BMC course in Statistical Learning, 2009

Lectures: Niels Richard Hansen

- Homepage: `http://www.math.ku.dk/∼richard/courses/bmc2009/`
- Co-taught with the regular Statistical Learning course at University of Copenhagen.
- Evaluation:
  A minor, individual assignment – practical
  A major, individual project – mostly practical
- Theoretical training exercises handed out 26-4-2009.
- Practical exercises: During the course I have planned 9 small practical R-exercises that you will solve/work on in class. Solutions will be provided. Additional selected exercises from the book will be given.
- Teaching material: *The Elements of Statistical Learning. Data Mining, Inference, and Prediction 2nd ed.* together with hand-outs from the lectures.

Statistical Learning

What is Statistical Learning?

Old wine on new bottles? Is it not just plain statistical inference and regression theory?

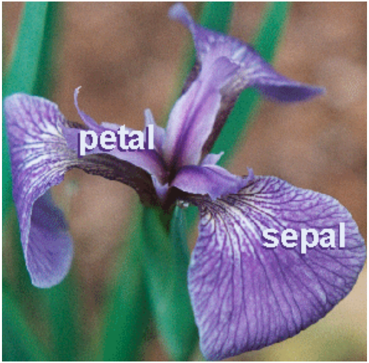
New(ish) field on how to use statistics to make the computer "learn"?

A merger of classical disciplines in statistics with methodology from areas known as machine learning, pattern recognition and artificial neural networks.

Major purpose: Prediction – as opposed to .... truth!?
Major point of view: Function approximation, solution of a mathematically formulated estimation problem – as opposed to algorithms.

Iris data

A classical dataset collected by the botanist Edgar Anderson, 1935, *The irises of the Gaspe Peninsula* and studied by statistician R. A. Fisher, 1936 *The use of multiple measurements in taxonomic problems*. Available as the iris dataset in the MASS library in R.



Sepal		Petal		
Length	Width	Length	Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
⋮	⋮	⋮	⋮	⋮
7.0	3.2	4.7	1.4	versicolor
6.4	3.2	4.5	1.5	versicolor
6.9	3.1	4.9	1.5	versicolor
⋮	⋮	⋮	⋮	⋮
6.3	3.3	6.0	2.5	virginica
5.8	2.7	5.1	1.9	virginica
7.1	3.0	5.9	2.1	virginica
⋮	⋮	⋮	⋮	⋮

Figure 1.1 – Prostate Cancer

A classical scenario from statistics. How does the response variable `lpsa` relate to a number of other measured or observed quantities – some continuous and some categorical?

Typical approach is regression – the scatter plot to the left might reveal some correlations.

Figure 1.2 – Hand Written Digits

A classical problem from pattern recognition. How do we classify an image of a handwritten number as 0 - 9?

This is the mail sorting problem based on zip codes.

It's not so easy – is
a nine or a five?

Figure 1.3 – Microarray Measurements

A problem of current importance. How does the many genes of our cells behave?

We can measure the activity of thousands of genes simultaneously – the gene expression levels – and want to know about the relation of gene expression patterns to "status of the cell" (healthy, sick, cancer, what type of cancer ...)

Classification

The objective in a classification problem is to be able to classify an object into a finite number of distinct groups based on observed quantities.

With hand written digits we have 10 groups and an 8x8 pixel gray tone image (a vector in $\mathbb{R}^{256}$).

With microarrays a typical scenario is that we have 2 groups (cancer type A and cancer type B) and a 10-30 thousand dimensional vector of gene expressions.

Setup – and One Simple Idea

We have observations $(x_1, y_1), \ldots, (x_N, y_N)$ with $x_i \in \mathbb{R}^p$ and $y_i \in \{0, 1\}$. We assume that the data arose as independent and identically distributed samples of a pair (X, Y) of random variables.

Assume $X = x_0 \in \mathbb{R}^p$ what is Y? Let

$$N_k(x_0) = \{i \mid x_i \text{ is one of the } k\text{'th nearest observations}\}.$$

Define

$$\hat{f}(x_0) = \frac{1}{k} \sum_{i \in N_k(x_0)} y_i \in [0, 1]$$

and classify using majority rules

$$\hat{y} = \begin{cases} 1 & \text{if } \hat{f}(x_0) \geq 1/2 \\ 0 & \text{if } \hat{f}(x_0) < 1/2 \end{cases}$$

Figure 2.2 – 15-Nearest Neighbor Classifier

A wiggly separation barrier between x_0's classified as zero's and one's is characteristic of nearest neighbors. With $k = 15$ we get a partition of the space into just two connected "classification components".

Figure 2.3 – 1-Nearest Neighbor Classifier

With $k = 1$ every observed point has its own “neighborhood of classification”. The result is a large(r) number of connected classification components.

