Probabilistic discriminative models

Linear models for classification

Francesco Corona

Probabilistic discriminative models

Linear models for classification

Probabilistic discriminative models

For the two-class classification problem, the posterior probability of class $\mathcal{C}_1$ can be written as a logistic sigmoid acting on a linear function of $\mathbf{x}$

$$p(\mathcal{C}_1|\mathbf{x}) = \sigma\left(\ln \frac{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1)}{p(\mathbf{x}|\mathcal{C}_2)p(\mathcal{C}_2)}\right) = \sigma(\mathbf{w}^T\mathbf{x} + w_0)$$

- for a wide choice of class-conditional distributions $p(\mathbf{x}|\mathcal{C}_k)$

For the multi-class case, the posterior probability of class $\mathcal{C}_k$ is given by a softmax transformation of a linear function of $\mathbf{x}$

$$p(\mathcal{C}_k|\mathbf{x}) = \frac{p(\mathbf{x}|\mathcal{C}_k)p(\mathcal{C}_k)}{\sum_{j=1}^{K} p(\mathbf{x}|\mathcal{C}_j)p(\mathcal{C}_j)} = \frac{\exp\left(\mathbf{w}_k^T\mathbf{x} + w_{k0}\right)}{\sum_{j=1}^{K} \exp\left(\mathbf{w}_j^T\mathbf{x} + w_{j0}\right)}$$

Probabilistic discriminative models (cont.)

For specific choices of class-conditionals $p(\mathbf{x}|\mathcal{C}_k)$, maximum likelihood can be used to determine the parameters of the densities and the class priors $p(\mathcal{C}_k)$

- Bayes' theorem is then used to find posterior class probabilities $p(\mathcal{C}_k\mathbf{x})$

An alternative approach is to use the functional form of the generalised linear model explicitly and determine its parameters directly by maximum likelihood

- There is an efficient algorithm finding such solutions
- **Iterative re-weighted least squares**, IRLS

Probabilistic discriminative models (cont.)

The indirect approach to find parameters of a generalised linear model, by fitting class-conditional densities and class priors separately and then by applying Bayes' theorem, represents an example of generative modelling

- We could take such a model and generate synthetic data by drawing values of $\mathbf{x}$ from the marginal distribution $p(\mathbf{x})$

In the direct approach, we maximise a likelihood function defined through the conditional distribution $p(\mathcal{C}_k|\mathbf{x})$, this is a form of discriminative training

- One advantage of the discriminative approach is that there will typically be fewer adaptive parameters to be determined
- It may also lead to improved predictive performance, particularly when the class-conditional density assumptions give a poor approximation to the true distributions

Outline

Fixed basis functions
Probabilistic discriminative models

Fixed basis functions

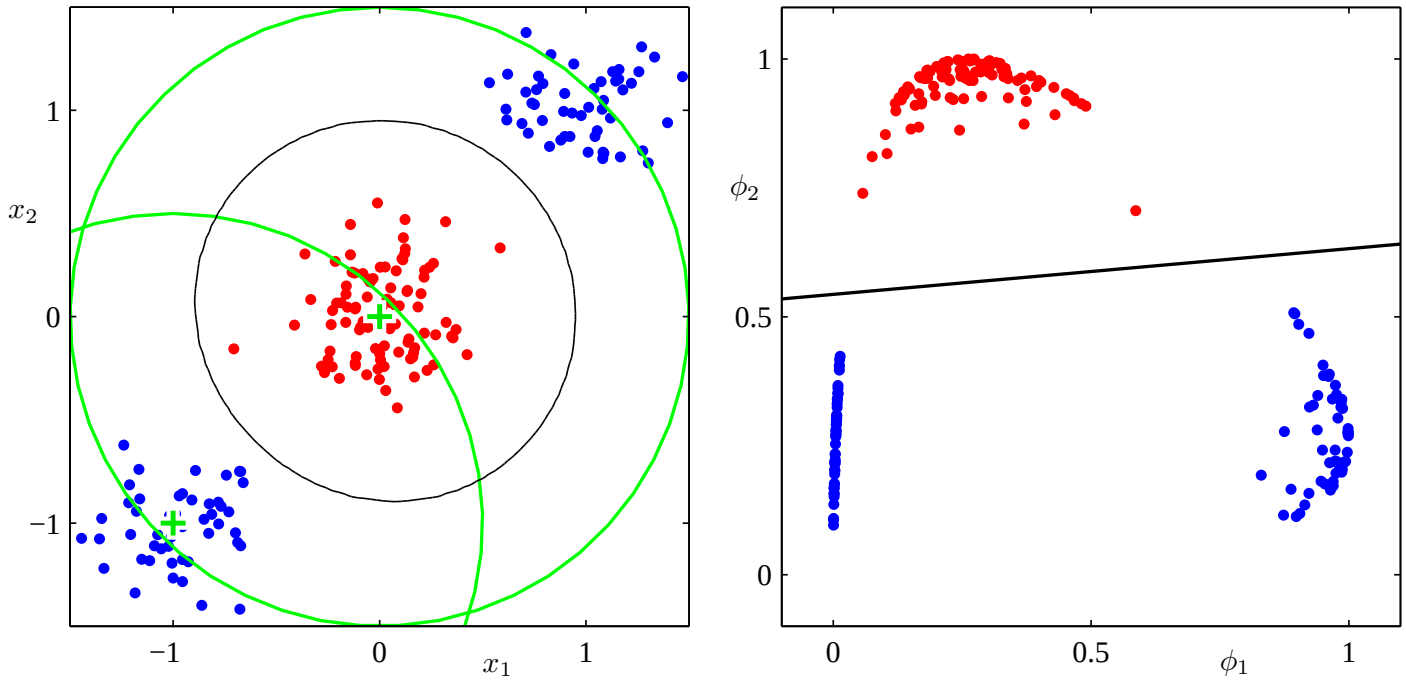
We considered classification models that work with the original input vector $\mathbf{x}$

