4 Linear Models for Classification

Discusses linear models and *generalised linear models* (GLM). GLM means that even if the prediction functions are non-linear, the decision surfaces are linear.

4.1 Discriminant functions

4.1.1 Two Classes

Describes the geometry of a discriminant function $y(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + w_0$. That is, $\mathbf{w}$'s are orthogonal to decision surface and $|w_0|/||\mathbf{w}||$ describes dislocation from origin.

4.1.2 Multiple Classes

Discuss the limitation of one-vs-rest and one-vs-one classifiers, introduce the benfits of multi-class linear discriminant.

4.1.3 Least-squares Classification

Least squares classification has one extra limitation wrt. limitation of least squares regression: The target vector $\mathbf{t}$ are of 1-of-K type.

4.1.4 Fisher's Linear Discriminant

Perform a dimensionality reduction and then discrimination. $J(\mathbf{w})$ is a function that does this and can be minimised via (4.2.9).

4.1.5 Relation to Least-squares

By changing the target variable representation for the 2-class problem, it's possible to relate Fisher and least-squares.

4.1.6 Fisher's Discriminant for multiclass

Consider generalisation to $K > 2$ classes. The extension is similar to 2-class. Now there are multiple possible choices of (Fisher) criterion.

4.1.7 The Perceptron Algorithm

Construct GLM $y(\mathbf{x}) = f(\mathbf{w}^T\phi(\mathbf{x}))$ where $f(a) = \begin{cases} +1 & a \geq 0 \\ -1 & a < 0 \end{cases}$. Patterns in C_1 become $+1$ and for $x_n \in C_1$ we want $\mathbf{w}^T\phi(\mathbf{x}) > 0$ and for $x \in C_2$ we want it to be < 0. Both can be summarised as $t\mathbf{w}^T\phi(\mathbf{x}) > 0$.

The perceptron criterion minimises error only on misclassified patterns. The weight update algorDenemeithm operates for each sample n:

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

where η is the *learning rate*. The update $(\tau + 1)$ happens in the *direction of misclassification* and *guarantees* the error on misclassified sample to be reduced. Of course it doesn't guarantee anything on *all* training samples.

4.2 Probabilistic Generative Models

Construct posterior $p(C_k|\mathbf{x})$ and represent via *logistic sigmoid*:

$$p(C_1|\mathbf{x}) = \frac{p(\mathbf{x}|C_1)p(C_1)}{\sum\limits_{j\in\{1,2\}} p(\mathbf{x}|C_j)p(C_j)} = \frac{1}{1+\exp(-\alpha)} = \sigma(\alpha) \tag{2}$$

where $\alpha = \ln\frac{p(\mathbf{x}|C_1)p(C_1)}{p(\mathbf{x}|C_2)p(C_2)}$ and $\sigma(\alpha)$ is the logistic sigmoid function.

We are interested in situations where $\alpha(\mathbf{x})$ is linear and therefore creates posteriors governed by GLMs.

4.2.1 Continuous Inputs

We sgtart by assuming that all classes C_k share same cov matrix Σ.

For K classes α_k becomes $\alpha_k(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + w_{k0}$ where $\mathbf{w}_k = \Sigma^{-1}\boldsymbol{\mu}_k$ and w_{k0} is as in (4.70).

That is, α_k is linear in $\mathbf{x}$. Decision boundaries (which correspond to misclassification rate) wioll be again linear in $\mathbf{x}$ so again we have GLM.

If we relax the "shared covariance matrix" assumption, then we'll have *quadratic discriminant* rather than GLM.

4.2.2 Maximum likelihood solution

Once $p(\mathbf{x}|C_k)$ defined, we can determine values of its parameters and parameters of $p(C_k)$ via *maximum likelihood*. Construct maximum function:

$$p(\mathbf{T}, \mathbf{X}|\pi, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \Sigma) = \prod_n [\pi\mathcal{N}(x_n|\boldsymbol{\mu}_1, \Sigma)]^{t_n}[(1-\pi)\mathcal{N}(x_n|\boldsymbol{\mu}_2, \Sigma)]^{(1-t_n)} \tag{3}$$

In the ML solution we get $\boldsymbol{\mu} = \frac{1}{N_1}\sum_n t_n\mathbf{x}_n$ and $\boldsymbol{\mu} = \frac{1}{N_2}\sum_n (1-t_n)\mathbf{x}_n$.

