Probabilistic classification

Course of Machine Learning
Master Degree in Computer Science

University of Rome “Tor Vergata”

a.a. 2019-2020

Giorgio Gambosi

Naive Bayes classifiers recap

A language model is a (categorical) probability distribution on a vocabulary of terms (possibly, all words which occur in a large collection of documents).

Use

A language model can be applied to predict (generate) the next term occurring in a text. The probability of occurrence of a term is related to its information content and is at the basis of a number of information retrieval techniques.

Hypothesis

It is assumed that the probability of occurrence of a term is independent from the preceding terms in a text (bag of words model).

Bayesian classifiers

A language model can be applied to derive document classifiers into two or more classes through Bayes' rule.

- given two classes C_1, C_2, assume that, for any document d, the probabilities $p(C_1|d)$ and $p(C_2|d)$ are known: then, d can be assigned to the class with higher probability
- how to derive $p(C_k|d)$ for any document, given a collection $\mathcal{C}_1$ of documents known to belong to C_1 and a similar collection $\mathcal{C}_2$ for C_2? Apply Bayes' rule:

$$p(C_k|d) \propto p(d|C_k)p(C_k)$$

  the evidence $p(d)$ is the same for both classes, and can be ignored.
- we have still the problem of computing $p(C_k)$ and $p(d|C_k)$ from $\mathcal{C}_1$ and $\mathcal{C}_2$

Bayesian classifiers

Computing $p(C_k)$

The prior probabilities $p(C_k)$ $(k = 1, 2)$ can be easily estimated from $\mathcal{C}_1, \mathcal{C}_2$: for example, by applying ML, we obtain

$$p(C_k) = \frac{|\mathcal{C}_1|}{|\mathcal{C}_1| + |\mathcal{C}_2|}$$

Naive bayes classifiers

Computing $p(d|C_k)$

For what concerns the likelihoods $p(d|C_k)$ ($k = 1, 2$), we observe that d can be seen, according to the bag of words assumption, as a multiset of n_d terms

$$d = \{\bar{t}_1, \bar{t}_2, \ldots, \bar{t}_{n_d}\}$$

By applying the product rule, it results

$$\begin{aligned} p(d|C_k) &= p(\bar{t}_1, \ldots, \bar{t}_{n_d}|C_k) \\ &= p(\bar{t}_1|C_k)p(\bar{t}_2|\bar{t}_1, C_k) \cdots p(\bar{t}_{n_d}|\bar{t}_1, \ldots, \bar{t}_{n_d-1}, C_k) \end{aligned}$$

Naive bayes classifiers

The naive Bayes assumption

Computing $p(d|C_k)$ is much easier if we assume that terms are pairwise conditionally independent, given the class C_k, that is, for $i, j = 1. \ldots, n_d$ and $k = 1, 2$,

$$p(\bar{t}_i, \bar{t}_j|C_k) = p(\bar{t}_i|C_k)p(\bar{t}_2|C_k)$$

as, a consequence,

$$p(d|C_k) = \prod_{j=1}^{n_d} p(\bar{t}_j|C_k)$$

Language models and NB classifiers

The probabilities $p(\bar{t}_j|C_k)$ are available for all terms if language models have been derived for C_1 and C_2, respectively from documents in $\mathcal{C}_1$ and $\mathcal{C}_2$.

Generative models

Given a language model, it is possible to sample from the distribution to generate random documents statistically equivalent to the documents in the collection used to derive the model.

Gaussian discriminant analysis

In Gaussian discriminant analysis (GDA) all class conditional distributions $p(\mathbf{x}|C_k)$ are assumed gaussians. This implies that the corresponding posterior distributions $p(C_k|\mathbf{x})$ can be easily derived.

Hypothesis

All distributions $p(\mathbf{x}|C_k)$ have same covariance matrix $\mathbf{\Sigma}$, of size $D \times D$. Then,

$$p(\mathbf{x}|C_k) = \frac{1}{(2\pi)^{d/2}|\mathbf{\Sigma}|^{1/2}} \mathit{exp}\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^T\mathbf{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_k)\right)$$

Binary case

If $K = 2$,

$$p(C_1|\mathbf{x}) = \sigma(a(\mathbf{x}))$$

where

$$\begin{aligned}
a(\mathbf{x}) &= \log \frac{p(\mathbf{x}|C_1)p(C_1)}{p(\mathbf{x}|C_2)p(C_2)} \\
&= \log \frac{\frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}|^{1/2}} exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_1)^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_1)\right)p(C_1)}{\frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}|^{1/2}} exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_2)^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu}_2)\right)p(C_2)} \\
&= \frac{1}{2}(\boldsymbol{\mu}_2^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_2 - \mathbf{x}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_2 - \boldsymbol{\mu}_2^T\boldsymbol{\Sigma}^{-1}\mathbf{x}) - \\
&\quad - \frac{1}{2}(\boldsymbol{\mu}_1^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1 - \mathbf{x}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1 - \boldsymbol{\mu}_1^T\boldsymbol{\Sigma}^{-1}\mathbf{x}) + \log\frac{p(C_1)}{p(C_2)}
\end{aligned}$$

Binary case

Observe that the results of all products involving $\mathbf{\Sigma}^{-1}$ are scalar, hence, in particular

$$\mathbf{x}^T\mathbf{\Sigma}^{-1}\boldsymbol{\mu}_1 = \boldsymbol{\mu}_1^T\mathbf{\Sigma}^{-1}\mathbf{x}$$
$$\mathbf{x}^T\mathbf{\Sigma}^{-1}\boldsymbol{\mu}_2 = \boldsymbol{\mu}_2^T\mathbf{\Sigma}^{-1}\mathbf{x}$$