However, all of the algorithms are equally applicable if we first make a fixed nonlinear transformation of the inputs using a vector of basis functions $\phi(\mathbf{x})$

The resulting decision boundaries will be linear in the feature space ϕ, and these correspond to nonlinear decision boundaries in the original $\mathbf{x}$ space

- Classes that are linearly separable in the feature space $\phi(\mathbf{x})$ need not be linearly separable in the original observation space $\mathbf{x}$

Fixed basis functions (cont.)



Original input space (x_1, x_2) together with points from two classes (red/blue)

- Two 'Gaussian' basis functions $\phi_1(\mathbf{x})$ and $\phi_2(\mathbf{x})$ are defined in this space with centres (green crosses) and with contours (green circles)

Feature space (ϕ_1, ϕ_2) together with the linear decision boundary (black line)

- Nonlinear decision boundary in the original input space (black curve)

Fixed basis functions (cont.)

Often, there is significant overlap between class-conditional densities $p(\mathbf{x}|\mathcal{C}_k)$

- This corresponds to posterior probabilities $p(\mathcal{C}_k|\mathbf{x})$, which are not 0 or 1
- At least, for some values of $\mathbf{x}$

In such cases, the optimal solution is obtained by modelling the posterior probabilities $p(\mathcal{C}_k|\mathbf{x})$ accurately and then applying standard decision theory

Note that nonlinear transformations $\phi(\mathbf{x})$ cannot remove such class overlap

- Indeed, they can increase the level of overlap, or even create overlap where none existed in the original observation space

However, suitable choices of nonlinearity can often make the process of modelling the posterior probabilities easier

Notwithstanding these limitations, models with fixed nonlinear basis functions play an important role

Logistic regression
Probabilistic discriminative models

Logistic regression

When considering the two-class problem using a generative approach and under general assumptions, the posterior probability of class $\mathcal{C}_1$ can be written as

- a logistic sigmoid on a linear function of the feature vector ϕ so that

$$p(\mathcal{C}_1|\phi) = y(\phi) = \sigma(\mathbf{w}^T\phi) \qquad \text{with } p(\mathcal{C}_2|\phi) = 1 - p(\mathcal{C}_1|\phi) \tag{1}$$

- The logistic sigmoid function is defined as

$$\sigma(a) = \frac{1}{1+\exp(-a)} \quad \text{with } a = \ln \frac{p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1)}{p(\mathbf{x}|\mathcal{C}_2)p(\mathcal{C}_2)}$$

In the terminology of statistics this model is known as **logistic regression**

- For an M-dimensional feature space ϕ, the model has M parameters

Logistic regression (cont.)

To fit Gaussian class conditional densities with maximum likelihood, we need

- $2M + M(M+1)/2$ parameters for means and (shared) covariance matrix

And a total of $M(M+5)/2+1$ parameters, if we include the class prior $p(\mathcal{C}_1)$

- The number of parameters grows quadratically with M

For the M parameters of logistic regression model, we use maximum likelihood

Logistic regression (cont.)

