Classification using Distance Nearest Neighbours

Nial Friel

University College Dublin
nial.friel@ucd.ie

March, 2009

Joint work with Tony Pettitt (QUT, Brisbane)

Main points of the talk

- MRFs give a flexible approach to modeling.
- Inference for discrete-valued MRF models using MCMC.
- Application to classification problems.

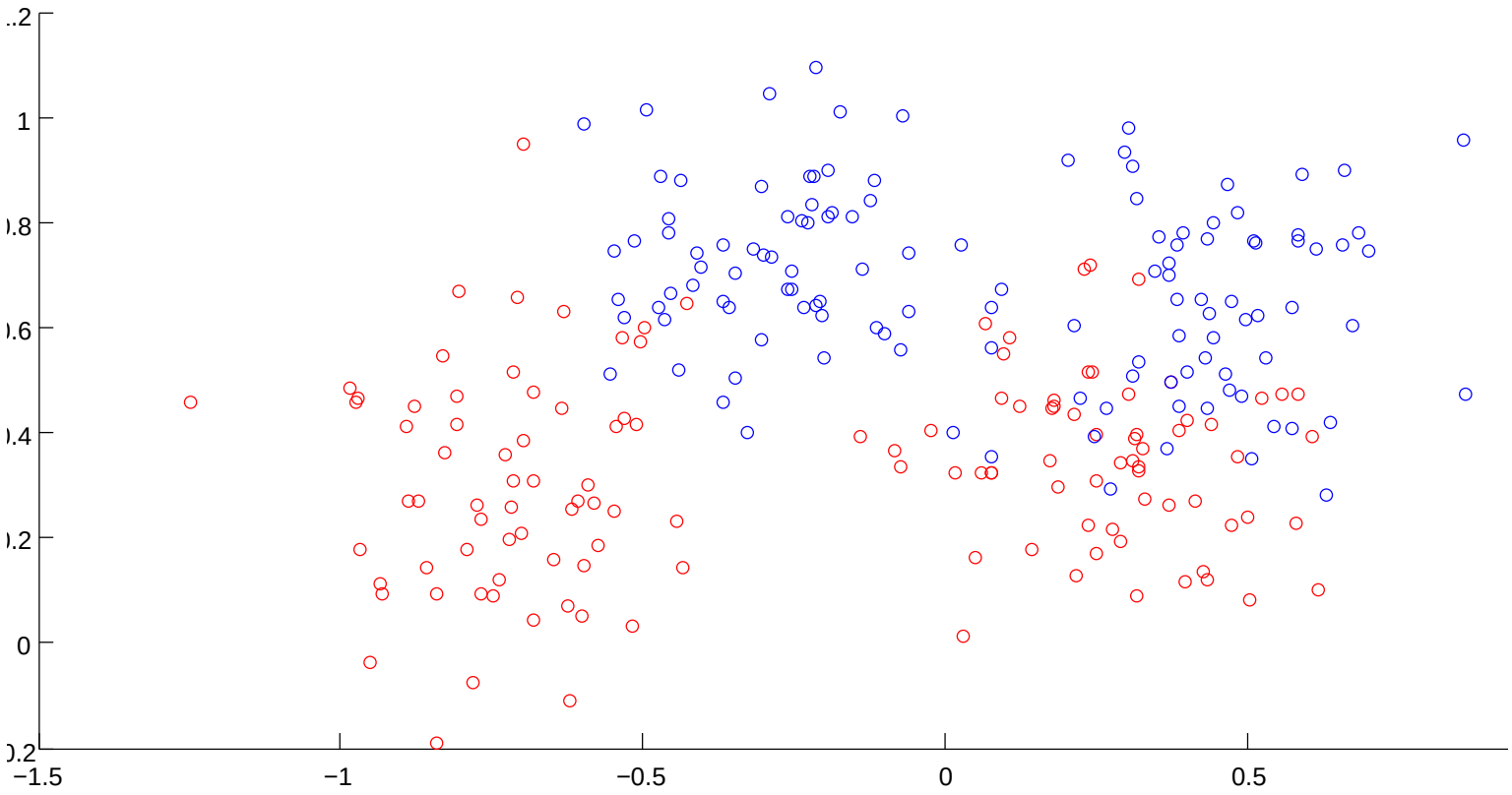
Classification and supervised learning

Given complete training data: $(x_i, y_i)_{i=1}^{n}$,
where the y_i's are class labels taking values $1, 2, \ldots, C$.

