Chapter 1. Introduction

Contents

1.0 Introduction

Searching for patterns in data with suitable algorithm – discovery of regularities. Example, classification problem for handwriting.

training set: using to tune the parameters for the model on the learning phase;

test set: *generalization*, predict the new input data based on the training phase;

feature extraction: usually do some typically pro processed on the original input variables which transform them into new space, such as centering, scaling, taking log and so on. It hopes to solve the problem easier and faster.

supervised learning: input variables with known corresponding targets, such as *regression* for continue response and *classfication* for discrete categories.

unsupervised learning: input variables without any corresponding targets value, such as *clustering* for discovering groups of similar input, *density estimation* and *visualization.*

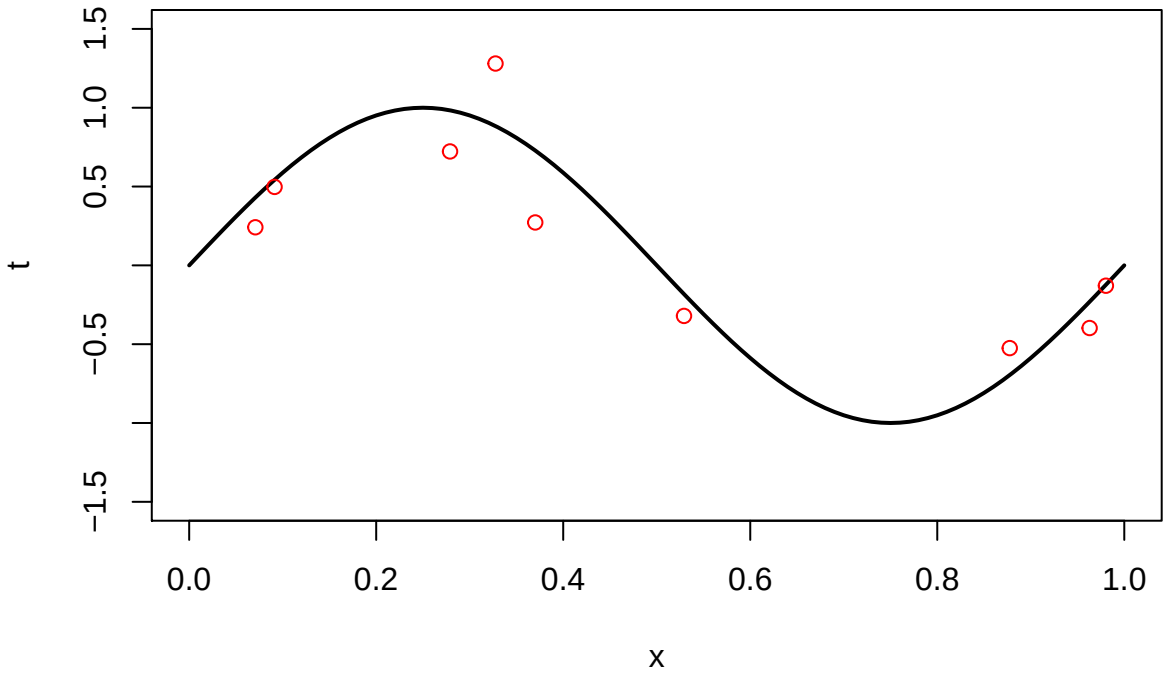
rainforcement learning(Sutton and Barton,1998): finding suitable actions to take in a given situation in order to maximize a reward; typically there is a sequence of states and actions such as neural network. The general feature of reinforcement learning is the trade-off between exploration, new kinds of action effect, and exploitation, known action to get higher reward.

1.1 Example: Polynomial Cure Fitting

One simple regression problem with known precise process.

(x_i, t_i), input x and target t value for subject $i \in 1 : N$ comes from function $sin(2\pi x)$ with normal noise.

```r
N <- 10
set.seed(2*N)
x_train <- runif(N, 0, 1)
t_train <- sin(2*pi*x_train) + rnorm(N, 0, 0.3)
curve(sin(2*pi*x), 0, 1, lwd = 2, ylim = c(-1.5, 1.5), ylab = "t")
points(x_train, t_train, col=2)
```



linear model linear for the coefficients.

$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + ... + w_M x^M = \sum_{j=0}^{M} w_j x^j \tag{1.1}$$

Error function: minimize to fitting the polynomial to the training data with solution $\mathbf{w}^\star$,