For covariance Σ, define $\mathbf{S}_1, \mathbf{S}_2, \mathbf{S}$ as:

$$\mathbf{S}_1 = \frac{1}{N_1}\sum_{n\in C_1}(\mathbf{x}_n - \boldsymbol{\mu}_1)(\mathbf{x}_n - \boldsymbol{\mu}_1)^T \tag{4}$$

$$\mathbf{S}_2 = \frac{1}{N_2}\sum_{n\in C_2}(\mathbf{x}_n - \boldsymbol{\mu}_2)(\mathbf{x}_n - \boldsymbol{\mu}_2)^T \tag{5}$$

$$\mathbf{S} = \frac{N_1}{N}\mathbf{S}_1 + \frac{N_2}{N}\mathbf{S}_2 \tag{6}$$

Overall, process not robust to outliers because ML is not.

4.2.3 Discrete Features

4.2.4 Exponential Family

We manage to get GLMs for the above types too.

4.3 Probabilistic Discriminative Models

Advantage: There are less parameters to discover and usually leads to improved performance.

4.3.1 Fixed Basis Functions

4.3.2 Logistic Regression

Here we set M params whereas in generative modelling we set $(M+5)/2+1$ params.

Consider implementing a discriminative function directly as a via logistic sigmoid function:

$$p(C_1|\phi) = y(\phi) = \sigma(\mathbf{w}^T\phi) \tag{7}$$

and naturally $p(C_2|\phi) = 1 - p(C_1|\phi)$. We can set params via ML. We start by seeing that $\frac{d\sigma}{d\alpha} = \sigma(1-\sigma)$ (exercise 4.12). Likelihood can be written as:

$$p(\mathbf{T}|\mathbf{w}) = \prod_n y_n^{t_n}(1-y_n)^{1-t_n} \tag{8}$$

cross entropy error

where $y_n = p(C_1|\phi_n)$. Error function here is also called cross entropy error:

$$E(\mathbf{w}) = -\ln p(\mathbf{T}|\mathbf{w}) = -\sum_n [t_n \ln y_n + (1-t_n)\ln(1-y_n)] \tag{9}$$

Taking the gradient wrt $\mathbf{w}$:

$$\nabla E(\mathbf{w}) = \sum_n (y_n - t_n)\phi_n \tag{10}$$

4.3.3 Iterative Reweighted Least Squares

We no longer have closed-form solution (as we did for regression). Fortunately the error function is still convex there is the (iterative) Newton-Raphson or iterative reweighted least squares algorithm:

Newton-Raphson or iterative reweighted least squares

$$\mathbf{w}^{(new)} = \mathbf{w}^{(old)} = \mathbf{H}^{-1}\nabla E(\mathbf{w}) \tag{11}$$

where $\mathbf{H}$ is the hassian matrix whose elements comprise the second derivs of $E(\mathbf{w})$ wrt components of $\mathbf{w}$.

$$\nabla E(\mathbf{w}) = \sum_n (y_n - t_n)\phi_n = \mathbf{\Phi}^T(\mathbf{Y} - \mathbf{T}) \tag{12}$$

$$\mathbf{H} = \nabla\nabla E(\mathbf{w}) = \sum_n y_n(1 - y_n)\phi_n\phi_n^T = \mathbf{\Phi}^T\mathbf{R}\mathbf{\Phi} \tag{13}$$

design matrix

where $\mathbf{\Phi}$ is the $N \times M$ design matrix whose nth row is given by ϕ_n^T and $\mathbf{R}$ is the $N \times N$ diagonal matrix with elements $\mathbf{R}_{nn} = y_n(1 - y_n)$.

4.3.4 Multiclass logistic regression

The formalism is similar to 2-class logistic regression. Instead of sigmoid we use the *softmax* function. Again we have *cross-entropy* function as error function. The multiclass version of cross-entropy is:

$$E(\mathbf{w}_1, \ldots, \mathbf{w}_K) = -lnp(\mathbf{T}|\mathbf{w}_1, \ldots, \mathbf{w}_K) = -\sum_n \sum_k t_{nk} - \ln y_{nk}. \tag{14}$$

Again we can use *iterative reweighted least squares* (#210).

4.3.5 Probit regression

The inverse probit function (or the similar erf function) are similar to sigmoid in shape but have more plausible analytical properties. Will be discussed in Sec. 4.5.