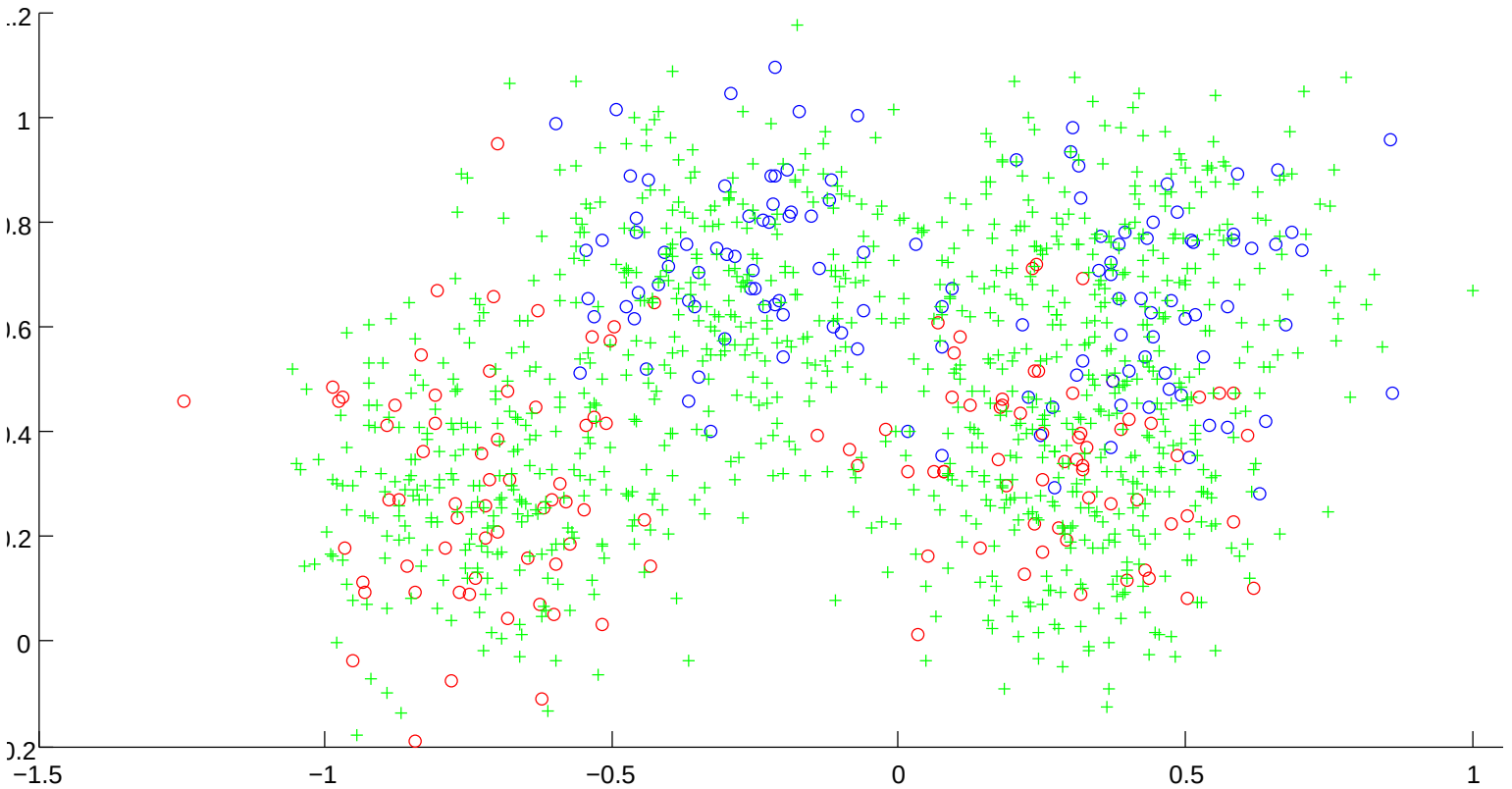
The problem:

Predict class labels y_i, for a collection of unlabelled/incomplete test features $(x_i)_{i=n+1}^{n+m}$.

An example



An example



k-nearest-neighbour algorithm

This algorithm dates back to Fix and Hodges (1951) and has been widely studied ever since.

knn algorithm

An unobserved class label y_{n+i} associated with a feature x_{n+i} is estimated by the most common class among the k nearest neighbours of x_{n+i} in the training set.

The *knn* algorithm

- It is deterministic, given the training data.
- It is not parametrised, but the choice of k is clearly crucial.