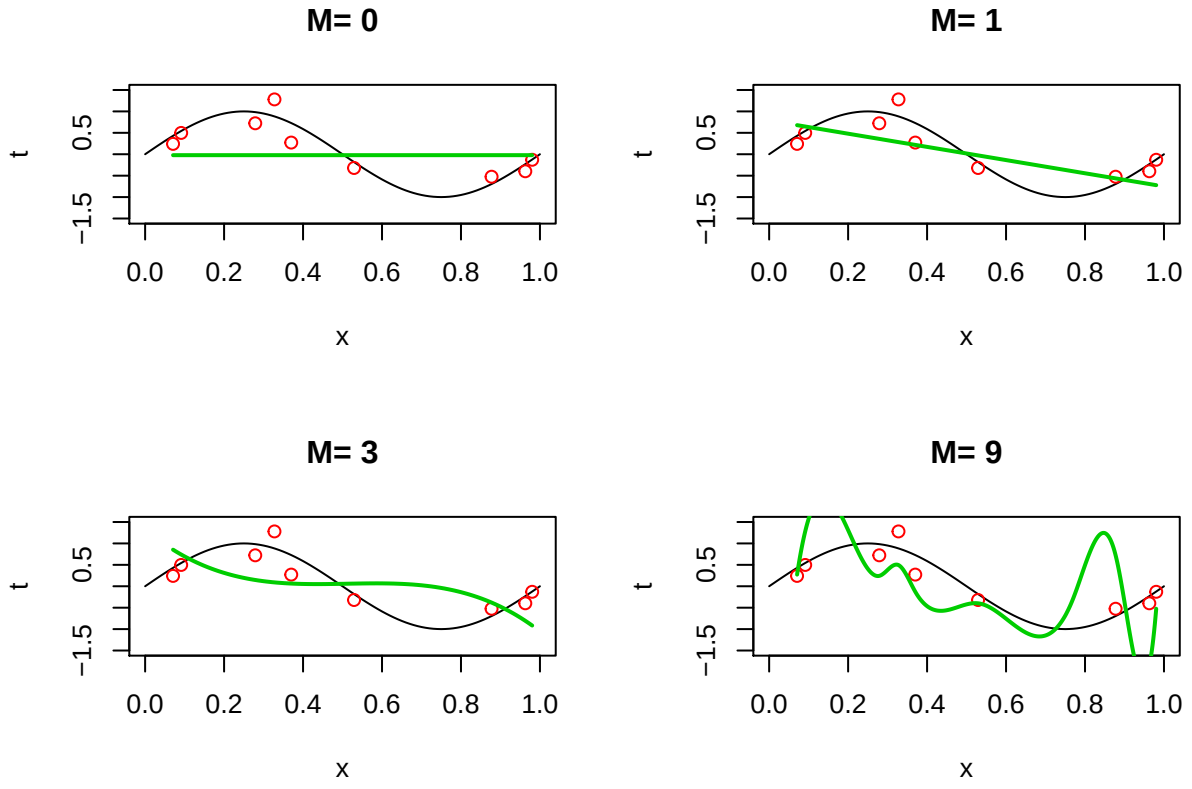
$$E(\mathbf{w}) = \frac{1}{2} \sum_{n=1}^{N} \{y(x_n, \mathbf{w}) - t_n\}^2 \tag{1.2}$$

Model comparison/model selection, how to chose M?

```r
linReg <- function(X, t) {
  ## X, with interspect term, N*p, here p=M+1
  ## y, N*1 vector

  XTX <- t(X)%*%X
  b <- solve(XTX, t(X)%*%t)
  se.b <- solve(XTX, diag(var(t), length(b)))
  return(cbind(b, se.b))
}
```

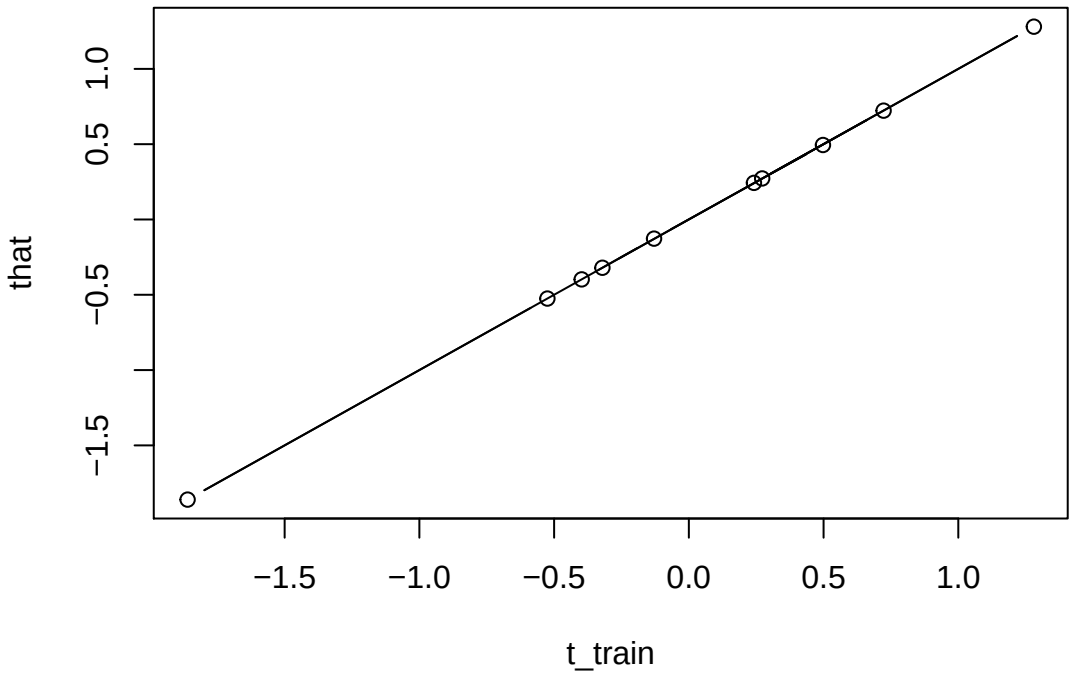
```r
M <- c(0, 1, 3, 9)
par(mfrow=c(2,2))
```

```
for(m in M){
  curve(sin(2*pi*x), 0, 1, lwd = 1, ylim = c(-1.5, 1.5), ylab = "t",
        main =paste("M=", m))
  points(x_train, t_train, col=2)
  m <- 0:m
  x1<- sort(x_train,  decreasing = T)
  X_train <- matrix(sapply(m, function(m) x1^m), N)
  B <- linReg(X_train, t_train)[, 1]
  that <- X_train%*%B
  sp <- spline(x1, that, n=500)
  lines(sp, col=3, lwd=2)
}
```



The fit value and the true value plot which is one line with $y = x$

```
plot(t_train, that,type = "b")
```



over-fitting: as the M become large, train error become smaller while test error become larger.

root-mean-square-error(RMS)