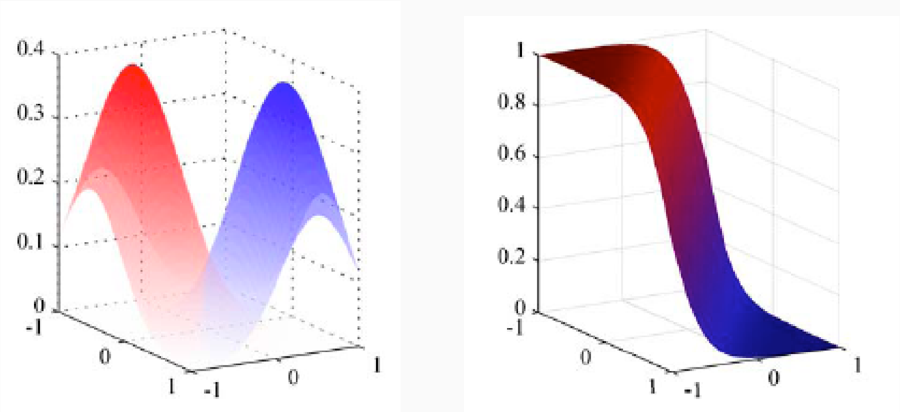
Then,

$$a(\mathbf{x}) = \frac{1}{2}(\boldsymbol{\mu}_2^T\mathbf{\Sigma}^{-1}\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1^T\mathbf{\Sigma}^{-1}\boldsymbol{\mu}_1) + (\boldsymbol{\mu}_1^T\mathbf{\Sigma}^{-1} - \boldsymbol{\mu}_2^T\mathbf{\Sigma}^{-1})\mathbf{x} + \log\frac{p(C_1)}{p(C_2)} = \mathbf{w}^T\mathbf{x} + w_0$$

with

$$\mathbf{w} = \mathbf{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$$
$$w_0 = \frac{1}{2}(\boldsymbol{\mu}_2^T\mathbf{\Sigma}^{-1}\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1^T\mathbf{\Sigma}^{-1}\boldsymbol{\mu}_1) + \log\frac{p(C_1)}{p(C_2)}$$

Example



Left, the class conditional distributions $p(\mathbf{x}|C_1), p(\mathbf{x}|C_2)$, gaussians with $D = 2$. Right the posterior distribution of C_1, $p(C_1|\mathbf{x})$ with sigmoidal slope.

Discriminant function

The discriminant function can be obtained by the condition $p(C_1|\mathbf{x}) = p(C_2|\mathbf{x})$, that is, $\sigma(a(\mathbf{x})) = \sigma(-a(\mathbf{x}))$.

This is equivalent to $a(\mathbf{x}) = -a(\mathbf{x})$ and to $a(\mathbf{x}) = 0$. As a consequence, it results

$$\mathbf{w}^T\mathbf{x} + w_0 = 0$$

or

$$\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)\mathbf{x} + \frac{1}{2}(\boldsymbol{\mu}_2^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1) + \log\frac{p(C_2)}{p(C_1)} = 0$$

Simple case: $\boldsymbol{\Sigma} = \lambda\mathbf{I}$ (that is, $\sigma_{ii} = \lambda$ for $i = 1, \ldots, d$). In this case, the discriminant function is

$$2(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)\mathbf{x} + ||\boldsymbol{\mu}_1||^2 - ||\boldsymbol{\mu}_2||^2 + 2\lambda\log\frac{p(C_2)}{p(C_1)} = 0$$

Multiple classes

In this case, we refer to the softmax function:

$$p(C_k|\mathbf{x}) = s(a_k(\mathbf{x}))$$

where $a_k(\mathbf{x}) = \log(p(\mathbf{x}|C_k)p(C_k))$.
By the above considerations, it easily turns out that

$$a_k(\mathbf{x}) = \frac{1}{2}\left(\boldsymbol{\mu}_k^T\boldsymbol{\Sigma}^{-1}\mathbf{x} - \boldsymbol{\mu}_k^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k\right) + \log p(C_k) - \frac{d}{2}\log(2\pi) - \frac{1}{2}\log|\boldsymbol{\Sigma}| = \mathbf{w}_k^T\mathbf{x} + w_{0k}$$

Multiple classes

Decision boundaries corresponding to the case when there are two classes C_j, C_k such that the corresponding posterior probabilities are equal, and larger than the probability of any other class. That is,

$$p(C_k|\mathbf{x}) = p(C_j|\mathbf{x}) \qquad\qquad p(C_i|\mathbf{x}) < p(C_k|\mathbf{x}) \quad i \neq j, k$$

hence

$$e^{a_k(\mathbf{x})} = e^{a_j(\mathbf{x})} \qquad\qquad e^{a_i(\mathbf{x})} < e^{a^k(\mathbf{x})} \quad i \neq j, k$$

that is,

$$a_k(\mathbf{x}) = a_j(\mathbf{x}) \qquad\qquad a_i(\mathbf{x}) < a^k(\mathbf{x}) \quad i \neq j, k$$

As shown, this implies that boundaries are linear.

General covariance matrices, binary case

The class conditional distributions $p(\mathbf{x}|C_k)$ are gaussians with different covariance matrices

$$
\begin{aligned}
a(\mathbf{x}) &= \log \frac{p(\mathbf{x}|C_1)p(C_1)}{p(\mathbf{x}|C_2)p(C_2)} \\
&= \log \frac{\mathit{exp}\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_1)^T\boldsymbol{\Sigma}_1^{-1}(\mathbf{x}-\boldsymbol{\mu}_1)\right)}{\mathit{exp}\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_2)^T\boldsymbol{\Sigma}_2^{-1}(\mathbf{x}-\boldsymbol{\mu}_2)\right)} + \frac{1}{2}\log\frac{|\boldsymbol{\Sigma}_2|}{|\boldsymbol{\Sigma}_1|} + \log\frac{p(C_1)}{p(C_2)} \\
&= \frac{1}{2}\Big((\mathbf{x}-\boldsymbol{\mu}_2)^T\boldsymbol{\Sigma}_2^{-1}(\mathbf{x}-\boldsymbol{\mu}_2) - (\mathbf{x}-\boldsymbol{\mu}_1)^T\boldsymbol{\Sigma}_1^{-1}(\mathbf{x}-\boldsymbol{\mu}_1)\Big) + \frac{1}{2}\log\frac{|\boldsymbol{\Sigma}_2|}{|\boldsymbol{\Sigma}_1|} + \log\frac{p(C_1)}{p(C_2)}
\end{aligned}
$$

General covariance matrices, binary case

By applying the same considerations, the decision boundary turns out to be

$$\left((\mathbf{x}-\boldsymbol{\mu}_2)^T\boldsymbol{\Sigma}_2^{-1}(\mathbf{x}-\boldsymbol{\mu}_2)-(\mathbf{x}-\boldsymbol{\mu}_1)^T\boldsymbol{\Sigma}_1^{-1}(\mathbf{x}-\boldsymbol{\mu}_1)\right)+\log\frac{|\boldsymbol{\Sigma}_2|}{|\boldsymbol{\Sigma}_1|}+2\log\frac{p(C_1)}{p(C_2)}=0$$

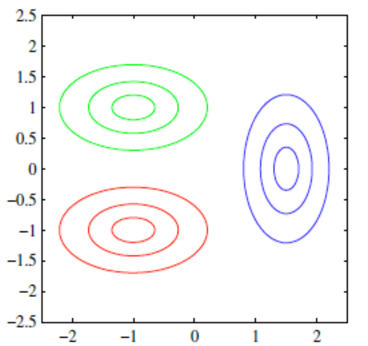
Classes are separated by a (at most) quadratic surface.