Linear Classifiers

A classifier is called linear if there is an affine function

$$x \mapsto x^T \beta + \beta_0$$

with the classifier at x_0

$$f(x) = \begin{cases} 1 & \text{if } x^T\beta + \beta_0 \geq 0 \\ 0 & \text{if } x^T\beta + \beta_0 < 0 \end{cases}$$

There are several examples of important linear classifiers. We encounter

- Linear discriminant analysis (LDA).
- Logistic regression.
- Support vector machines.

Tree based methods is a fourth method that relies on locally linear classifiers.

Regression

If the y variable is continuous we usually talk about regression. You should all know the linear regression model

$$Y = X^T \beta + \beta_0 + \epsilon$$

where ϵ and X are independent, $E(\epsilon) = 0$ and $V(\epsilon) = \sigma^2$.

We talk about a prediction $f(x)$ of Y given $X = x$ where $f : \mathbb{R}^p \rightarrow \mathbb{R}$ is a predictor. In the linear regression model above

$$f(x) = E(Y|X = x) = x^T \beta + \beta_0$$

is a natural choice of linear predictor.

Statistical Decision Theory

Question: How do we make optimal decisions of action/prediction under uncertainty?

We need to

- decide how we measure the quality of the decision – loss functions,
- decide how we model the uncertainty – probability measures,
- decide how we weigh together the losses.

Loss Functions

A loss function in the framework of ordinary regression analysis is a function $L : \mathbb{R} \times \mathbb{R} \to [0, \infty)$.

A predictor is a function $f : \mathbb{R}^p \to \mathbb{R}$. If $(x, y) \in \mathbb{R}^p \times \mathbb{R}$ the quality of predicting y as $f(y)$ is measured by the loss

$$L(y, f(x)).$$

Large values are bad! Examples where $L(y, \hat{y}) = V(y - \hat{y})$:

- The squared error loss; $V(t) = t^2$.
- The absolute value loss; $V(t) = |t|$.
- Huber for $c > 0$; $V(t) = t^2 1(|t| \leq c) + (2c|t| - c^2) 1(|t| > c)$.
- The ϵ-insensitive loss; $V(t) = |t| 1(|t| > \epsilon)$.

Probability Models

Let (X, Y) be a random variable with values in $\mathbb{R}^p \times \mathbb{R}$ and decompose the distribution of P into the conditional distribution P_x of Y given $X = x$ and the marginal distribution P_1 of X. This means

$$\Pr(X \in A, Y \in B) = \int_A P_x(B) P_1(\mathrm{d}x).$$

Recall that if the joint distribution has density $f(x, y)$ w.r.t. the Lebesgue measure the marginal distribution has density

$$f_1(x) = \int f(x, y) \mathrm{d}y$$

and the conditional distribution has density

$$f(y|x) = \frac{f(x, y)}{f_1(x)},$$

and we have Bayes formula $f(x, y) = f(y|x) f_1(x)$.

Weighing the Loss

If L is a loss function, (X, Y) a random variable and $f : \mathbb{R}^p \to \mathbb{R}$ a predictor then $L(Y, f(X))$ has a probability distribution on $[0, \infty)$.

Single number summaries of the distribution include

- Expected prediction error; $\mathrm{EPE}(f) = E(L(Y, f(X)))$.
- Median prediction error; $\mathrm{MPE}(f) = \mathrm{median}(L(Y, f(X)))$.
- Complicated 1; $C_1(f) = E(L(Y, f(X)))^2 + \lambda V(L(Y, f(X)))$.
- Complicated 2; $C_2(f) = E(L(Y, f(X))) + \lambda \mathrm{Pr}(L(Y, f(X)) > r)$.

Take Home Message

The quality of a predictor and the theory of statistical decision theory depend upon several highly subjective choices.

In practice the choices are mathematically convenient surrogates. We investigate the resulting methodology and try to understand pros and cons of the choices.

Using expected prediction error combined with the squared error loss is the best understood setup.

The model choice is not entirely subjective – we return to that below.

Optimality is never an unconditional quality – a predictor can only be optimal given the choice of loss function, probability model and weighing method.

Optimal Prediction

We find that

$$\begin{aligned}\mathrm{EPE}(f) &= \int L(y, f(x)) P(\mathrm{d}x, \mathrm{d}y) \\ &= \int \underbrace{\int L(y, f(x)) P_x(\mathrm{d}y)}_{E(L(Y,f(x))|X=x)} P_1(\mathrm{d}x).\end{aligned}$$

This quantity is minimized by minimizing the expected loss conditionally on $X = x$,

$$f(x) = \operatorname*{argmin}_{\hat{y}} E(L(Y, \hat{y})|X = x).$$

- Squared error loss; $L(y, \hat{y}) = (y - \hat{y})^2$
$$f(x) = E(Y|X = x)$$
- Absolute value loss; $L(y, \hat{y}) = |y - \hat{y}|$
$$f(x) = \mathrm{median}(Y|X = x)$$

Optimal Classification

For classification problems the discrete variable Y does not take values in $\mathbb{R}$ but we can encode the values as $\{1, \ldots, K\}$. We require that the classifier $f : \mathbb{R}^p \to \{1, \ldots, K\}$ only take these finite number of values.