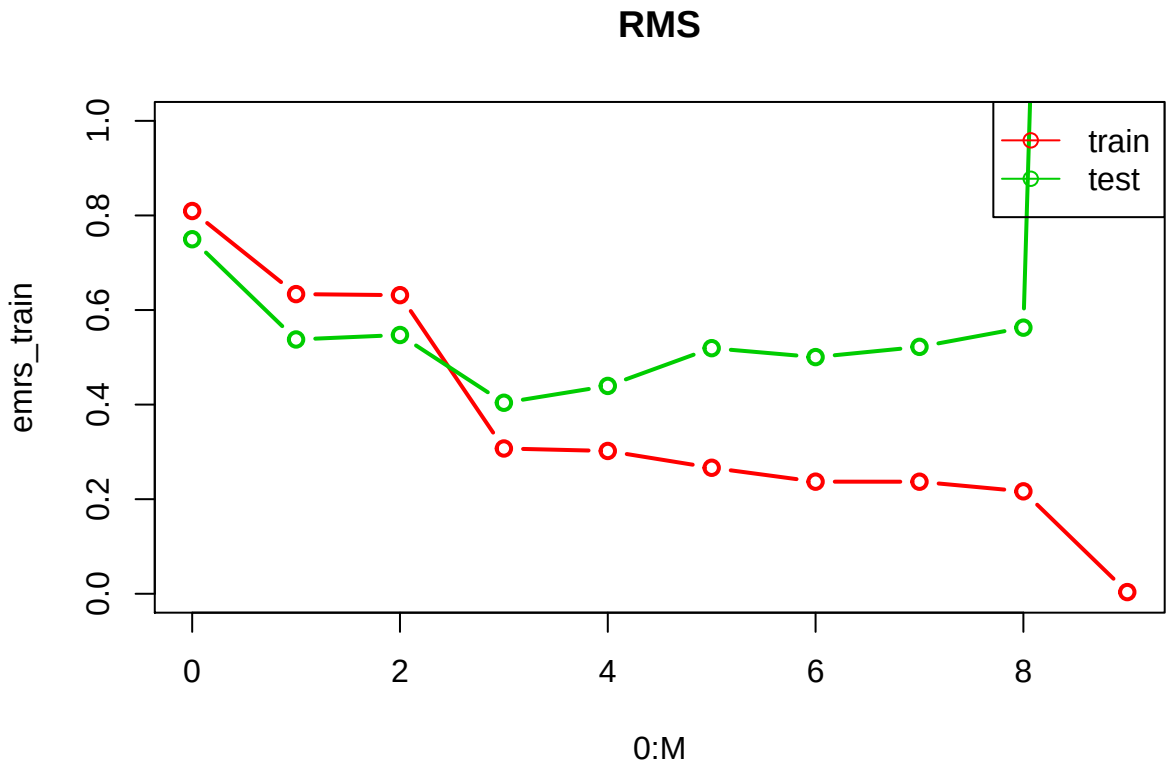
$$E_{RMS} = \sqrt{2E(\mathbf{w}^\star)/N}$$

```
Erms <- function(B, X, t) {
  that <- X%*%B
  res  <- sqrt(mean((t-that)^2))
  return(res)
}

x_test <- runif(100, 0, 1)
t_test <- sin(2*pi*x_test) + rnorm(100, 0, 0.3)

M <- 9
emrs_train <- emrs_test <- rep(0, M+1)
for(i in 0:M) {
  m <- 0:i
  X_train <- matrix(sapply(m, function(m) x_train^m), N)
  X_test  <- matrix(sapply(m, function(m) x_test^m), 100)
  B <- linReg(X_train, t_train)[, 1]
  emrs_train[i+1] <- Erms(B, X_train, t_train)
  emrs_test[i+1]  <- Erms(B, X_test, t_test)
}
# coefficient of M=9
round(B, 5)
```

```
##  [1] -7.595341e+01  2.855584e+03 -4.057954e+04  2.868830e+05 -1.145905e+06
##  [6]  2.750555e+06 -4.044145e+06  3.563544e+06 -1.726921e+06  3.537904e+05
```

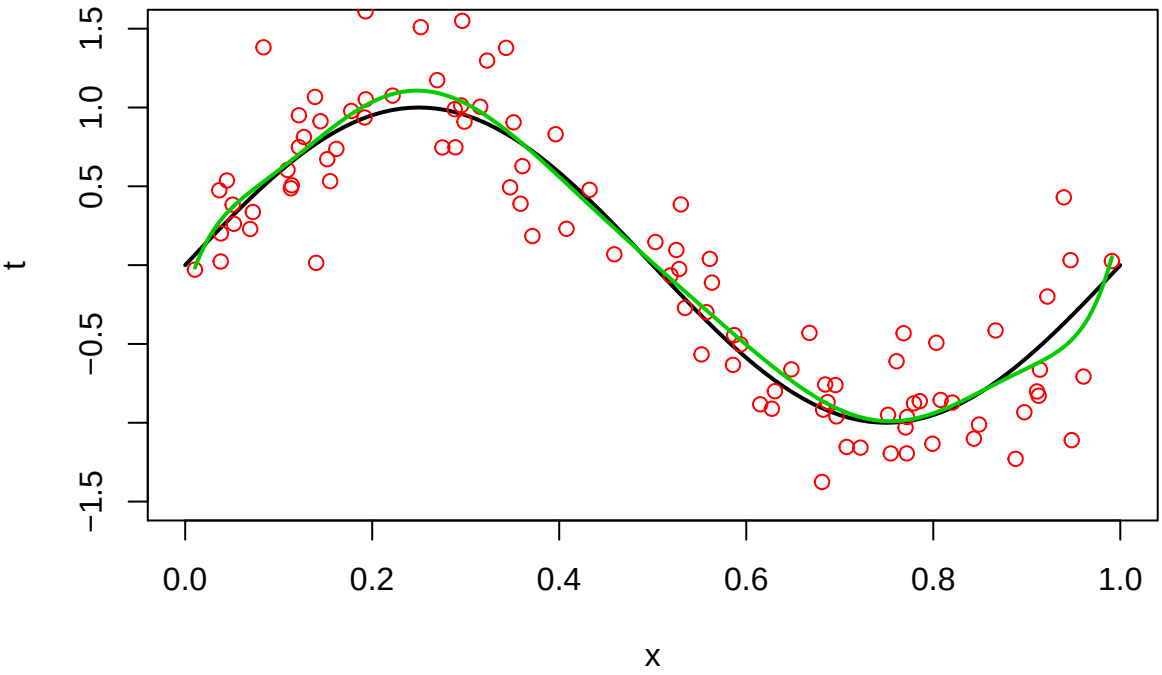
```
plot(0:M, emrs_train, type = "b", ylim = c(0,1), col=2, main = "RMS", lwd=2)
lines(0:M, emrs_test,type = "b", col = 3, lwd=2)
```

```
legend("topright", legend = c("train", "test"), col=2:3, lwd = 1, pch = 1)
```