Probabilistic *knn*

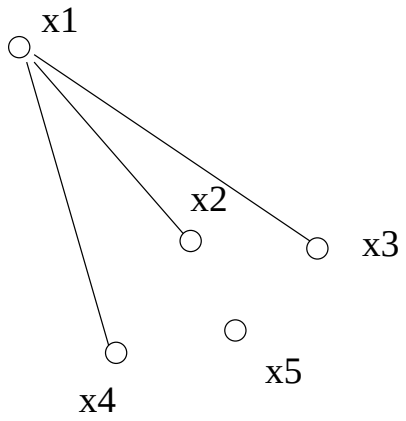
Holmes & Adams (2003) address the shortcomings of *knn*. They define a full-conditional distribution as

$$p(y_i|\mathbf{y}_{-i}, \mathbf{x}, \beta, k) = \frac{\exp\left(\beta \sum_{j\sim^k i} I(y_j = y_i)/k\right)}{\sum_{g=1}^{G} \exp\left(\beta \sum_{j\sim^k i} I(y_j = g)/k\right)}$$

Here $j \sim^k i$ means 'x_j is one of the k nearest neighbours of x_i'. The parameter $\beta > 0$ controls the degree of uncertainty: $\beta = 0$ implies independence among $\mathbf{y}$, while increasing values of β lead to stronger dependence among $\mathbf{y}$.

A problem! There does not necessarily exists a valid joint distribution for $\mathbf{y}$ which has the above full-conditional distribution.

Probabilistic *knn* – Cucala, Marin, Robert, Titterington (2009)



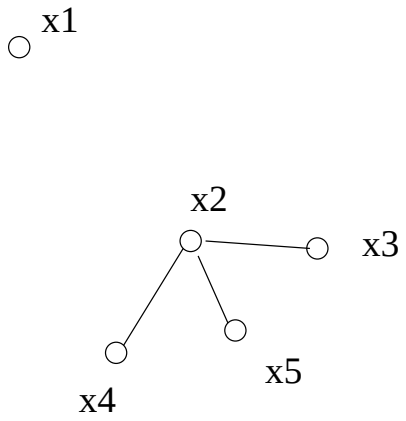
x_2 is one of the 3 nearest neighbours of x_1, but x_1 is not one of the 3 nearest neighbours of x_2.

This method attempts to build a joint distribution for $\mathbf{y}$ from a collection of full-conditional distributions $y_i|\mathbf{y}_{-i}$. This is how Markov random fields are often constructed.

But a necessary condition for a valid MRF is that the neighbour cliques are symmetric.

$$x_i \sim x_j \iff x_j \sim x_i$$

Probabilistic *knn* – Cucala, Marin, Robert, Titterington (2009)



x_2 is one of the 3 nearest neighbours of x_1, but x_1 is not one of the 3 nearest neighbours of x_2.

This method attempts to build a joint distribution for $\mathbf{y}$ from a collection of full-conditional distributions $y_i|\mathbf{y}_{-i}$. This is how Markov random fields are often constructed.

But a necessary condition for a valid MRF is that the neighbour cliques are symmetric.

$$x_i \sim x_j \iff x_j \sim x_i$$

Probabilistic *knn* – Cucala, Marin, Robert, Titterington (2009)

Cucala et al (2009) rectify this with

$$p(\mathbf{y}|\beta, \mathbf{x}) \exp\left\{ \beta \sum_{1}^{n} \sum_{j \sim^k i} I(y_i = y_j)/k \right\}.$$

Here $j \sim^k i$ means that the sum is taken over cases j which have k nearest neighbours of x_j.

The full-conditional distribution now appears as

$$p(y_i|\mathbf{y}_{-i}, \beta, \mathbf{x}) \propto \exp\left\{ \beta/k \left(\sum_{j \sim^k i} I(y_i = y_j) + \sum_{i \sim^k j} I(y_j = y_i) \right) \right\}$$

Probabilistic *knn* – Cucala, Marin, Robert, Titterington (2009)

Cucala et al (2009) rectify this with

$$p(\mathbf{y}|\beta, \mathbf{x}) \exp\left\{\beta \sum_{1}^{n} \sum_{j \sim^k i} I(y_i = y_j)/k\right\}.$$

Here $j \sim^k i$ means that the sum is taken over cases j which have k nearest neighbours of x_j.

The full-conditional distribution now appears as

$$p(y_i|\mathbf{y}_{-i}, \beta, \mathbf{x}) \propto \exp\left\{\beta/k \left(\sum_{j \sim^k i} I(y_i = y_j) + \sum_{i \sim^k j} I(y_j = y_i)\right)\right\}$$

Therefore, some points get summed twice, if they are mutual neighbours, and summed once if they aren't mutual neighbours.

Remarks on *pknn*

- Both of these algorithms might be criticised from the aspect that points are always included as neighbours regardless of their distance.
- The neighbourhood model of *pknn* is an Ising type model where all neighbouring points have equal influence regardless of distance.
- This suggests that both algorithms might not handle outliers in a sensible way.

