\mu}_1^T \Sigma^{-1} \vec{\mu}_1 + \frac{1}{2}\mu_2^T \Sigma^{-1} \mu_2 + \ln\frac{P(C_1)}{P(C_2)}}$$

We compute the likelihood and then we solve the optimization problem to find the maximum likelihood

WE ASSUME THAT ALL THE SAMPLES ARE INDEPENT EACH OTHER

we have just products

$$P(C_1) = \pi \quad \text{and} \quad P(C_2) = 1 - \pi$$

- Dataset $D = \{(x_M, t_M)_{M=1}^{N}\}$ where
- $t_M = \begin{cases} 0 & \text{if } x_M \in C_2 \\ 1 & \text{if } x_M \in C_1 \end{cases}$
- $N_1 = \#\text{samples} \in C_1$, $N_2 = \#\text{samples} \in C_2$

Likelihood:
propability that the values t_M will be generated given the input $\vec{x}$ and the parameter of the models. The parameters are UNKNOWN.

$$P(t|\pi, \mu_1, \mu_2, \Sigma) = \prod_{M=1}^{N} [\pi N(x|\mu_1, \Sigma)]^{t_M} [(1-\pi)N(x|\mu_2, \Sigma)]^{(1-t_M)}$$

We first COMPUTE the logarithm of the likelyhood, since the log is monotonic and does not affect the argmax. Then we compute the DERIVATIVE with respect to π, μ_1, μ_2 and put it to zero. The solution is simple and intuitive!

$$\pi = \frac{N_1}{N}$$ is estimated in this way.

$$\mu_1 = \frac{1}{N_1} \sum_{n=1}^{N} t_n x_n \qquad \mu_2 = \frac{1}{N_2} \sum_{n=1}^{N} (1-t_n) x_n$$

$$\Sigma = \frac{N_1}{N} S_1 + \frac{N_2}{N} S_2 \longrightarrow \text{weighted avg of } S_1 \text{ and } S_2$$

where $S_i = \frac{1}{N_i} \sum_{n \in C_i} (x_n - \mu_i)(x_n - \mu_i)^T$ ← square diff. between samples and the mean

The results can be affected by the fact that we assumed the Gaussian distribution.

Decision rule for two classes : $c = C_1 \iff P(c = C_1 | x) > 0.5$

For more classes, the decision is the argmax wrt all the classes

2) Probabilistic discriminative models

Again based on the maximum likelyhood but it does it DIRECTLY, withouth using the Bayes theorem.

Existimate $P(C_i|x)$ directly. Logistic regression is a classification method based on maximum likelihood.

Given a dataset $D = \{x_n, t_n\}_{n=1}^{N}$, with $t_n \in \{0,1\}$, but we consider a new set of samples where x_n are transformed by a non linear function ϕ: $D = \{\phi_n, t_n\}$.

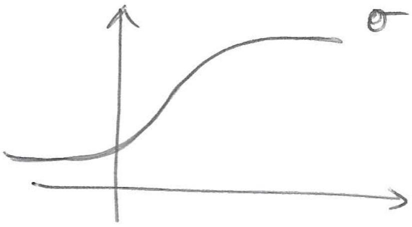
Likelihood function

$$p(t|w) = \prod_{n=1}^{N} y_n^{t_n} (1-y_n)^{1-t_n}$$

where $y_n = p(C_1|\phi_n) = \sigma(\vec{w}^T \phi_n)$ → linear combination of input and weights.

↑ sigmoid function of a linear model.

The output of the linear transformation $\vec{w}^T \phi_n$ is modified by the sigmoid function. The sigmoid function has a shape like this:



is similar to sign function, but it is continuos and DIFFERENTIABLE, so it is more useful in some ML context.

We find parameter $\vec{w}$ by MAXIMIZING the maximum likelyhood. We do the logarithm of the likelyhood:

$$\ln(p(t|w)) \Rightarrow \underbrace{-\ln(p(t|w)) = E(w)}_{\text{ERROR FUNCTION.}}$$

$$E(w) = -\sum_{n=1}^{N} \left[t_n \ln y_n + (1-t_n)\ln(1-y_n) \right]$$

To maximize the likelihood, we should MINIMIZE THE ERROR and by doing so we compute the gradient (the derivative).

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

derivative wrt $\vec{w}$

The method that we apply is the NEWTON-RAPHSON, that is based on an iterative approach for minimizing $E(\vec{w})$

$$\boxed{w \leftarrow w - H^{-1} \nabla E(w)}$$

H = is the second derivative of the error function

$H = \nabla\nabla E(w)$, HESSIAN MATRIX.

The idea of the solution of the problem is to MAXIMIZE LIKELIHOOD FUNCTION $\Rightarrow$ MINIMIZATION of an ERROR FUNCTION.

We can rewrite formulas as follow:

$$\nabla E(\vec{w}) = \vec{\phi}^T (\vec{y} - \vec{t})$$

$$H = \nabla\nabla E(\vec{w}) = \sum_{M=1}^{N} y_M (1 - y_M) \phi_M \phi_M^T = \vec{\phi}^T R \vec{\phi}$$

$$\vec{t} = (t_1, \dots, t_M)^T, \quad \vec{y} = (y_1, \dots, y_M)^T$$

R: diagonal matrix with $R_{MM} = y_M (1 - y_M)$

$$\vec{\phi} = \begin{pmatrix} \phi_1^T \\ \vdots \\ \phi_N^T \end{pmatrix}$$

The iterative method:

1. initialize $\vec{w}$ (at random)
2. Repeat until TERMINATION CONDITION:

$$\vec{w} \leftarrow \vec{w} - (\vec{\phi}^T R \vec{\phi})^{-1} \vec{\phi}^T (\vec{y} - \vec{t})$$

Once you have this sigmoid function you can use it as a classification discriminant. This method is called LOGISTIC REGRESSION, that is different from REGRESSION (learning a continuous function), LOGISTIC REGRESSION IS A CLASSIFICATION METHOD.

When you take any regression method and you apply the logistic function, than this model can be used for classification.

This method extends to K classes, now you have K $\vec{w}$ parameters for each class. In making the gradient we have to consider the derivatives for each w_J:

$$\nabla = \left(\frac{\partial}{\partial w_1} ; \frac{\partial}{\partial w_2} ; \dots ; \frac{\partial}{\partial w_K} \right)$$

$$\nabla_{w_J} (E(w_1, \dots, w_K)) = \dots$$