As M increase, the coefficient of the polynomials of various order get larger. Intuitively, more flexible polynomial with larger M are becoming increasingly tuned to the random noise on the target values.

```
# figure 1.6
N <- 100
x <- runif(N, 0, 1)
t <- sin(2*pi*x) + rnorm(N, 0, 0.3)
curve(sin(2*pi*x), 0, 1, lwd = 2, ylim = c(-1.5, 1.5), ylab = "t")
points(x, t, col=2)

m <- 0:9
X <- matrix(sapply(m, function(m) x^m), N)
B <- linReg(X, t)[, 1]
that <- X%*%B
sp <- spline(x, that, n=5000)
lines(sp, col=3, lwd=2)
```



When N become large, the more complex or classifiable the model that the data can afford to fit. **So, usually advocates that the number of data points should be no less then 5 or 10 multiple ot the parameter numbers.**

It is necessary to consider the number of the parameters, many parameters model may not easy to interpret and needs more data. It should to choose the complexity of the model according to the complexity of the problem to being solved.

Least square approach is one special case of maximum likelihood methods, and that over-fitting problem can be understood as a general property of maximum likelihood. Bayesian model can be taken to avoid it by adapts the number of parameters according to the size of the data set.

With the current approach, one technique to control over-fitting in such case is to use regularization which adding one penalty terms to the RMS in order to discourage the coefficient from reaching larger value.

$$\hat{E}(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n, \mathbf{w}) - t_n\}^2 + \frac{\lambda}{2}||\mathbf{w}||^2 \tag{1.4}$$

coefficient λ governs the relative importance of the penalty terms compared with the sum-of-square error term. Intersect term w_0 is omitted from the plenary terms. Here is quadratic regularize, called ridge regression. It also use in neural network which known as *weight decay*.

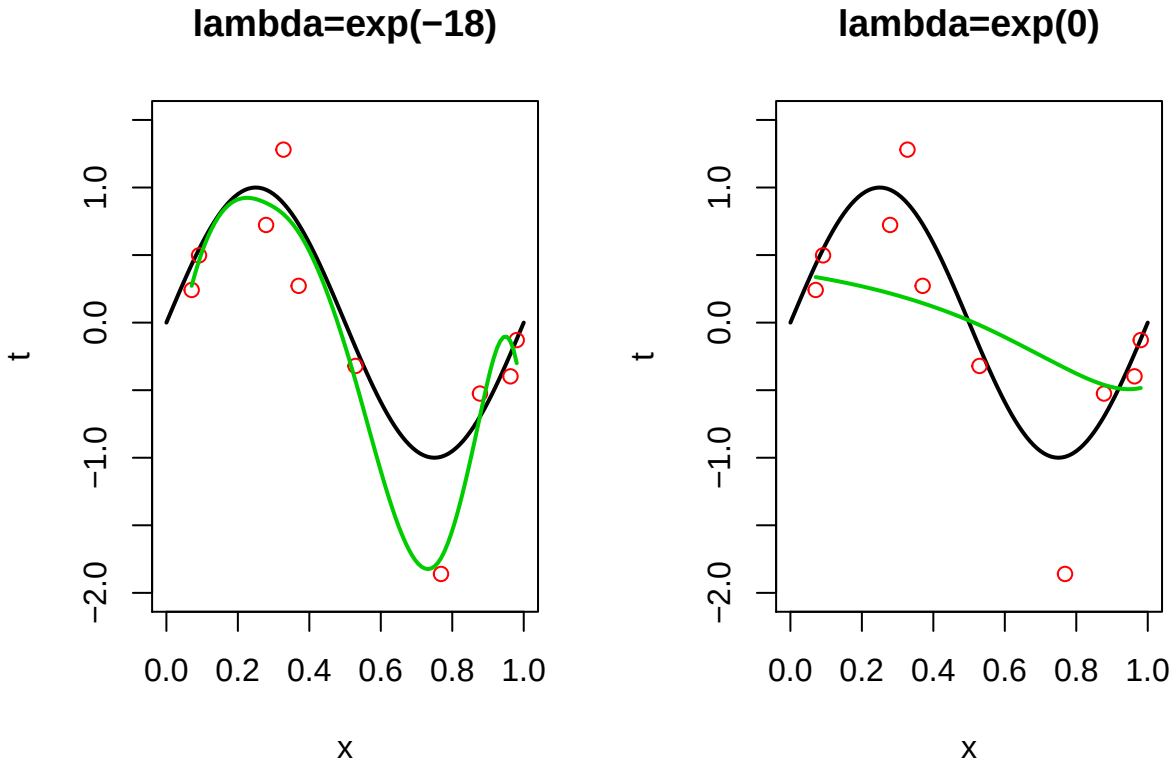
shrinkage methods for it reduce the value of the coefficients.

In ridge regression, the coefficient has the closed form solution

$$\hat{\mathbf{w}} = (X^T X + \lambda I)^{-1} X^T t$$

```
LinRidge <- function(X, t, lambda=0) {
  ## X
  XTX <- t(X)%*%X + lambda*diag(1, ncol(X))
  b <- solve(XTX, t(X)%*%t)
  b
}
```

```r
par(mfrow=c(1, 2))
curve(sin(2*pi*x), 0, 1, lwd = 2, ylim = c(-2, 1.5), ylab = "t",
      main="lambda=exp(-18)")
points(x_train, t_train, col=2)

B <- LinRidge(X_train, t_train, lambda = exp(-18))
that <- X_train%*%B
sp <- spline(x_train, that, n=100)
lines(sp, col=3, lwd=2)

curve(sin(2*pi*x), 0, 1, lwd = 2, ylim = c(-2, 1.5), ylab = "t",
       main="lambda=exp(0)")
points(x_train, t_train, col=2)
B <- LinRidge(X_train, t_train, lambda = exp(0))
that <- X_train%*%B
sp <- spline(x_train, that, n=100)
lines(sp, col=3, lwd=2)
```