Distance nearest neighbours

$$p(y_i|\mathbf{y}_{-i}, \mathbf{x}, \beta, \rho) \propto \exp\left(\beta \sum_{j \neq i} w_j^i I(y_j = y_i)\right).$$

The weights sum to 1 and decrease as $d(x_i, x_j)$ increases, eg,

$$w_j^i \propto \exp\left(\frac{-d(x_i, x_j)^2}{2\rho^2}\right); \;\; j \in \{1, \ldots, n\} \setminus \{i\}$$

so that features which are closer to x_i have more influence than those which are further away.

Here the neighbourhood of each point is of maximal size.

Computational difficulties

The joint distribution appears as

$$p(\mathbf{y}|\mathbf{x}, \beta, \rho) = \frac{\exp\left(\beta \sum_i \sum_{j \sim_\rho i} w_j^i I(y_j = y_i)\right)}{z(\beta, \rho)}$$

The normalising constant is difficult to evaluate:

$$z(\beta, \rho) = \sum_{y_1} \cdots \sum_{y_n} \exp\left(\beta \sum_i \sum_{j \sim_\rho i} w_j^i I(y_j = y_i)\right)$$

An approximate solution – Pseudolikelihood:

$$p(\mathbf{y}|\mathbf{x}, \beta, \rho) = \prod_{i=1}^{n} p(y_i|\mathbf{y}_{-i}, \mathbf{x}, \beta, \rho)$$

This gives a fast solution.
But it ignores dependencies in the underlying graph. Only first order dependencies are accounted for.

Usual MCMC doesn't work in this case

Consider the general problem of sampling from $p(\theta|\mathbf{y}) \propto p(\mathbf{y}|\theta)p(\theta)$, where $p(\mathbf{y}|\theta) = q(\mathbf{y}|\theta)/z(\theta)$ and $z(\theta)$ is impossible to evaluate.

Metropolis-Hasting algorithm:

$$\alpha(\theta, \theta') = \min\left(1, \frac{q(\mathbf{y}|\theta')p(\theta')}{q(\mathbf{y}|\theta)p(\theta)} \times \frac{z(\theta)}{z(\theta')}\right)$$

The intracability of $z(\theta)$ makes this algorithm infeasible.

Overcoming the NC problem – Auxiliary variable method

Møller *et al.* (2006)

Introduce an auxiliary variable $\mathbf{x}$ on the same space as the data $\mathbf{y}$ and extend the target distribution

$$p(\theta, \mathbf{x}|\mathbf{y}) \propto p(\mathbf{y}|\theta)p(\theta)p(\mathbf{x}|\theta_0),$$

for some fixed θ_0.
Joint update $(\theta^*, \mathbf{x}^*)$ with proposal:

$$h(\theta^*, \mathbf{x}^*|\theta, \mathbf{x}) = h_1(\mathbf{x}^*|\theta^*)h_2(\theta^*|\theta, \mathbf{x}^*)$$

where

$$h_1(\mathbf{x}^*|\theta^*) = p(\mathbf{x}^*|\theta^*) = \frac{q(\mathbf{x}^*|\theta^*)}{z(\theta^*)}.$$

$$\alpha(\theta^*, \mathbf{x}^*|\theta, \mathbf{x}) = \frac{p(\mathbf{y}|\theta^*)p(\theta^*)p(\mathbf{x}^*|\theta_0)p(\mathbf{x}|\theta)h_2(\theta|\theta^*)}{p(\mathbf{y}|\theta)p(\theta)p(\mathbf{x}|\theta_0)p(\mathbf{x}^*|\theta^*)h_2(\theta^*|\theta)}$$

$z(\theta^*)$ appears in $p(\mathbf{y}|\theta^*)$ above and in $p(\mathbf{y}'|\theta')$ below, and therefore cancels. Similarly $z(\theta)$ cancels above and below.

The choice of θ_0 is important. eg the maximum pseudolikelihood estimate based on $\mathbf{y}$.