For data $\{\phi_n, t_n\}_{n=1}^{N}$ with $t_n = \{0, 1\}$ and $\phi_n = \phi(\mathbf{x}_n)$, the likelihood function

$$p(\mathrm{t}|\mathbf{w}) = \prod_{n=1}^{N} y_n^{t_n}(1 - y_n)^{1-t_n} \tag{2}$$

is written for $\mathrm{t} = (t_1, \ldots, t_N)^T$ and $y_n = p(\mathcal{C}_1|\phi_n)$

By taking the negative log of the likelihood, our error function is defined by

$$E(\mathbf{w}) = -\ln p(\mathrm{t}|\mathbf{w}) = -\sum_{n=1}^{N} \Big(t_n \ln(y_n) + (1 - t_n) \ln(1 - y_n) \Big) \tag{3}$$

which is the **cross-entropy error function** with $y_n = \sigma(a_n)$ and $a_n = \mathbf{w}^T \phi_n$

Logistic regression (cont.)

By taking the gradient of the error function with respect to $\mathbf{w}$, we get

$$\nabla E(\mathbf{w}) = \sum_{n=1}^{N} (y_n - t_n)\phi_n \tag{4}$$

The contribution to the gradient from point n comes from the error $(y_n - t_n)$ between target value and model prediction, times the basis function vector ϕ_n

- The gradient takes the same form as the gradient of the sum-of-squares error function for linear regression models

Logistic regression (cont.)

Maximum likelihood can show severe over-fitting for linearly separable data

- The solution occurs when the hyperplane for $\sigma = 0.5$, or $\mathbf{w}^T\phi = 0$, separates the two classes and the magnitude of $\mathbf{w}$ goes to infinity
- The logistic sigmoid becomes infinitely steep (Heaviside) in feature space, and every point from each class k gets a posterior probability $p(\mathcal{C}_k|\mathbf{x}) = 1$

There is also a continuum of such solutions because any separating hyperplane gives rise to the same posterior probabilities at the training data points

- Maximum likelihood does not favour one such solution over another
- The solution depends on the optimisation algorithm and initialisation

One possibility would be to introduce a prior over $\mathbf{w}$ and finding a MAP solution

- Add a regularisation term to the error function

Iterative reweighted least squares
Probabilistic discriminative models

Iterative reweighted least squares

In the case of the linear regression models, the maximum likelihood solution, on the assumption of a Gaussian noise model, leads to a closed-form solution

- A consequence of quadratic dependence of log likelihood function on $\mathbf{w}$

For logistic regression, due to the nonlinearity of the logistic sigmoid function

- There is no longer a closed-form solution
- Departure from quadratic is not substantial

Specifically, the error function is convex, and hence it has a unique minimum

Furthermore, the error function can be minimised by an efficient iterative technique based on the **Newton-Raphson** iterative optimisation scheme

- A local quadratic approximation to the log likelihood function

Iterative reweighted least squares (cont.)

The Newton-Raphson update, for minimising a function $E(\mathbf{w})$, takes the form

$$\mathbf{w}^{(\text{new})} = \mathbf{w}^{(\text{old})} - \mathbf{H}^{-1}\nabla E(\mathbf{w}) \tag{5}$$

$\mathbf{H}$ is the Hessian matrix, with elements the second derivatives of $E(\mathbf{w})$ wrt $\mathbf{w}$

We apply the Newton-Raphson method to

1. the sum-of-squares error function (linear regression model)

$$E_D(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\left(t_n - \mathbf{w}^T\phi(\mathbf{x}_n)\right)^2$$

2. the cross-entropy error function (logistic regression model)

$$E(\mathbf{w}) = -\ln p(\mathbf{t}|\mathbf{w}) = -\sum_{n=1}^{N}\left(t_n \ln(y_n) + (1-t_n)\ln(1-y_n)\right)$$

Iterative reweighted least squares (cont.)

Gradient and Hessian of the sum-of-squares error function are

$$\nabla E(\mathbf{w}) = \sum_{n=1}^{N} \left(\mathbf{w}^T \phi_n - t_n\right)\phi_n = \mathbf{\Phi}^T\mathbf{\Phi}\mathbf{w} - \mathbf{\Phi}^T\mathtt{t} \tag{6}$$

$$\mathbf{H} = \nabla\nabla E(\mathbf{w}) = \sum_{n=1}^{N} \phi_n \phi^T = \mathbf{\Phi}^T\mathbf{\Phi} \tag{7}$$

where $\mathbf{\Phi}$ is the $N \times N$ design matrix with ϕ_n^T in the n-th row

The Newton-Raphson update takes the form

$$\begin{aligned} \mathbf{w}^{(\text{new})} &= \mathbf{w}^{(\text{old})} - (\mathbf{\Phi}^T\mathbf{\Phi})^{-1}(\mathbf{\Phi}^T\mathbf{\Phi}\mathbf{w}^{(\text{old})} - \mathbf{\Phi}^T\mathtt{t}) \\ &= (\mathbf{\Phi}^T\mathbf{\Phi})^{-1}\mathbf{\Phi}^T\mathtt{t} \end{aligned} \tag{8}$$

which is the classical least-squares solution

The error function is quadratic, N-R formula gets the exact solution in one step

Iterative reweighted least squares (cont.)