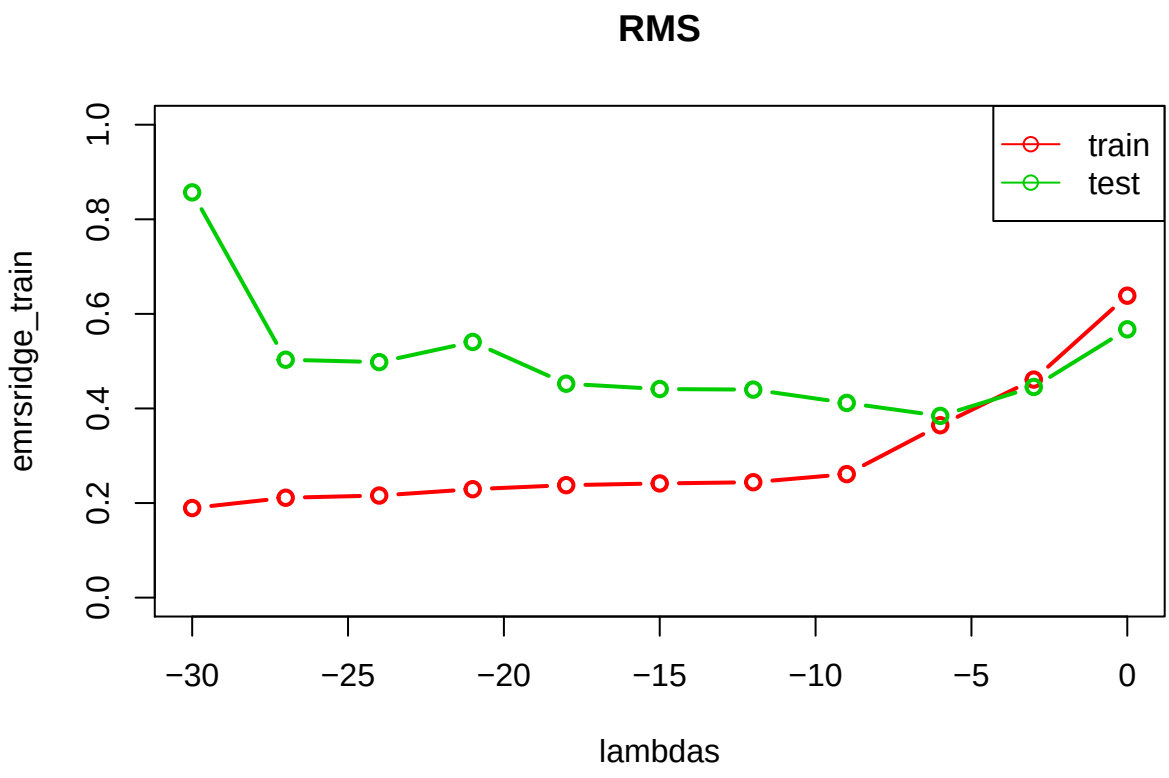
Too larger value of λ obtain a poor fit as above figure show.

```r
M <- 9
lambdas <- seq(-30, 0, 3)

emrsridge_train <- emrsridge_test <- rep(0, length(lambdas))
B <- matrix(0, nrow = 10, ncol = length(lambdas))
for(i in 1:length(lambdas)){
  lambda <- exp(lambdas[i])
  B[,i] <- LinRidge(X_train, t_train, lambda = lambda)
  emrsridge_train[i] <- Erms(B[,i], X_train, t_train)
  emrsridge_test[i]  <- Erms(B[,i], X_test, t_test)
```

```
}
round(B, 4)
```

```
##               [,1]        [,2]        [,3]       [,4]      [,5]      [,6]
##  [1,]      -7.0684      0.5743     -0.4796    -2.0755   -0.8616  -0.1970
##  [2,]     260.6210    -28.2357      0.5242    50.9013   23.0696   8.4569
##  [3,]   -3607.7473    520.5502    258.7078  -317.1236 -119.8638 -21.1257
##  [4,]   26080.8146  -3121.6899  -2214.6103   847.3456  312.9708  31.9682
##  [5,] -108625.5440   7814.6006   7621.1168  -620.8032 -356.7162 -51.9830
##  [6,]  274151.5212  -4489.0525 -11599.6895 -1233.7244  -96.1475 -26.3444
##  [7,] -425686.7387 -17645.5391   3216.5793  1642.1870  338.8560  46.0260
##  [8,]  396800.9382  38942.0838  11861.0869  1305.9956  151.2869  64.7190
##  [9,] -203327.5476 -30798.7719 -13626.3936 -2751.0604 -346.0277  10.2002
## [10,]   43961.3743   8805.9278   4483.3023  1078.0294   92.7398 -62.4404
##            [,7]     [,8]    [,9]   [,10]   [,11]
##  [1,]  -0.0570  -0.2910  0.2628  0.7070  0.3671
##  [2,]   5.2570   9.5929  3.7426 -0.2922 -0.3924
##  [3,]   0.4199 -16.0133 -7.0951 -1.7904 -0.4312
##  [4,] -21.7068 -11.6635 -4.5479 -1.4596 -0.3065
##  [5,] -17.5741   1.3373 -0.0064 -0.6886 -0.1677
##  [6,]   8.4899  11.2378  2.8023  0.0095 -0.0504
##  [7,]  30.1905  13.9789  3.5285  0.5260  0.0410
##  [8,]  30.2632   9.3183  2.6601  0.8721  0.1102
##  [9,]   4.9115  -1.3923  0.7482  1.0846  0.1618
## [10,] -40.7749 -16.3372 -1.7697  1.1991  0.1997
```

```
plot(lambdas, emrsridge_train, type = "b", ylim = c(0,1), col=2,
     main = "RMS", lwd=2)
lines(lambdas, emrsridge_test,type = "b", col = 3, lwd=2)
legend("topright", legend = c("train", "test"), col=2:3, lwd = 1, pch = 1)
```



From the coefficient value changes with the value λ, we can see the effect of the penalty both in train and test data. It can control the effective complexity of the model and hence determines the degree of over-fitting.