Overcoming the NC problem – the exchange algorithm

Murray, Ghahramani and MacKay (2007)

Augment the intractable distribution $p(\theta|\mathbf{y})$ with variables θ' and $\mathbf{y}'$, where $\mathbf{y}'$ belongs to the same space as $\mathbf{y}$.

$$p(\theta, \theta', \mathbf{y}'|\mathbf{y}) \propto p(\mathbf{y}|\theta)p(\theta)h(\theta'|\theta)p(\mathbf{y}'|\theta')$$

1. Gibbs update of $(\theta', \mathbf{y}')$:
   draw θ' from $h(\theta'|\theta)$ and then $\mathbf{y}' \sim p(\mathbf{y}'|\theta')$.
2. Exchange $(\mathbf{y}, \theta), (\mathbf{y}', \theta')$ with $(\mathbf{y}, \theta'), (\mathbf{y}', \theta)$ using an MH transition.

$$\alpha = \min\left(1, \frac{p(\mathbf{y}|\theta')p(\theta')h(\theta|\theta')p(\mathbf{y}'|\theta)}{p(\mathbf{y}|\theta)p(\theta)h(\theta'|\theta)p(\mathbf{y}'|\theta')}\right)$$

Notice that $z(\theta')$ appears in $p(\mathbf{y}|\theta')$ above and $p(\mathbf{y}'|\theta')$ below, and therefore cancels! Similarly $z(\theta)$ cancels above and below.

Implementing the exchange algorithm

- The main difficulty with implementation of the exchange algorithm is the need to draw an exact sample $\mathbf{y}' \sim p(\mathbf{y}'|\theta')$.
- Perfect sampling is an obvious approach, if this is possible.
- A pragmatic alternative is to take a realisation from a long MCMC run with stationary distribution, $p(\mathbf{y}'|\theta)$ as an approximate draw.

The exchange algorithm and importance sampling.

The MH ratio in the exchange algorithm (assuming that $h(\theta, |\theta')$ is symmetric) can be written as

$$\frac{q(\mathbf{y}|\theta')p(\theta')}{q(\mathbf{y}|\theta)p(\theta)} \frac{q(\mathbf{y}'|\theta)}{q(\mathbf{y}'|\theta')}.$$

Compare this to the standard MH ratio:

$$\frac{q(\mathbf{y}|\theta')p(\theta')}{q(\mathbf{y}|\theta)p(\theta)} \frac{z(\theta)}{z(\theta')}$$

We see that the ratio of normalising constants, $z(\theta)/z(\theta')$, is replaced by $q(\mathbf{y}'|\theta)/q(\mathbf{y}'|\theta')$.
This ratio can be interpreted as an importance sampling estimate of $z(\theta)/z(\theta')$, since

$$\mathbf{E}_{\mathbf{y}|\theta'} \frac{q(\mathbf{y}'|\theta)}{q(\mathbf{y}'|\theta')} = \int \frac{q(\mathbf{y}'|\theta)}{q(\mathbf{y}'|\theta')} \frac{q(\mathbf{y}'|\theta')}{z(\theta)} \, d\mathbf{y} = \frac{z(\theta)}{z(\theta')}.$$

Predicting the unlabelled class data

We adopt a Bayesian model and consider the problem of classifying the unlabelled $\{y_{n+1}, \ldots, y_{n+m}\}$.

One alternative is to consider that all the class labels, $y_1, \ldots, y_n, y_{n+1}, \ldots, y_{n+m}$, arose from a single joint model

$$p(y_1, \ldots, y_{n+m} | \mathbf{x}, x_{n+1}, \ldots, x_{n+m}, \theta, \rho)$$

where some of the class labels are missing at random.

But this is computationally very challenging!

Predicting the unlabelled class data

Unclassified points can be labelled based on the marginal predictive distribution of y_{n+i}

$$p(y_{n+i}|x_{n+i}, \mathbf{x}, \mathbf{y}) = \int_{\rho} \int_{\beta} p(y_{n+i}|x_{n+i}, \mathbf{x}, \mathbf{y}, \beta, \rho) p(\beta, \rho|\mathbf{x}, \mathbf{y}) d\rho \; d\beta$$

where

$$p(\beta, \rho|\mathbf{x}, \mathbf{y}) \propto p(\mathbf{y}|\mathbf{x}, \beta, \rho) p(\beta) p(\rho)$$

is the posterior of (β, ρ) given the training data $(\mathbf{x}, \mathbf{y})$.

Recall: $p(\mathbf{y}|\mathbf{x}, \beta, \rho)$ is difficult to evaluate.

Results: Benchmark datasets

	C	F	N
Pima	2	8	532
Forensic glass	4	9	214
Iris	3	4	150
Crabs	4	5	130
Wine	3	13	178
Olive	3	9	572

Here C, F, N corresponds to the number of classes, the dimension of the feature vectors and the overall number of observations, respectively.

In each situation, the training dataset was approximately 25% of the size of the overall dataset.

In the Bayesian model, diffuse non-informative priors were chosen.

The dnn algorithm was run for $20,000$ iterations, with the first $10,000$ serving as burn-in iterations. The auxiliary chain within the exchange algorithm was run for $1,000$ iterations.