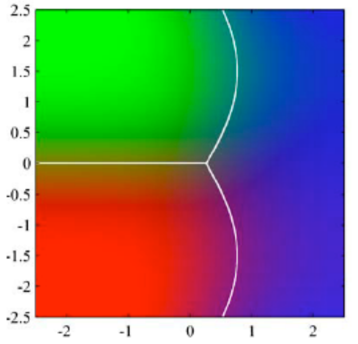
General covariance, multiple classe

It can be proved that boundary surfaces are at most quadratic.

Example

Left: 3 classes, modeled by gaussians with different covariance matrices.
Right: posterior distribution of classes, with boundary surfaces.





GDA and maximum likelihood

The class conditional distributions $p(\mathbf{x}|C_k)$ can be derived from the training set by maximum likelihood estimation.

For the sake of simplicity, assume $K = 2$ and both classes share the same $\mathbf{\Sigma}$.

It is then necessary to estimate $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2$, $\mathbf{\Sigma}$, and $\pi = p(C_1)$ (clearly, $p(C_2) = 1 - \pi$).

GDA and maximum likelihood

Training set $\mathcal{T}$: includes n elements $(\mathbf{x}_i, t_i)$, with

$$t_i = \begin{cases} 0 & \text{se } \mathbf{x}_i \in C_2 \\ 1 & \text{se } \mathbf{x}_i \in C_1 \end{cases}$$

If $\mathbf{x} \in C_1$, then $p(\mathbf{x}, C_1) = p(\mathbf{x}|C_1)p(C_1) = \pi \cdot \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_1, \boldsymbol{\Sigma})$
If $\mathbf{x} \in C_2$, $p(\mathbf{x}, C_2) = p(\mathbf{x}|C_2)p(C_2) = (1-\pi) \cdot \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_2, \boldsymbol{\Sigma})$

The likelihood of the training set $\mathcal{T}$ is

$$L(\pi, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\Sigma}|\mathcal{T}) = \prod_{i=1}^{n} (\pi \cdot \mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}))^{t_i} ((1-\pi) \cdot \mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_2, \boldsymbol{\Sigma}))^{1-t_i}$$

GDA and maximum likelihood

The corresponding log likelihood is

$$l(\pi, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\Sigma}|\mathcal{T}) = \sum_{i=1}^{n} (t_i \log \pi + t_i \log(\mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}))) +$$
$$+ \sum_{i=1}^{n} ((1-t_i)\log(1-\pi) + (1-t_i)\log(\mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_2, \boldsymbol{\Sigma})))$$

Its derivative wrt π is

$$\frac{\partial l}{\partial \pi} = \frac{\partial}{\partial \pi}\sum_{i=1}^{n}(t_i \log \pi + (1-t_i)\log(1-\pi)) = \sum_{i=1}^{n}\left(\frac{t_i}{\pi} - \frac{(1-t_i)}{1-\pi}\right) = \frac{n_1}{\pi} - \frac{n_2}{1-\pi}$$

which is equal to 0 for

$$\pi = \frac{n_1}{n}$$

GDA and maximum likelihood

The maximum wrt $\boldsymbol{\mu}_1$ (and $\boldsymbol{\mu}_2$) is obtained by computing the gradient

$$\frac{\partial l}{\partial \boldsymbol{\mu}_1} = \frac{\partial}{\partial \boldsymbol{\mu}_1} \sum_{i=1}^{n} t_i \log(\mathcal{N}(\mathbf{x}_i | \boldsymbol{\mu}_1, \boldsymbol{\Sigma})) = \cdots = \boldsymbol{\Sigma}^{-1} \sum_{i=1}^{n} t_i (\mathbf{x}_i - \boldsymbol{\mu}_1)$$

As a consequence, we have $\dfrac{\partial l}{\partial \boldsymbol{\mu}_1} = 0$ for

$$\sum_{i=1}^{n} t_i \mathbf{x}_i = \sum_{i=1}^{n} t_i \boldsymbol{\mu}_1$$

hence, for

$$\boldsymbol{\mu}_1 = \frac{1}{n_1} \sum_{\mathbf{x}_i \in C_1} \mathbf{x}_i$$

GDA and maximum likelihood

Similarly, $\dfrac{\partial l}{\partial \boldsymbol{\mu}_2} = 0$ for

$$\boldsymbol{\mu}_2 = \frac{1}{n_2} \sum_{\mathbf{x}_i \in C_2} \mathbf{x}_i$$

GDA and maximum likelihood

Maximizing the log-likelihood wrt $\boldsymbol{\Sigma}$ provides

$$\boldsymbol{\Sigma} = \frac{n_1}{n}\mathbf{S}_1 + \frac{n_2}{n}\mathbf{S}_2$$

where

$$\mathbf{S}_1 = \frac{1}{n_1}\sum_{\mathbf{x}_i \in C_1}(\mathbf{x}_i - \boldsymbol{\mu}_1)(\mathbf{x}_i - \boldsymbol{\mu}_1)^T$$

$$\mathbf{S}_2 = \frac{1}{n_2}\sum_{\mathbf{x}_i \in C_2}(\mathbf{x}_i - \boldsymbol{\mu}_2)(\mathbf{x}_i - \boldsymbol{\mu}_2)^T$$

and let

$$\mathbf{S} = \frac{n_1}{n}\mathbf{S}_1 + \frac{n_2}{n}\mathbf{S}_2$$

Generative models

For a large set of distributions type for $p(\mathbf{x}|C_k)$ the posterior class distributions $p(C_k|\mathbf{x})$ are sigmoidal (in the binary case) or softmax (for more classes): in both cases, with argument given by a linear combination of features in $\mathbf{x}$.

We may derive both the parameters of $p(\mathbf{x}|C_k)$ and the prior class probabilities $p(C_k)$ through maximum likelihood estimation, and next apply Bayes' rule to derive $p(C_k|\mathbf{x})$, at least up to a normalization factor.