How to choose λ: validation set or hold out set used to optimize the model complexity both M and λ. *Wastefu of the data!.*

1.2 Probability Theory

Describe uncertainty which from the noise or limit data size.

sum rule : $p(X) = \sum_Y p(X, Y)$ (summing out or marginalizing)

product rule: $p(X, Y) = p(X)p(Y|X) = p(Y)p(X|Y)$.

Joint distribution/probability, *marginal probability*, *conditional probability*, *random varibale*, *prior probability*, *posterior probability*.

1.2.1 probability densities

probability density, *cumulative distirbution probability mass function*, which x is discrete.

1.2.2 Expectations and covariances

expectation, *conditional expectation*, *variance*, *covariance*,

1.2.3 Bayesian probabilities

classical or frequentist, is not suitable to describe rare events.

Thomas Bayes
1701–1761

Thomas Bayes was born in Tunbridge Wells and was a clergyman as well as an amateur scientist and a mathematician. He studied logic and theology at Edinburgh University and was elected Fellow of the Royal Society in 1742. During the 18th century, issues regarding probability arose in connection with gambling and with the new concept of insurance. One particularly important problem concerned so-called inverse probability. A solution was proposed by Thomas Bayes in his paper 'Essay towards solving a problem in the doctrine of chances', which was published in 1764, some three years after his death, in the *Philosophical Transactions of the Royal Society*. In fact, Bayes only formulated his theory for the case of a uniform prior, and it was Pierre-Simon Laplace who independently rediscovered the theory in general form and who demonstrated its broad applicability.

Figure 1:

Convert a prior probability($p(\mathbf{w})$) into a posterior probability by incorporating the evidence provided by the observed data $D = \{t_1, ..., t_N\}$ which is conditioned on the given parameters $p(D|\mathbf{w})$, *likelihood function*.

$$p(\mathbf{w}|D) = \frac{p(D|\mathbf{w})p(\mathbf{w})}{p(D)}$$

Both frequentest and Bayesian take the likelihood function as the central role. Frequentest take the w as a fixed parameters which used to estimated by *maximum likelihood* winch evaluated by the error bar, while Bayesian viewpoint there is only single data set with uncertainty parameters.

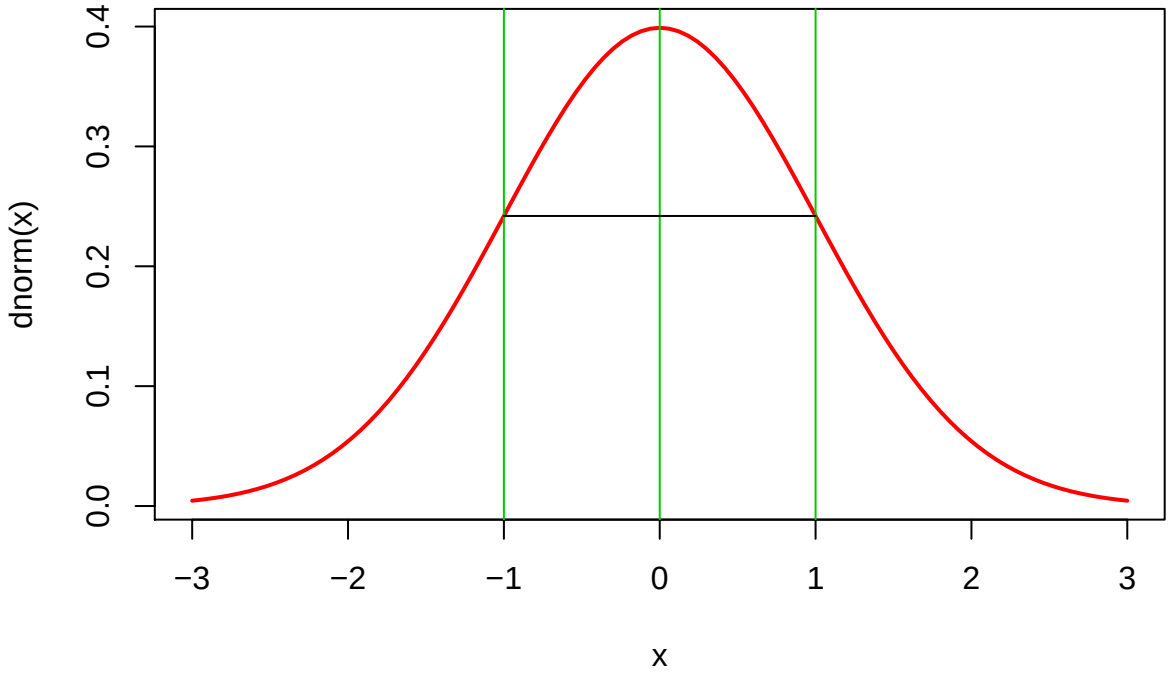
bootstrap, *noninformative prior*, * Markov chain Monte Carlo*.

1.2.4 The Gaussian distribution.

$$N(x|\mu,\sigma^2) = \frac{1}{\sqrt{(2\pi\sigma^2)}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

with $E(x) = \mu$, $var(x) = \sigma^2$.

```
curve(dnorm, -3, 3, col=2, lwd=2)
abline(v=c(-1,0,1), col=3)
lines(x=c(-1,0,1),y=rep(dnorm(1),3))
```



MLE:

$$\mu_{ML} = \frac{1}{N}\sum_{n=1}^{N} x_n, \quad E(\mu_{ML}) = \mu$$

$$\sigma^2_{ML} = \frac{1}{N}\sum_{n=1}^{N}(x_n - \mu_{ML})^2, \quad E(\sigma^2_{ML}) = \frac{N-1}{N}\sigma^2$$

Reason for variance bias it that it is measured relative to the sample mean μ_{ML} and not to the true mean μ. The bias becomes less significant as the sample size N increase. As the parameters increase, the bias will be much more severe, and this lies at the root of over-fitting problem.