Results: Benchmark datasets

Misclassification rates

	knn	*dnn*
Pima	33%	29%
Forensic glass	40%	33%
Iris	5%	5%
Crabs	17%	17%
Wine	5%	3%
Olive	5%	3%

In all examples, the data was standardised to give transformed features with zero mean and unit variance.

The value of k in the *knn* algorithm was chosen as the value that minimises the leave-one-out cross-validation error rate in the training dataset.

In all cases *dnn* performs at least as well as *knn*.

Classification with a large feature set: Food authenticity

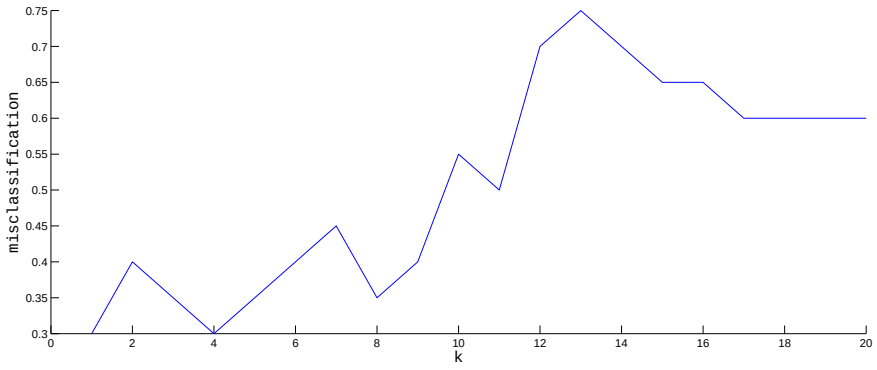
65 samples of Greek virgin olive-oil were analysed using near infra-red spectroscopy giving rise to 1050 reflectance values for wavelengths in the range $400 - 2098$nm. These values serve as the feature vector for each observation.

The aim of this study was to see if these measurements could be used to classify each olive-oil sample to its correct geographical region.

There are 3 possible classes: Crete (18 locations), Peloponnese (28 locations) and other regions (19 locations).

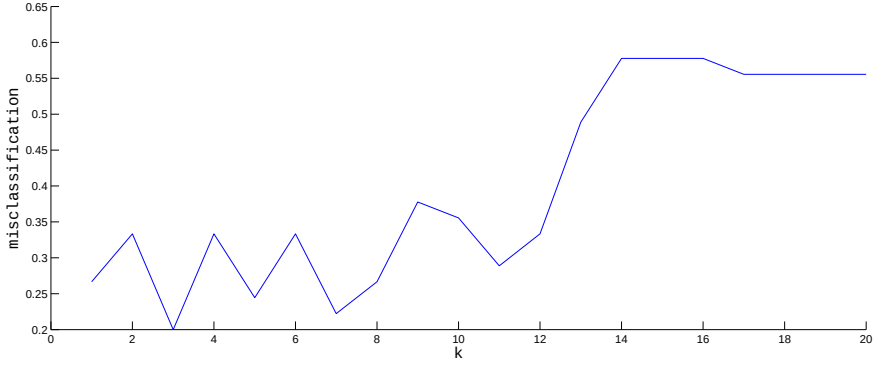
In our experiment the data were randomly split into a training set of 25 observations and a test set of 40 observations.

Results of *knn* algorithm



Training data: leave-one-out cross-validation.