Gradient and Hessian of the cross-entropy error function are

$$\nabla E(\mathbf{w}) = \sum_{n=1}^{N} (y_n - t_n)\phi_n = \mathbf{\Phi}^T(\mathrm{y} - \mathrm{t}) \tag{9}$$

$$\mathbf{H} = \nabla\nabla E(\mathbf{w}) = \sum_{n=1}^{N} y_n(1 - y_n\phi_n\phi_n^T) = \mathbf{\Phi}^T\mathbf{R}\mathbf{\Phi} \tag{10}$$

where $\mathbf{R}(\mathbf{w})$ is a $N \times N$ diagonal matrix with (n, n) elements

$$R_{nn} = y_n(1 - y_n) \tag{11}$$

The Hessian is no longer constant, depends on $\mathbf{w}$ through weighting matrix $\mathbf{R}$

Iterative reweighted least squares (cont.)

Because $0 < y_n < 1$, for an arbitrary vector $\mathbf{u}$, we have that $\mathbf{u}^T\mathbf{H}\mathbf{u} > 0$

- $\mathbf{H}$ is positive definite

The error function is concave in $\mathbf{w}$ and hence it has a unique minimum

The Newton-Raphson update formula becomes

$$\begin{aligned}\mathbf{w}^{(\text{new})} &= \mathbf{w}^{(\text{old})} - (\mathbf{\Phi}^T\mathbf{R}\mathbf{\Phi})^{-1}\mathbf{\Phi}^T(\mathrm{y}-\mathrm{t}) \\ &= (\mathbf{\Phi}^T\mathbf{R}\mathbf{\Phi})^{-1}\left(\mathbf{\Phi}^T\mathbf{R}\mathbf{\Phi}\mathbf{w}^{(\text{old})} - \mathbf{\Phi}^T(\mathrm{y}-\mathrm{t})\right) \\ &= (\mathbf{\Phi}^T\mathbf{R}\mathbf{\Phi})^{-1}\mathbf{\Phi}^T\mathbf{R}\mathrm{z} \qquad (12)\end{aligned}$$

where z is a N-vector with elements

$$\mathrm{z} = \mathbf{\Phi}\mathbf{w}^{(\text{old})} - \mathbf{R}^{-1}(\mathrm{y}-\mathrm{t}) \qquad (13)$$

Iterative reweighted least squares (cont.)

$$\mathbf{w}^{\text{new}} = (\mathbf{\Phi}^T \mathbf{R} \mathbf{\Phi})^{-1} \mathbf{\Phi}^T \mathbf{R} \mathbf{z} \quad \text{with } \mathbf{z} = \mathbf{\Phi}\mathbf{w}^{(\text{old})} - \mathbf{R}^{-1}(\mathbf{y} - \mathbf{t})$$

The update is the set of normal equations for a weighted least-squares problem

Because the weighing matrix $\mathbf{R}$ is not constant but depends on the parameter vector $\mathbf{w}$, we must apply the normal equations iteratively

- each time using the new weight vector $\mathbf{w}$ to compute revised weights $\mathbf{R}$

For this reason, the algorithm is **iterative reweighted least squares**, or IRLS

Iterative reweighted least squares (cont.)

As in weighted least-squares problems, the elements of the diagonal weighting matrix $\mathbf{R}$ can be interpreted as variances because the mean and variance of t ($t^2 = t$, for $t \in \{0, 1\}$) in the logistic regression model are

$$\mathbb{E}[t] = \sigma(\mathbf{x}) = y \tag{14}$$

$$\text{var}[t] = \mathbb{E}[t^2] - \mathbb{E}[t]^2 = \sigma(\mathbf{x}) - \sigma(\mathbf{x})^2 = y(1-y) \tag{15}$$

We interpret IRLS as solution to a linearised problem in the space of $a = \mathbf{w}^T\phi$