1.2.5 Curve fitting re-visited

$$p(t|x,\mathbf{w},\beta) = N(t|y(x,\mathbf{w}),\beta^{-1}) \tag{1.60}$$

here β is the inverse variance which called precision. It can be show that minimizing the sum-of-squares error is equivalent to maximizing likelihood as the equation 1.61 to 1.62.

$$\frac{1}{\beta_{ML}} = \frac{1}{N}\sum_{n=1}^{N}(y(x,\mathbf{w}) - t_n)^2 \tag{1.63}$$

Taking the estimate of the parameters into 1.60 which build the *predictive distribution*. In traditional regression, we can take the point estimate as the predictive function which is fix value change with the x. However, lets take a Bayesian approach introduce a prior for the coefficient w and consider a normal distribution,

$$p(\mathbf{w}|\alpha) = N(\mathbf{w}|0,\alpha^{-1}I) = \frac{\alpha}{2\pi}^{(M+1)/2}\exp(-\frac{\alpha}{2}\mathbf{w}^T\mathbf{w})$$

So we can get the posterior distribution which proportional to the product of the prior and likelihood function, then we would like to maximum the probability for the coefficient which called maximum posterior **MAP**. The finally solution is equally to minimum of:

$$\frac{\beta}{2}\sum_{n=1}^{N}(y(x,\mathbf{w}) - t_n)^2 + \frac{\alpha}{2}\mathbf{w}^T\mathbf{w}$$

which is equally to the ridge regression with regularization parameter give by $\lambda = \alpha/\beta$. This means that use Bayesian methods which can avoid over-fitting problem.

1.2.6 bayesian curve fitting

All above is do point estimation for the coefficients. while our goal is to do prediction and evaluate the predictive distribution $p(t|x,\mathbf{x},\mathbf{t})$. As we have the posterior and predictive distribution, we can get it with known parameters which can estimated from the data as empirical Bayesian:

$$p(t|x,\mathbf{x},\mathbf{t}) = \int p(t|x,\mathbf{w})p(\mathbf{w}|\mathbf{x},\mathbf{t})d\mathbf{w}$$

Taking the t as the random variable and other variable or parameters as known, on the right equation which is also one quadratic form for t, so it is given by one normal form:

$$p(t|x,\mathbf{X},\mathbf{t}) = N(t|m(x),s^2(x))m(x) = \beta\phi(x)^T SXt, \quad s^2(x) = \beta^{-1} + X^T SXS^{-1} = \alpha I + \beta X\otimes\phi(x)^T$$

1.3 Model Selection

In practical application, we need to determine the model complexity, number of parameters, to get the best predictive performance on the new data.

cross-validation, **leave-one-out**, both computationally expensive. The number of train sets will increase exponential with the hyper parameters increase.

information criteria: adding a penalty term to compensate for the over-fitting of more complex models, such as **AIC**, **BIC**. Meanwhile, it also need to consider the model complexity, simple maybe better enough.

1.4 The Curse of DImensionality

High dimensional.

For example, use the simplest way divide the input space into regular cells to do classification, which needs data point exponentially increased to make sure each cell has at least one point to do prediction.

Should be careful to generalize one conclusion from low dimensions to high dimension spaces.

Two reason to development effective techniques in high dimensional spaces: -1. confine of low dimensional space -2. local interpolation-like techniques

1.5 Decision Theory

General inference: determining the joint distribution which gives us the most complete probabilistic description of the situation.

How to make decision?

1.5.1 Minimizing the misclassification rate

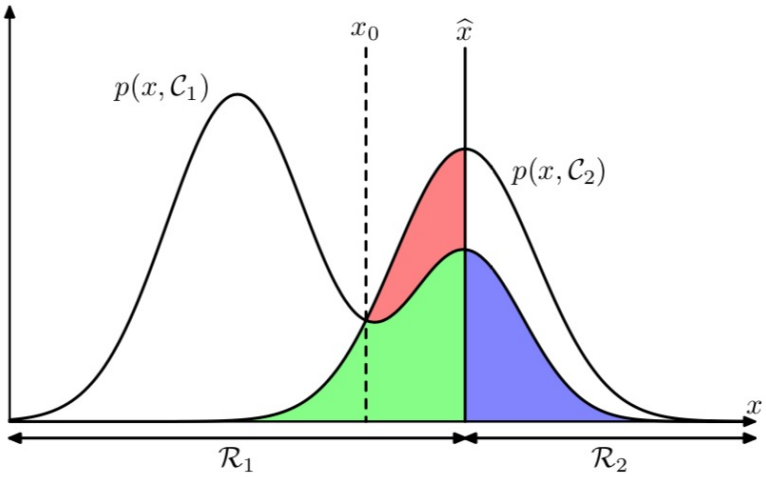


Figure 1.24 Schematic illustration of the joint probabilities $p(x, \mathcal{C}_k)$ for each of two classes plotted against x, together with the decision boundary $x = \widehat{x}$. Values of $x \geqslant \widehat{x}$ are classified as class $\mathcal{C}_2$ and hence belong to decision region $\mathcal{R}_2$, whereas points $x < \widehat{x}$ are classified as $\mathcal{C}_1$ and belong to $\mathcal{R}_1$. Errors arise from the blue, green, and red regions, so that for $x < \widehat{x}$ the errors are due to points from class $\mathcal{C}_2$ being misclassified as $\mathcal{C}_1$ (represented by the sum of the red and green regions), and conversely for points in the region $x \geqslant \widehat{x}$ the errors are due to points from class $\mathcal{C}_1$ being misclassified as $\mathcal{C}_2$ (represented by the blue region). As we vary the location $\widehat{x}$ of the decision boundary, the combined areas of the blue and green regions remains constant, whereas the size of the red region varies. The optimal choice for $\widehat{x}$ is where the curves for $p(x, \mathcal{C}_1)$ and $p(x, \mathcal{C}_2)$ cross, corresponding to $\widehat{x} = x_0$, because in this case the red region disappears. This is equivalent to the minimum misclassification rate decision rule, which assigns each value of x to the class having the higher posterior probability $p(\mathcal{C}_k|x)$.