Some general considerations

Observe that, in general, it is possible to write, in the binary classification case,

$$p(C_1|\mathbf{x}) = \frac{p(\mathbf{x}|C_1)p(C_1)}{p(\mathbf{x}|C_1)p(C_1) + p(\mathbf{x}|C_2)p(C_2)} = \frac{1}{1 + \frac{p(\mathbf{x}|C_2)p(C_2)}{p(\mathbf{x}|C_1)p(C_1)}}$$

if we define

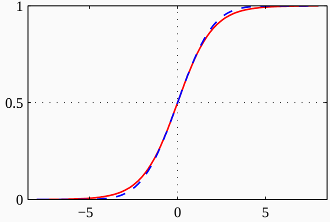
$$a = \log \frac{p(\mathbf{x}|C_1)p(C_1)}{p(\mathbf{x}|C_2)p(C_2)} = \log \frac{p(C_1|\mathbf{x})}{p(C_2|\mathbf{x})} = \log$$

that is, if a is the log of the ratio between the posterior probabilities (log odds), we obtain that

$$p(C_1|\mathbf{x}) = \frac{1}{1+e^{-a}} = \sigma(a) \qquad p(C_2|\mathbf{x}) = 1 - \frac{1}{1+e^{-a}} = \frac{1}{1+e^{a}}$$

where $\sigma(x)$ is the logistic function or (sigmoid)

Sigmoid



Useful properties of the sigmoid

- $\sigma(-x) = 1 - \sigma(x)$
- $\dfrac{d\sigma(x)}{dx} = \sigma(x)(1 - \sigma(x))$

Softmax

In the case $K > 2$, the general formula holds

$$p(C_k|\mathbf{x}) = \frac{p(\mathbf{x}|C_k)p(C_k)}{\sum_j p(\mathbf{x}|C_j)p(C_j)} = \frac{e^{a_k}}{\sum_j e^{a_j}} = s(a_k)$$

If we define, for each $k = 1, \ldots, K$

$$a_k(\mathbf{x}) = \log(p(\mathbf{x}|C_k)p(C_k)) = \log p(C_k|\mathbf{x}) + \log p(C_k)$$

then we may write

$$p(C_k|\mathbf{x}) = \frac{e^{a_k}}{\sum_j e^{a_j}} = s(a_k)$$

$s(\mathbf{x})$ is the softmax function (or normalized exponential) and it can be seen as an extension of the sigmoid to the case $K > 2$

$s(\mathbf{x})$ can be seen as a smoothed version of the maximum:

if $a_k \gg a_j$ for all $j \neq k$, then $s(a_k) \simeq 1$ and $s(a_j) \simeq 0$ for all $j \neq k$

Generalized linear models

A generalized linear model (GLM) is a function

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

where f is in general a non linear function.

Each iso-surface of $y(\mathbf{x})$, such that by definition $y(\mathbf{x}) = c$ (for some constant c), is such that

$$f(\mathbf{w}^T\mathbf{x} + w_0) = c$$

and

$$\mathbf{w}^T\mathbf{x} + w_0 = f^{-1}(y) = c'$$

(c' constant).
Hence, iso-surfaces of a GLM are hyper-planes, thus implying that boundaries are hyperplanes themselves.

Generative models and the exponential family

The property that $p(C_k|\mathbf{x})$ is a generalized linear model with sigmoid (for the binary case) and softmax (for the multiclass case) activation function holds more in general than assuming a gaussian or bernoulli class conditional distribution $p(\mathbf{x}|C_k)$.

Indeed, let the class conditional probability wrt C_k belong to the exponential family, that is it may be written in the form

$$p(\mathbf{x}|\boldsymbol{\theta}_k) = g(\boldsymbol{\theta}_k)f(\mathbf{x})e^{\boldsymbol{\phi}(\boldsymbol{\theta}_k)^T\mathbf{u}(\mathbf{x})}$$

with the additional constraint that $\mathbf{u}$ is the identity function, that is $\mathbf{u}(\mathbf{x}) = \mathbf{x}$.

Generative models and the exponential family

In the case of binary classification, we check that $a(\mathbf{x})$ is a linear function

$$
\begin{aligned}
a(\mathbf{x}) &= \log \frac{p(\mathbf{x}|\boldsymbol{\theta}_1)p(\boldsymbol{\theta}_1)}{p(\mathbf{x}|\boldsymbol{\theta}_2)p(\boldsymbol{\theta}_2)} = \log \frac{g(\boldsymbol{\theta}_1)e^{\frac{1}{s}\boldsymbol{\phi}(\boldsymbol{\theta}_1)^T\mathbf{x}}p(\boldsymbol{\theta}_1)}{g(\boldsymbol{\theta}_2)e^{\frac{1}{s}\boldsymbol{\phi}(\boldsymbol{\theta}_2)^T\mathbf{x}}p(\boldsymbol{\theta}_2)} \\
&= (\boldsymbol{\phi}(\boldsymbol{\theta}_1) - \boldsymbol{\phi}(\boldsymbol{\theta}_2))^T\mathbf{x} + \log g(\boldsymbol{\theta}_1) - \log g(\boldsymbol{\theta}_2) + \log p(\boldsymbol{\theta}_1) - \log p(\boldsymbol{\theta}_2)
\end{aligned}
$$

Similarly, for multiclass classification, we may easily derive that

$$
a_k(\mathbf{x}) = \boldsymbol{\phi}(\boldsymbol{\theta}_k)^T\mathbf{x} + \log g(\boldsymbol{\theta}_k) + p(\boldsymbol{\theta}_k)
$$

for all k.

Discriminative approach

Alternative idea

We could directly assume that $p(C_k|\mathbf{x})$ is sigmoidal (indeed a generalized linear model with sigmoidal non linear function) and derive it (for example through ML estimation of its parameters).

Comparison wrt the generative approach:

- Less information derived (we do not know $p(\mathbf{x}|C_k)$, thus we are not able to generate new data)
- Simpler method, usually a smaller set of parameters to be derived
- Better predictions, if the assumptions done with respect to $p(\mathbf{x}|C_k)$ are poor.

Logistic regression

Logistic regression is a GLM such that

$$p(C_1|\mathbf{x}) = \sigma(\mathbf{w}^T\mathbf{x}) = \frac{1}{1 + e^{-\mathbf{w}^T\overline{\mathbf{x}}}}$$

where the use of basis functions is explicitly considered.

The model is equivalent, for the binary classification case, to linear regression for the regression case.