We only need to specify the losses $L(k, l)$ for $k, l = 1, \ldots, K$ and we get the conditional expected prediction error

$$E(L(Y, f(x))|X = x) = \sum_{k=1}^{K} L(k, f(x))P_x(k).$$

The optimal classifier is in general given by

$$f(x) = \text{argmax}_l \sum_{k=1}^{K} L(k, l)P_x(k).$$

0-1 loss and the Bayes classifier

The 0-1 loss function is $L(k,l) = 1(k \neq l)$ is very popular with

$$E(L(Y, f(x))|X = x) = 1 - P_x(f(x)).$$

The Bayes classifier is the optimal solution given by

$$f_B(x) = \mathrm{argmax}_k P_x(k)$$

The Bayes rate

$$EPE(f_B) = 1 - E(\max_k P_X(k))$$

is the expected prediction error for the Bayes classifier.

Figure 2.5 – The Bayes Classifier

The example data used for nearest neighbor are simulated and the Bayes classifier can be calculated exactly.

It can be computed using Bayes formula for $k = 0, 1$

$$\mathrm{Pr}(Y = k|X = x) = \frac{\pi_k f_k(x)}{\pi_0 f_0(x) + \pi_1 f_1(x)}$$

and the argmax is found to be

$$f(x) = \mathrm{argmax}_{k=0,1} \pi_k f_k(x).$$

In the example f_0 and f_1 are mixtures of 10 Gaussian distributions.

Estimation Methodology

The choice of optimal predictor is dictated by the probability model. Let $(P_\theta)_{\theta \in \Theta}$ denote a parametrized family of distributions for (X, Y) and f_θ the P_θ-optimal predictor.

How can we estimate f_θ from the sample $(x_1, y_1), \ldots, (x_N, y_N)$?

- The plug-in principle: Let $\hat{\theta}$ denote an estimator of θ and take $f_{\hat{\theta}}$.
- The conditional plug-in principle: Assume that the conditional distribution, $P_{x,\tau(\theta)}$, of Y given $X = x$ depends upon θ through a parameter function $\tau : \Theta \to \Theta_2$. Then $f_\theta = f_{\tau(\theta)}$ and if $\hat{\tau}$ is an estimator of τ we take $f_{\hat{\tau}}$.
- Direct method: Forget the probabilistic model.
    - Aim for a direct, non-parametric estimator of $f_\theta(x)$, e.g. the idea behind nearest neighbors for estimation of $E(Y|X = x)$.
    - Empirical risk minimization: Take $\mathcal{F}$ to be a set of predictor functions and take

$$\hat{f} = \operatorname*{argmin}_{f \in \mathcal{F}} \frac{1}{n} \sum L(y_i, f(x_i)).$$

Figure 2.6 – Curse of dimension

The side lengths (Distance) of a subcube in dimension d as a function of its volume r is $r^{1/d}$, which increases rapidly with d. Almost everything is far away/close to the boundary in high dimensions. The median distance from the origin to the closest data point for N uniform points in the d-dimensional unit ball is

$$\left(1 - \frac{1}{2}^{1/N}\right)^{1/d}.$$

The Bias-Variance Tradeoff for nearest neighbors

Consider the case with $Y = f(X) + \epsilon$ where X and ϵ are independent, $E(\epsilon) = 0$ and $V(\epsilon) = \sigma^2$.

With $\hat{f}_k$ the k-nearest neighbor regressor the test error in x_0 is

$$\begin{aligned} E((Y - \hat{f}_k(x_0))^2 | X = x_0) \quad = \quad & E((Y - f(x_0))^2 | X = x_0) \\ & + E((f(x_0) - E(\hat{f}_k(x_0) | X_0 = x_0))^2 | X = x_0) \\ & + E((\hat{f}_k(x_0) - E(\hat{f}_k(x_0) | X_0 = x_0))^2 | X = x_0) \end{aligned}$$

$$= \sigma^2 + \underbrace{E\left[f(x_0) - \frac{1}{k} \sum_{l \in N_k(x_0)} f(X_l)\right]^2}_{\text{Squared bias}} + \underbrace{\frac{\sigma^2}{k}}_{\textit{variance}}$$

Small choices of k (complex model) will give a large variance and generally a smaller bias, and vice versa for large choices of k (simple model).

Figure 2.11 – The Generic Bias-Variance Tradeoff

The training error is the number $\overline{\text{err}} = \frac{1}{n}\sum_{i=1}^{n} L(y_i, \hat{f}(x_i))$. It generally decays with model complexity. The test error generally decays up to a point depending upon the sample size – and then it increases again.

The increase of the test error for complex models is known as overfitting – it is a variance phenomena. Bad performance for simple models is a bias phenomena.

The training error is a bad estimator for the test error and the expected prediction error.

Figure 2.4 – Bias-Variance Tradeoff for *k*-nearest neighbors

What is called the test error here is in reality an estimate of the expected prediction error for the estimated predictor based on an independent dataset.