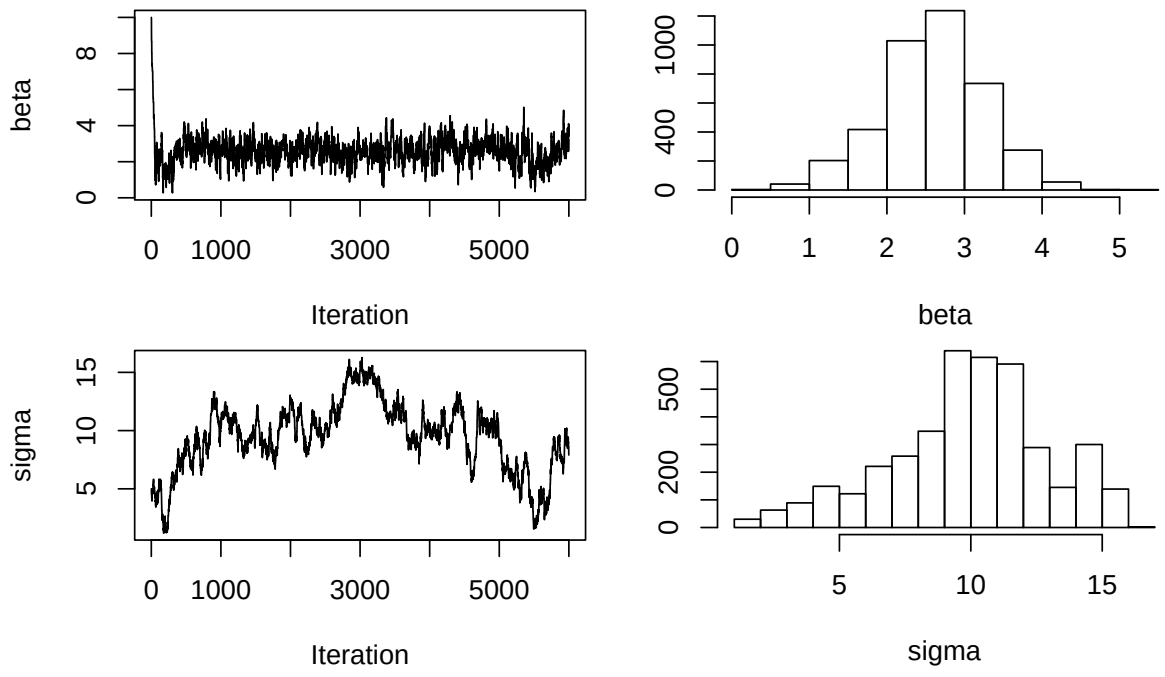
Leave-one-out cross-validation on the training data gives a minimum misclass rate for $k = 1, 4$.



Test data: misclass rate versus k.

The misclass rate for the test data at $k = 1$ and $k = 4$ is 27% and 33%, respectively. The minimum misclass rate is 20% ($k = 3$).

Results of *dnn* algorithm



50,000 Iterations; 20,000 burn-in; 1,000 iterations for the auxiliary chain. Acceptance rate is 24%.

The misclassification rate for the *dnn* algorithm was 17%. This compares favourably with the misclassification rate for the *knn* algorithm (27% or 33%).

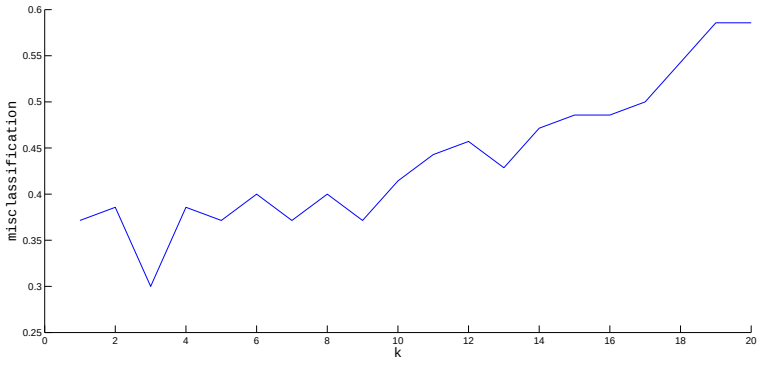
Food authenticity: classification of meat samples

Similar to the last example, the aim of this experiment was to classify 5 meat types according to 1050 reflectance values from near infra-red spectroscopy for 231 meat sample.

The data was randomly split into training and test data with the following frequencies within each class:

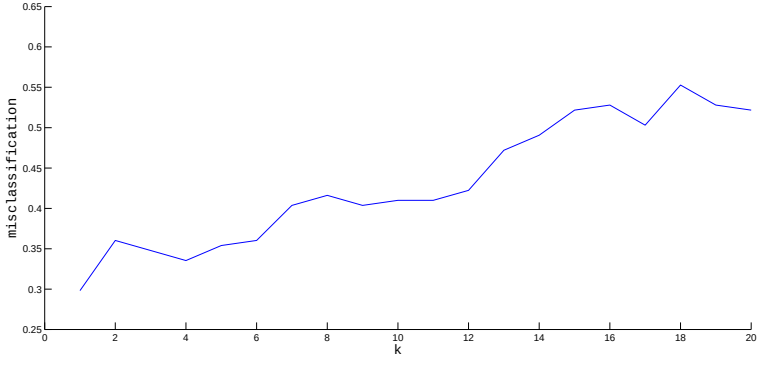
	Training data	Test data
Chicken	15	40
Turkey	20	35
Pork	13	42
Beef	11	21
Lamb	11	23

Results of *knn* algorithm



Training data: leave-one-out cross-validation.