Degrees of freedom

- In the case of d features, logistic regression requires $d+1$ coefficients $w_0, \ldots, w_d$ to be derived from a training set
- A generative approach with gaussian distributions requires:
  - $2d$ coefficients for the means $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2$,
  - for each covariance matrix
$$\sum_{i=1}^{d} i = d(d+1)/2 \qquad \text{coefficients}$$
  - one prior cla probability $p(C_1)$
- As a total, it results into $d(d+1) + 2d + 1 = d(d+3) + 1$ coefficients (if a unique covariance matrix is assumed $d(d+1)/2 + 2d + 1 = d(d+5)/2 + 1$ coefficients)

Maximum likelihood estimation

Let us assume that targets of elements of the training set can be conditionally (with respect to model coefficients) modeled through a Bernoulli distribution. That is, assume

$$p(t_i|\mathbf{x}_i, \mathbf{w}) = p_i^{t_i}(1-p_i)^{1-t_i}$$

where $p_i = p(C_1|\mathbf{x}_i) = \sigma(\mathbf{w}^T\mathbf{x}_i)$.
Then, the likelihood of the training set targets $\mathbf{t}$ given $\mathbf{X}$ is

$$p(\mathbf{t}|\mathbf{X}, \mathbf{w}) = L(\mathbf{w}|\mathbf{X}, \mathbf{t}) = \prod_{i=1}^{n} p(t_i|\mathbf{x}_i, \mathbf{w}) = \prod_{i=1}^{n} p_i^{t_i}(1-p_i)^{1-t_i}$$

and the log-likelihood is

$$l(\mathbf{w}|\mathbf{X}, \mathbf{t}) = \log L(\mathbf{w}|\mathbf{X}, \mathbf{t}) = \sum_{i=1}^{n} (t_i \log p_i + (1-t_i)\log(1-p_i))$$

Maximum likelihood estimation

- Since

$$\frac{\partial l(\mathbf{w}|\mathbf{X},\mathbf{t})}{\partial \mathbf{w}} = \sum_{i=1}^{n} \frac{\partial l(\mathbf{w}|\mathbf{X},\mathbf{t})}{\partial p_i} \frac{\partial p_i}{\partial a_i} \frac{\partial a_i}{\partial \mathbf{w}} \qquad \frac{\partial l(\mathbf{w}|\mathbf{X},\mathbf{t})}{\partial p_i} = \frac{t_i}{p_i} - \frac{1-t_i}{1-p_i} = \frac{t_i - p_i}{p_i(1-p_i)}$$

$$\frac{\partial p_i}{\partial a_i} = \frac{\partial \sigma(a_i)}{\partial a_i} = \sigma(a_i)(1-\sigma(a_i)) = p_i(1-p_i)$$

$$\frac{\partial a_i}{\partial \mathbf{w}} = \overline{\mathbf{x}}_i$$

- it results,

$$\frac{\partial l(\mathbf{w}|\mathbf{X},\mathbf{t})}{\partial \mathbf{w}} = \sum_{i=1}^{n} (t_i - p_i)\overline{\mathbf{x}}_i = \sum_{i=1}^{n} (t_i - \sigma(\mathbf{w}^T \overline{\mathbf{x}}_i))\overline{\mathbf{x}}_i$$

Maximum likelihood estimation

To maximize the likelihood, we could apply a gradient ascent algorithm, where at each iteration the following update of the currently estimated $\mathbf{w}$ is performed

$$\begin{aligned}
\mathbf{w}^{(j+1)} &= \mathbf{w}^{(j)} + \alpha \frac{\partial l(\mathbf{w}|\mathbf{X}, \mathbf{t})}{\partial \mathbf{w}}\Big|_{\mathbf{w}^{(j)}} \\
&= \mathbf{w}^{(j)} + \alpha \sum_{i=1}^{n} (t_i - \sigma((\mathbf{w}^{(j)})^T \overline{\mathbf{x}}_i))\overline{\mathbf{x}}_i \\
&= \mathbf{w}^{(j)} + \alpha \sum_{i=1}^{n} (t_i - y(\mathbf{x}_i))\overline{\mathbf{x}}_i
\end{aligned}$$

Maximum likelihood estimation

As a possible alternative, at each iteration only one coefficient in $\mathbf{w}$ is updated

$$\begin{aligned}
w_k^{(j+1)} &= w_k^{(j)} + \alpha \frac{\partial l(\mathbf{w}|\mathbf{X},\mathbf{t})}{\partial w_k}\Big|_{\mathbf{w}^{(j)}} \\
&= w_k^{(j+1)} + \alpha \sum_{i=1}^{n} (t_i - \sigma((\mathbf{w}^{(j)})^T \overline{\mathbf{x}}_i)) x_{ik} \\
&= w_k^{(j+1)} + \alpha \sum_{i=1}^{n} (t_i - y(\mathbf{x}_i)) x_{ik}
\end{aligned}$$

Newton-Raphson method

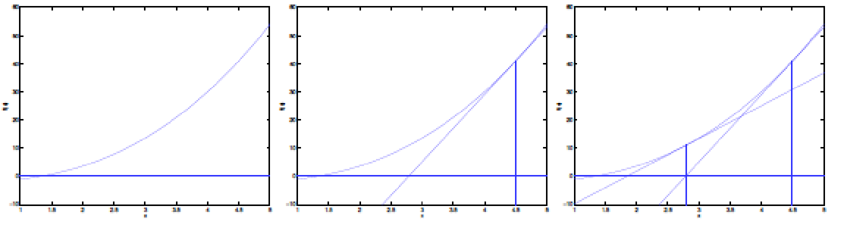
- Maximization of $l(\mathbf{w}|\mathbf{X}, \mathbf{t})$ through the well-known Newton-Raphson algorithm to compute the roots of a given function
- Given $f : \mathbb{R} \mapsto \mathbb{R}$, the algorithm finds $z \in \mathbb{R}$ such that $f(z) = 0$ through a sequence of iterations, starting from an initial value z_0 and performing the following update

$$z_{i+1} = z_i - \frac{f(z_i)}{f'(z_i)}$$

- At each iteration, the algorithm approximates f by a line tangent to f in $(z_i, f(z_i))$, and defines z_{i+1} as the value where the line intersects the x axis