The quantity z_n (n-th element of $\mathbf{z}$) can then be given an interpretation as an effective target value in this space by making a local linear approximation to the logistic sigmoid function around the current operating point $\mathbf{w}^{(\text{old})}$

$$\begin{aligned} a_n(\mathbf{w}) &\simeq a_n\mathbf{w}^{(\text{old})} + \left.\frac{da_n}{dy_n}\right|_{\mathbf{w}^{(\text{old})}}(t_n - y_n) \\ &= \phi_n^T\mathbf{w}^{(\text{old})} - \frac{(y_n - t_n)}{y_n(1-y_n)} = z_n \end{aligned} \tag{16}$$

Multiclass logistic regression
Probabilistic discriminative models

Multiclass logistic regression

In the discussion of generative models for multiclass classification, we have seen that for a large class of distributions, the posterior probabilities are given by a softmax transformation of linear functions of feature variables

$$p(\mathcal{C}_k|\phi) = y_k(\phi) = \frac{\exp(a_k)}{\sum_j \exp(a_j)} \tag{17}$$

where the activations a_k are

$$a_k = \mathbf{w}_k^T \phi \tag{18}$$

We used maximum likelihood to determine separately the class-conditional densities and the class priors and then found the corresponding posterior probabilities using Bayes' theorem, implicitly determining parameters $\{\mathbf{w}_k\}$

Multiclass logistic regression (cont.)

We can use maximum likelihood to get parameters $\{\mathbf{w}_k\}$ of this model directly

To do this, we need the derivatives of y_k with respect to all of the activations a_j

$$\frac{\partial y_k}{\partial a_j} = y_k(I_{kj} - y_j) \tag{19}$$

where I_{kj} are the elements of the identity matrix

Next we need to write the likelihood function using the 1-of-K coding scheme

- The target vector $\mathbf{t}_n$ for feature vector ϕ_n belonging to class $\mathcal{C}_k$ is a binary vector with all elements zero except for element k

The likelihood is then given by

$$p(\mathrm{T}|\mathbf{w}_1, \ldots, \mathbf{w}_K) = \prod_{n=1}^{N}\prod_{k=1}^{K} p(\mathcal{C}_k|\phi_n)^{t_{nk}} = \prod_{n=1}^{N}\prod_{k=1}^{K} y_k(\phi_n)^{t_{nk}} \tag{20}$$

where t_{nk} is an element in the $N \times K$ matrix T of target variables

Multiclass logistic regression (cont.)

Taking the negative logarithm gives

$$E(\mathbf{w}_1, \ldots, \mathbf{w}_K) = -\ln p(\mathbf{T}|\mathbf{w}_1, \ldots, \mathbf{w}_K) = -\sum_{n=1}^{N}\sum_{k=1}^{K} t_{nk} \ln(y_{nk}) \qquad (21)$$

This is the **cross-entropy** error function for the multiclass classification problem

We now take the gradient of the error function wrt to one parameter vector $\mathbf{w}_j$

$$\nabla_{\mathbf{w}_j} E(\mathbf{w}_1, \ldots, \mathbf{w}_K) = \sum_{n=1}^{N} (y_{nj} - t_{nj})\phi_n \qquad (22)$$

We used the result for derivatives of the softmax function, $\frac{\partial y_k}{\partial a_j} = y_k(I_{kj} - y_j)$[1]

[1] We used also $\sum_k t_{nk} = 1$

Multiclass logistic regression (cont.)

The same form for the gradient as found for the sum-of-squares error function with the linear model and the cross-entropy error for logistic regression model

- The product of the error $(y_{nj} - t_{nj})$ times the basis function ϕ_n

The derivative of the log likelihood function for a linear regression model with respect to the parameter vector **w** for a data point n took the same form

- The error $(y_n - t_n)$ times the feature vector ϕ_n

Similarly, for the combination of logistic sigmoid activation function and cross-entropy error function, and for the softmax activation function with the multiclass cross-entropy error function, we again obtain this same simple form

Multiclass logistic regression (cont.)

To find a batch algorithm, we can use the Newton-Raphson update to obtain the corresponding IRLS algorithm for the multiclass problem

This requires evaluation of the Hessian matrix that comprises blocks of size $M \times M$ in which block (j, k) is given by

$$\nabla_{\mathbf{w}_k} \nabla_{\mathbf{w}_j} E(\mathbf{w}_1, \ldots, \mathbf{w}_k) = -\sum_{n=1}^{N} y_{nk}(I_{kj} - y_{nj})\phi_n \phi_n^T \tag{23}$$

As with two-classes, the Hessian matrix for the multiclass logistic regression models is positive definite and the error function has a unique minimum