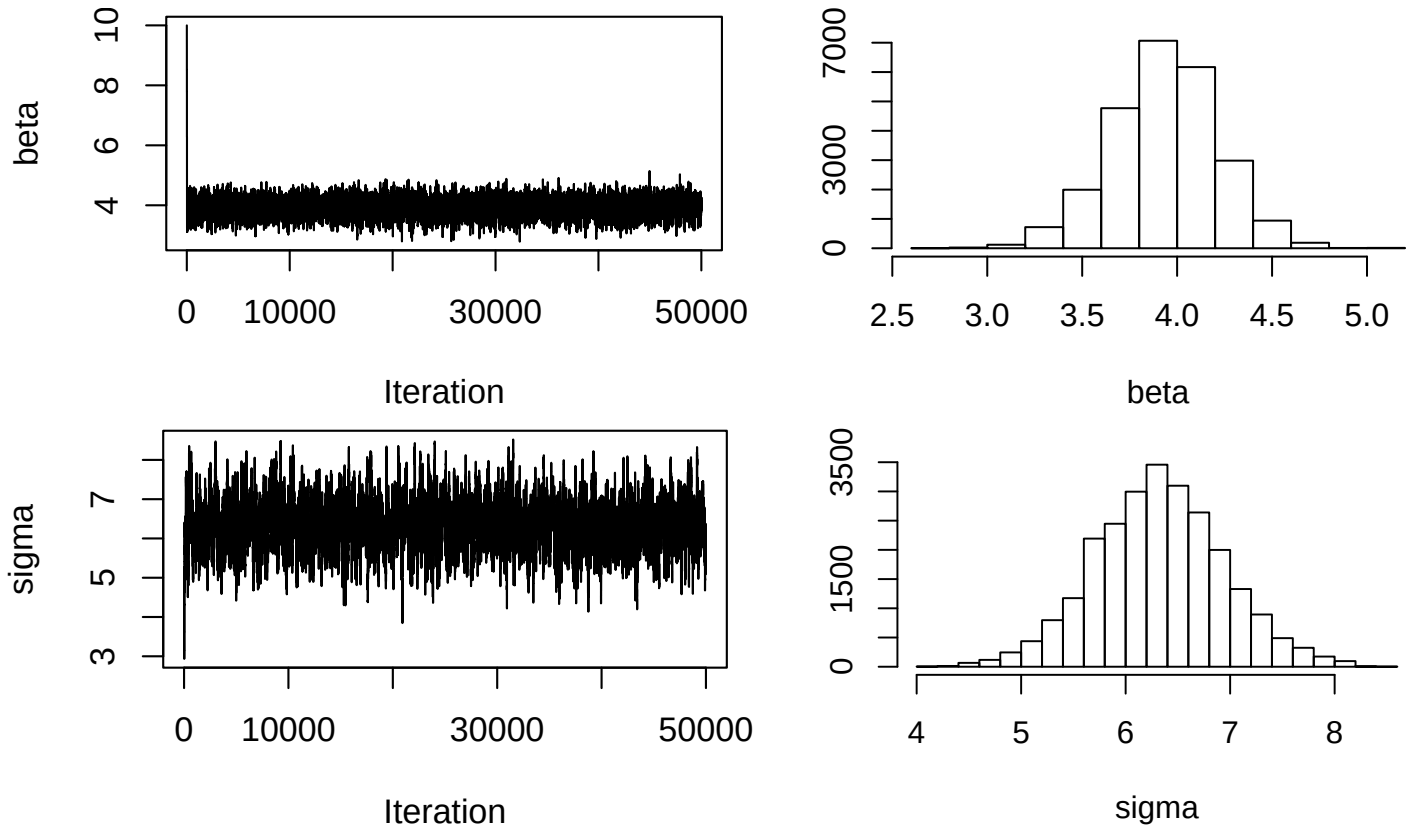
Leave-one-out cross-validation on the training data gives a minimum misclass rate for $k = 3$.



Test data: misclass rate versus k.

The misclass rate for the test data at $k = 3$ is 35%. The minimum misclass rate over all values of k is 30% ($k = 1$).

Results for *dnn*



50,000 Iterations; 20,000 burn-in; 1,000 iterations for the auxiliary chain. Acceptance rate is 22%.

The misclassification rate for the *dnn* algorithm was 30% (compared to 35% for *knn* algorithm).

Summary

- The *dnn* algorithm provides a probabilistic approach to a Bayesian analysis of supervised learning, building on the work of Cucala *et al* (2009) and shares many of the advantages of the approach there, providing a sound setting for Bayesian inference.
- The most likely allocations for the test dataset can be evaluated and also the uncertainty that goes with them. In addition, the Bayesian framework allows for an automatic approach to choosing weights for neighbours.
- Our work also also addresses the computational difficulties related to the well-known issue of the intractable normalising constant for discrete exponential family models. Our algorithm based on the exchange algorithm has very good mixing properties and therefore computational efficiency.