Newton-Raphson method

- Example of application of the method



- Newton-Raphson method can be also applied to compute maximum and minimum points for a function by finding zeros of the first derivative: this corresponds to applying the following update

$$z_{i+1} = z_i - \frac{f'(z_i)}{f''(z_i)}$$

Newton-Raphson and multivariate functions

- To apply Newton-Raphson to logistic regression we have to extend it to the case of a vector variable, since the maximization has to be performed with respect to the vector $\mathbf{w}$ of coefficients
- In a multivariate framework, the first derivative is substituted by the gradient $\frac{\partial}{\partial \mathbf{w}}$, while the second derivative corresponds to the Hessian matrix $\mathbf{H}$, defined as follows

$$\mathbf{H}_{ij}(f) = \frac{\partial^2 f}{\partial w_i \partial w_j}$$

- The update operation turns out to be

$$\mathbf{w}^{(i+1)} = \mathbf{w}^{(i)} - \left(\mathbf{H}(f)|_{\mathbf{w}^{(i)}}\right)^{-1} \frac{\partial f}{\partial \mathbf{w}}\Big|_{\mathbf{w}_{(i)}}$$

Newton-Raphson and linear regression

- In the case of linear regression, the error function to be minimized is

$$E(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{n}(\mathbf{w}^T\overline{\mathbf{x}}_i - t_i)^2$$

- Then,

$$\frac{\partial E}{\partial \mathbf{w}} = \sum_{i=1}^{n}(\mathbf{w}^T\overline{\mathbf{x}}_i - t_i)\overline{\mathbf{x}}_i = \overline{\mathbf{X}}^T\overline{\mathbf{X}}\mathbf{w} - \overline{\mathbf{X}}^T\mathbf{t}$$

$$\mathbf{H} = \frac{\partial}{\partial \mathbf{w}}\frac{\partial E}{\partial \mathbf{w}} = \sum_{i=1}^{n}\overline{\mathbf{x}}_i\overline{\mathbf{x}}_i^T = \overline{\mathbf{X}}^T\overline{\mathbf{X}}$$

- At each iteration, the update is

$$\mathbf{w}^{(i+1)} = \mathbf{w}^{(i)} - (\overline{\mathbf{X}}^T\overline{\mathbf{X}})^{-1}(\overline{\mathbf{X}}^T\overline{\mathbf{X}}\mathbf{w}^{(i)} - \overline{\mathbf{X}}^T\mathbf{t}) = (\overline{\mathbf{X}}^T\overline{\mathbf{X}})^{-1}\overline{\mathbf{X}}^T\mathbf{t}$$

- We get the well-known solution, which is obtained in a single iteration.

Newton-Raphson and logistic regression

Here, we have

$$E(\mathbf{w}) = -l(\mathbf{w}|\mathbf{X}, \mathbf{t}) = -\sum_{i=1}^{n}\left(t_i \ln \sigma(\mathbf{w}^T\overline{\mathbf{x}}_i) + (1-t_i)\ln(1-\sigma(\mathbf{w}^T\overline{\mathbf{x}}_i))\right)$$

(this is called cross-entropy function). Hence,

$$\frac{\partial E}{\partial \mathbf{w}} = -\frac{\partial l(\mathbf{w}|\mathbf{X}, \mathbf{t})}{\partial \mathbf{w}} = \sum_{i=1}^{n}(\sigma(\mathbf{w}^T\overline{\mathbf{x}}_i) - t_i)\overline{\mathbf{x}}_i = \overline{\mathbf{X}}^T(\mathbf{s}_{\mathbf{w}} - \mathbf{t})$$

$$\mathbf{H} = \frac{\partial}{\partial \mathbf{w}}\frac{\partial E}{\partial \mathbf{w}} = \sum_{i=1}^{n}\sigma(\mathbf{w}^T\overline{\mathbf{x}}_i)(1-\sigma(\mathbf{w}^T\overline{\mathbf{x}}_i))\overline{\mathbf{x}}_i\overline{\mathbf{x}}_i^T = \overline{\mathbf{X}}^T\mathbf{R}_{\mathbf{w}}\overline{\mathbf{X}}$$

where

- $\mathbf{s}_{\mathbf{w}}$ is a vector of size n such that $\mathbf{s}_{\mathbf{w}i} = y(\mathbf{x}_i) = \sigma(\mathbf{w}^T\overline{\mathbf{x}}_i)$ for $i = 1, \ldots, n$
- $\mathbf{R}_{\mathbf{w}}$ is a $n \times n$ diagonal matrix such that

$$\mathbf{R}_{\mathbf{w}ii} = y(\mathbf{x}_i)(1-y(\mathbf{x}_i)) = \sigma(\mathbf{w}^T\overline{\mathbf{x}}_i)(1-\sigma(\mathbf{w}^T\overline{\mathbf{x}}_i)) = \mathbf{s}_{\mathbf{w}i}(1-\mathbf{s}_{\mathbf{w}i})$$

Newton-Raphson and logistic regression

- In the case of logistic regression, the update is then

$$\begin{aligned}\mathbf{w}^{(i+1)} &= \mathbf{w}^{(i)} - (\overline{\mathbf{X}}^T\mathbf{R}_{\mathbf{w}^{(i)}}\overline{\mathbf{X}})^{-1}\overline{\mathbf{X}}^T(\mathbf{s}_{\mathbf{w}^{(i)}} - \mathbf{t})\\ &= (\overline{\mathbf{X}}^T\mathbf{R}_{\mathbf{w}^{(i)}}\overline{\mathbf{X}})^{-1}((\overline{\mathbf{X}}^T\mathbf{R}_{\mathbf{w}^{(i)}}\overline{\mathbf{X}})\mathbf{w}^{(i)} - \overline{\mathbf{X}}^T(\mathbf{s}_{\mathbf{w}^{(i)}} - \mathbf{t}))\\ &= (\overline{\mathbf{X}}^T\mathbf{R}_{\mathbf{w}^{(i)}}\overline{\mathbf{X}})^{-1}\overline{\mathbf{X}}^T\mathbf{R}_{\mathbf{w}^{(i)}}\mathbf{z}_{\mathbf{w}^{(i)}}\end{aligned}$$

  where $\mathbf{z}_{\mathbf{w}^{(i)}}$ is a vector of size n defined as

$$\mathbf{z}_{\mathbf{w}^{(i)}} = \overline{\mathbf{X}}\mathbf{w}^{(i)} - \mathbf{R}_{\mathbf{w}^{(i)}}^{-1}(\mathbf{s}_{\mathbf{w}^{(i)}} - \mathbf{t})$$

  As it can be seen, $\mathbf{z}_{\mathbf{w}^{(i)}}$ is a function of $\mathbf{w}^{(i)}$, hence of the step i.