4.3.6 Canonical link function

This is one of the most frequently-referred sections of the book. The choices of sigmoid/softmax in earlier sections were not arbitrary — they were chosen to convert the error function to a simple form that involves $y_n - t_n$. This is a general result of assuming a conditional distribution for the activation function known as the canonical link function.

canonical link function

A GLM is a model for which y is a nonlinear function of a linear combination of input variables:

$$y = f(\mathbf{w}^T\phi) \tag{15}$$

where $f(\cdot)$ is the *activation function* and $f^{-1}(\cdot)$ is known as the *link function*.

Let the conditional distro be $p(\mathbf{T}|\eta, s)$. We formulate its derivative in the following form:

$$\nabla_{\mathbf{w}} \ln p(\mathbf{T}|\eta, s) = \ldots (\text{see } \#213) = \sum_n \frac{1}{s}\psi'(y_n)f'(y_n)\phi_n \tag{16}$$

. The canonical link function chosen as $f^{-1}(y) = \psi(y)$ provides a great simplification:

$$\nabla E(\mathbf{w}) = \frac{1}{s}\sum_n (y_n - t_n)\phi_n \tag{17}$$

4.4 The Laplace Approximation

To perform closed-form analysis for Bayesian logistic regression, we'll need to do approximation. The Laplace approx. is used for this purpose. Approximation is performed by matching the *mode* of the target distribution with the mode of a Gaussian via Taylor expansion (where the first-order term disappears as expansion is made around a local maximum). Let $\mathbf{z}_0$ be the mode of the target distribution. The 2^{nd} order Taylor expansion around $\mathbf{z}_0$ is:

$$f(\mathbf{z}) \approx \ln f(\mathbf{z}_0) - \frac{1}{2}(\mathbf{z} - \mathbf{z}_0)^T\mathbf{A}(\mathbf{z} - \mathbf{z}_0). \tag{18}$$

This will enable us to compute the approximated distribtion $q(\mathbf{z})$ directly as $q(\mathbf{z}) = \mathcal{N}(\mathbf{z}|\mathbf{z}_0, \mathbf{A})$.

Better methods will be explored in Chapter 10

Better methods will be explored in Chapter 10.

4.4.1 Model comparison and BIC

We can use the approximation above for model comparison, which will lead to Bayesian Information Criterion (BIC). Start with the normalisation term:

$$Z \approx f(\mathbf{z}_0) \int \exp\left[-\frac{1}{2}(\mathbf{z}-\mathbf{z}_0)^T \mathbf{A}(\mathbf{z}-\mathbf{z}_0)\right] d\mathbf{z} = f(\mathbf{z}_0)\frac{(2\pi)^{M/2}}{|\mathbf{A}|^{1/2}}. \tag{19}$$

Consider data set $\mathcal{D}$ and models $\{\mathcal{M}_i\}$ with parameters $\{\boldsymbol{\theta}_i\}$. For each model we define a likelihood function $p(\mathcal{D}|\boldsymbol{\theta}_i, \mathcal{M}_i)$ — or shortly, $p(\mathcal{D}|\boldsymbol{\theta}_i)$.

Defining $f(\boldsymbol{\theta}) = p(\mathcal{D}|\boldsymbol{\theta})p(\theta)$ and identifying that $Z = p(\mathcal{D})$, we can apply the result above to get:

$$\ln p(\mathcal{D}) \approx \ln p(\mathcal{D}|\boldsymbol{\theta}_{\mathrm{MAP}}) + \overbrace{\ln p(\boldsymbol{\theta}_{\mathrm{MAP}}) + \frac{M}{2}\ln(2\pi) - \frac{1}{2}\ln|\mathbf{A}|}^{\text{Occam factor}}. \tag{20}$$

With further simplifications via (not necessarily realistic) assumptions (see #217) we get the BIC:

$$\ln p(\mathcal{D}) \approx \ln p(\mathcal{D}|\boldsymbol{\theta}_{\mathrm{MAP}}) - \frac{1}{2}M \ln N \tag{21}$$

Essentially this is an information criterion that penalizes model complexity

Miscellaneous

Model Comparison The more rigorous section is Sec. 3.4 (and 3.5) with a proper treatment of a theoretically plausible model selection approach. AIC (see 1.73) and BIC (Sec 4.4.1, #217) offer simpler model comparison criteria.