Iterated reweighted least squares

- Let us consider the weighted extension of the least squares cost function, denoted as weighted least squares cost function, defined as

$$\sum_{i=1}^{n} \psi_i (\mathbf{w}^T \overline{\mathbf{x}}_i - t_i)^2$$

  for given weights $\psi_1, \ldots, \psi_n$. Clearly, the least squares problems corresponds to the case $\psi_i = 1$ for $i = 1, \ldots, n$

- It can be proved that, for this problem, the optimum is

$$\mathbf{w} = (\overline{\mathbf{X}}^T \mathbf{\Psi} \overline{\mathbf{X}})^{-1} \overline{\mathbf{X}}^T \mathbf{\Psi} \mathbf{t}$$

  where $\mathbf{\Psi}$ is a diagonal matrix such that $\mathbf{\Psi}_{ii} = \psi_i$

Iterated reweighted least squares

- Let us remind that, at each step of NR algorithm applied to logistic regression, the following update is performed

$$\mathbf{w}^{(k+1)} = (\overline{\mathbf{X}}^T\mathbf{R}_{\mathbf{w}^{(k)}}\overline{\mathbf{X}})^{-1}\overline{\mathbf{X}}^T\mathbf{R}_{\mathbf{w}^{(k)}}\mathbf{z}_{\mathbf{w}^{(k)}}$$

- This corresponds to optimizing the weighted least squares cost function for feature matrix $\mathbf{X}$, target vector $\tilde{\mathbf{t}} = \overline{\mathbf{X}}\mathbf{w}^{(i)} - \mathbf{R}^{-1}_{\mathbf{w}^{(i)}}(\mathbf{s}_{\mathbf{w}^{(i)}} - \mathbf{t})$, and weights $\psi_k = \sigma((\mathbf{w}^{(k)})^T\overline{\mathbf{x}}_i)(1 - \sigma((\mathbf{w}^{(k)})\overline{\mathbf{x}}_i))$
- The update of $\mathbf{w}^{(i)}$ performed at each iteration can then be computed by solving a new instance of the weighted least square problem, setting $\mathbf{w}^{(i+1)}$ to the solution obtained, and deriving the new values of $\mathbf{\Psi} = \mathbf{R}_{\mathbf{w}^{(i+1)}}$ and $\tilde{t} = \mathbf{z}_{\mathbf{w}^{(i+1)}}$.

Logistic regression and GDA

- Observe that assuming $p(\mathbf{x}|C_1)$ are $p(\mathbf{x}|C_2)$ as multivariate normal distributions with same covariance matrix $\mathbf{\Sigma}$ results into a logistic $p(C_1|\mathbf{x})$.
- The opposite, however, is not true in general: in fact, GDA relies on stronger assumptions than logistic regression.
- The more the normality hypothesis of class conditional distributions with same covariance is verified, the more GDA will tend to provide the best models for $p(C_1|\mathbf{x})$

Logistic regression and GDA

- Logistic regression relies on weaker assumptions than GDA: it is then less sensible from a limited correctness of such assumptions, thus resulting in a more robust technique
- Since $p(C_i|\mathbf{x})$ is logistic under a wide set of hypotheses about $p(\mathbf{x}|C_i)$, it will usually provide better solutions (models) in all such cases, while GDA will provide poorer models as far as the normality hypotheses is less verified.

Softmax regression

- In order to extend the logistic regression approach to the case $K > 2$, let us consider the vector $\mathbf{w} = (\mathbf{w}_1, \ldots, \mathbf{w}_K)$ of model coefficients, of size $(d+1)K$, where the j-th block of $\mathbf{w}$ $(j = 1, \ldots, K)$ corresponds to the vector $\mathbf{w}_j$ of the $d+1$ coefficients for class C_j.
- In this case, the likelihood is defined as

$$
\begin{aligned}
p(\mathbf{T}|\mathbf{X}, \mathbf{w}) &= \prod_{i=1}^{n} \prod_{k=1}^{K} p(C_k|\mathbf{x}_i)^{t_{ik}} \\
&= \prod_{i=1}^{n} \prod_{k=1}^{K} \left(\frac{e^{\mathbf{w}_k^T \overline{\mathbf{x}}_i}}{\sum_{r=1}^{K} e^{\mathbf{w}_r^T \overline{\mathbf{x}}_i}} \right)^{t_{ik}}
\end{aligned}
$$

where $\mathbf{X}$ is the usual matrix of features and $\mathbf{T}$ is an $n \times K$ matrix such that the i-th row of $\mathbf{T}$ is the 1-to-K coding of t_i. That is, if $\mathbf{x}_i \in C_k$ then $t_{ik} = 1$ and $t_{ir} = 0$ for $r \neq k$.

ML and softmax regression

The log-likelihood is then defined as

$$l(\mathbf{w}) = \sum_{i=1}^{n}\sum_{k=1}^{K} t_{ik} \log\left(\frac{e^{\mathbf{w}_k^T\overline{\mathbf{x}}_i}}{\sum_{r=1}^{K} e^{\mathbf{w}_r^T\overline{\mathbf{x}}_i}}\right)$$

The gradient is the vector of size $(d+1)K$ defined as

$$\frac{\partial l(\mathbf{w})}{\partial \mathbf{w}} = \left(\frac{\partial l(\mathbf{w})}{\partial \mathbf{w}_1}, \ldots, \frac{\partial l(\mathbf{w})}{\partial \mathbf{w}_K}\right)$$

ML and softmax regression

- To derive $\frac{\partial l(\mathbf{w})}{\partial \mathbf{w}_j}$ let

$$y_{ik} = \frac{e^{a_{ik}}}{\sum_{r=1}^{K} e^{a_{ir}}} \qquad \text{with} \qquad a_{ik} = \mathbf{w}_k^T \overline{\mathbf{x}}_i$$

for $k = 1, \ldots, K$ and $i = 1, \ldots, n$. Then,

$$l(\mathbf{w}) = \sum_{i=1}^{n} \sum_{k=1}^{K} t_{ik} \log y_{ik}$$

- For each $i = 1, \ldots, n$, $j = 0, \ldots, d$, $k = 1, \ldots, K$,

$$\frac{\partial a_{ik}}{\partial w_{kj}} = \frac{\partial}{\partial w_{kj}} \mathbf{w}_k^T \overline{\mathbf{x}}_i = x_{ik} \qquad \frac{\partial y_{ik}}{\partial a_{ik}} = y_{ik}(1 - y_{ik}) \qquad \frac{\partial y_{ik}}{\partial a_{ir}} = -y_{ir} y_{ik} \qquad \text{if } r \neq k$$

ML and softmax regression

Hence,

$$
\begin{aligned}
\frac{\partial l(\mathbf{w})}{\partial \mathbf{w}_j} &= \frac{\partial}{\partial \mathbf{w}_j}\sum_{k=1}^{K}\sum_{i=1}^{n} t_{ik}\log y_{ik} = \frac{\partial}{\partial \mathbf{w}_j}\sum_{i=1}^{n} t_{ij}\log y_{ij} + \frac{\partial}{\partial \mathbf{w}_j}\sum_{k\neq j}\sum_{i=1}^{n} t_{ik}\log y_{ik}\\
&= \sum_{i=1}^{n} t_{ij}\frac{1}{y_{ij}}\frac{\partial y_{ij}}{\partial a_{ij}}\frac{\partial a_{ij}}{\partial \mathbf{w}_j} + \sum_{k\neq j}\sum_{i=1}^{n} t_{ik}\frac{1}{y_{ik}}\frac{\partial y_{ik}}{\partial a_{ik}}\frac{\partial a_{ik}}{\partial \mathbf{w}_j}\\
&= \sum_{i=1}^{n} t_{ij}\frac{1}{y_{ij}} y_{ij}(1-y_{ij})\overline{\mathbf{x}}_i - \sum_{k\neq j}\sum_{i=1}^{n} t_{ik}\frac{1}{y_{ik}} y_{ik}y_{ij}\overline{\mathbf{x}}_i\\
&= \left(\sum_{i=1}^{n} t_{ij} - \sum_{i=1}^{n} y_{ij}\sum_{k=1}^{K} t_{ik}\right)\overline{\mathbf{x}}_i = \left(\sum_{i=1}^{n} t_{ij} - \sum_{i=1}^{n} y_{ij}\right)\overline{\mathbf{x}}_i = \sum_{i=1}^{n}(t_{ij}-y_{ij})\overline{\mathbf{x}}_i
\end{aligned}
$$

Observe that the gradient has the same structure than in the case of linear regression and logistic regression.

Bayesian logistic regression

- Used to overcome the overfitting problem by assuming a prior distribution
- The aim is to estimate the posterior class (predictive) distribution, that is the expectation of the model prediction wrt to the distribution of model coefficients,

$$\begin{aligned} p(\mathcal{C}_1|\mathbf{x}, \mathbf{X}, \mathbf{t}) &= \int p(\mathcal{C}_1|\mathbf{x}, \mathbf{w}) p(\mathbf{w}|\mathbf{X}, \mathbf{t}) d\mathbf{w} \\ &= \int \sigma(\mathbf{w}^T \boldsymbol{\phi}(\mathbf{x})) p(\mathbf{w}|\mathbf{X}, \mathbf{t}) d\mathbf{w} \end{aligned}$$

- we need some way to evaluate the posterior distribution of coefficients $p(\mathbf{w}|\mathbf{X}, \mathbf{t})$ for any $\mathbf{w}$

Posterior distribution of coefficients

By Bayes' rule,

$$p(\mathbf{w}|\mathbf{X},\mathbf{t}) = \frac{p(\mathbf{t}|\mathbf{X},\mathbf{w})p(\mathbf{w})}{p(\mathbf{t}|\mathbf{X})} = \frac{p(\mathbf{t}|\mathbf{X},\mathbf{w})p(\mathbf{w})}{\int p(\mathbf{t}|\mathbf{X},\mathbf{w}')p(\mathbf{w}')d\mathbf{w}'}$$

where the likelihood is $p(\mathbf{t}|\mathbf{X},\mathbf{w}) = \prod_{i=1}^{n} p(t_i|\mathbf{x}_i,\mathbf{w})$, with

$$p(t_i|\mathbf{x}_i,\mathbf{w}) = \begin{cases} \sigma(\mathbf{w}^T\overline{\mathbf{x}}) & \text{if } t_i = 1 \\ 1-\sigma(\mathbf{w}^T\overline{\mathbf{x}}) & \text{if } t_i = 0 \end{cases}$$

Posterior distribution of coefficients

That is,

$$p(\mathbf{t}|\mathbf{X},\mathbf{w}) = \prod_{i=1}^{n} \sigma(\mathbf{w}^T\overline{\mathbf{x}})^{t_i}\left(1-\sigma(\mathbf{w}^T\overline{\mathbf{x}})\right)^{1-t_i}$$

and

$$p(\mathbf{w}|\mathbf{X},\mathbf{t}) = \frac{p(\mathbf{w})\prod_{i=1}^{n} \sigma(\mathbf{w}^T\overline{\mathbf{x}})^{t_i}\left(1-\sigma(\mathbf{w}^T\overline{\mathbf{x}})\right)^{1-t_i}}{Z}$$

with the normalization factor

$$Z = \int p(\mathbf{w})\prod_{i=1}^{n} \sigma(\mathbf{w}^T\overline{\mathbf{x}})^{t_i}\left(1-\sigma(\mathbf{w}^T\overline{\mathbf{x}})\right)^{1-t_i}\,d\mathbf{w}$$

Predictive distribution intractability

Z is hard to compute: we are only able to evaluate the numerator

$$g(\mathbf{w};\mathbf{X},\mathbf{t}) = p(\mathbf{w})\prod_{i=1}^{n}\sigma(\mathbf{w}^T\overline{\mathbf{x}})^{t_i}\left(1-\sigma(\mathbf{w}^T\overline{\mathbf{x}})\right)^{1-t_i}$$

which is proportional to $p(\mathbf{w}|\mathbf{X},\mathbf{t})$ through an unknown proportionality coefficient.

Predictive distribution intractability

Possible options:

1. find a single value of $\mathbf{w}$ which maximizes $p(\mathbf{w}|\mathbf{X}, \mathbf{t})$: this corresponds to the value which maximizes $g(\mathbf{w}; \mathbf{X}, \mathbf{t})$ (this is the usual MAP approach)
2. approximate $p(\mathbf{w}|\mathbf{X}, \mathbf{t})$ with some other probability density which can be treated analytically (*variational* approach)
3. sample from $p(\mathbf{w}|\mathbf{X}, \mathbf{t})$, knowing only $g(\mathbf{w}; \mathbf{X}, \mathbf{t})$ (*Montecarlo* approach)