Figure 2:

1.5.2 Minimizing the expected loss

Different classification has different consequence, which can describe by the loss matrix.

loss function or *cost function*, *unility function* negative of the loss.

True class fiction is unknown, so use the average loss to instead as equation 1.80. With the joint distribution can be show as posterior product the likelihoods which is common factor, so we calculate the posterior first.

1.5.3 The reject option

rejection criterion, based on the posterior probability, take θ as the threshold to do decision to minimize the expected loss.

Figure 1.26 Illustration of the reject option. Inputs x such that the larger of the two posterior probabilities is less than or equal to some threshold θ will be rejected.

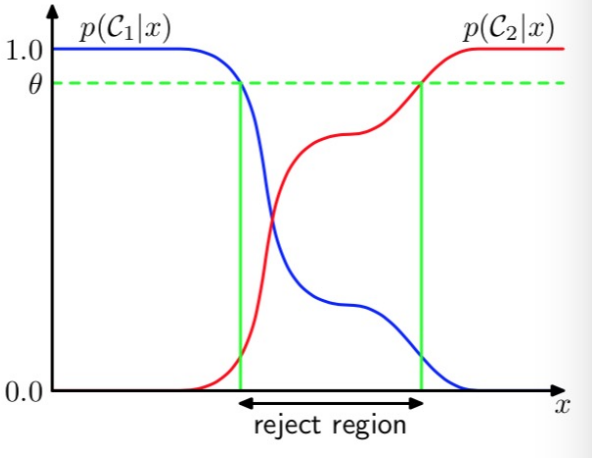


Figure 3:

1.5.4 Inference and decision.

Inference stage : using training data to learn a model; decision stage : using posterior probabilities make optimal decision.

It can also solve both problems together *discriminant function* which maps input directly into decisions.

Three Dsitint approach

-a. *generative model* model the distribution of input as well as output $p(x|C_k)$ and $p(x, C_k)$. Solving inference problem of each class-conditional densities or joint distribution to get the posterior distribution to do decision. This methods can get the joint distribution and marginal distribution which can use to deceit outlive which with lower probability, also known as novelty detection. It will cost too much computational resources if we only interest in the posterior probability.

-b. *discriminative models* model posterior probability directly $p(C_k|x)$.

-c.*discriminant function* maps each input x directly onto a class label, which combine inference and decision stage together.But we usually need to do decision on the posterior probability to *Minimizing risk*, *reject option*, *compensating for class priors* which need balanced data for each class to find a more accurate model, *combining models* such as naive Bayes model which assume conditional independence to combine two models.

1.5.5 Loss function for regression

Loss function,square loss such as square loss$L(t, y(x)) = (t - y(x))^2$, one general form *Minkowski loss*$L(t, y(x)) = |t - y(x)|^q$.

$$E(L) = \int \int L(t, y(x))p(x, t)dxdt$$

Goal is to minimize $E(L)$, it can be show with square loss, we have $y(x) = E_t[t|x]$. so regression function is the conditional average of t conditioned on x.

Figure 1.28 The regression function $y(x)$, which minimizes the expected squared loss, is given by the mean of the conditional distribution $p(t|x)$.

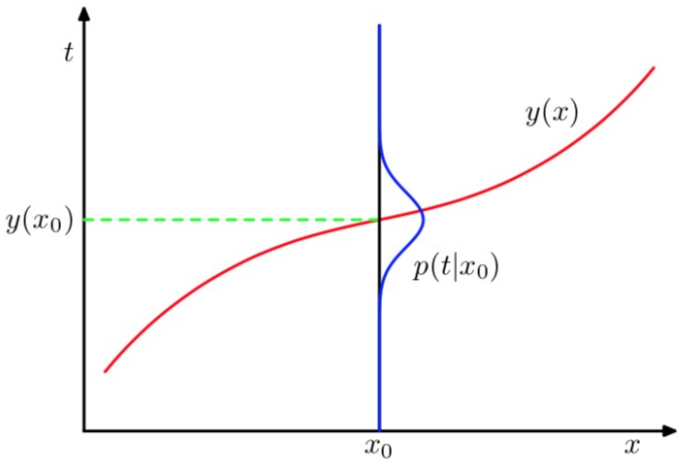


Figure 4:

$$E(L) = \int \{y(x) - E[t|x]\}^2 p(x)dx + \int \{E[t|x] - t\}^2 dx$$

which delete the interaction term which is equal to zero. to minimize the average loss, which can also derives $y(x) = E[t|x]$, the left term is independent to y which is the irreducible minimum value of the loss function can be regarded as noise.There also are three approaches for regression (Page 47-48).

1.6 Information theroy

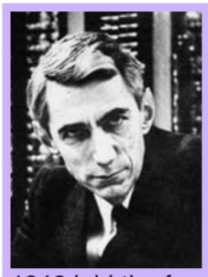
The amount of information can be viewed as 'degree of suprise' on learning the value of the value of one random variable.

Higher uncertainty, much more information. Average amount of information, **entropy** which in terms of disorder:

$$H[x] = -\sum_x p(x) \log p(x) \tag{1.93}$$

Nonuniform distribution has a smaller entropy then the uniform one. When x is categorical, equal distribution of probability across the possible states will get the maximum entropy $H = \ln M$. When x is continuous, with three contain equations 1.105-1.107, using Lagrange multipliers to maximum the entropy, can get the normal distribution with $H[x] = \frac{1+ln(2\pi\sigma^2)}{2}$. Unlike the discrete variable, entropy of the continuous variable can be negative.

conditional entropy: $H[y|x] = -\int \int p(y, x)lnp(y|x)dydx$.



Claude Shannon
1916–2001

After graduating from Michigan and MIT, Shannon joined the AT&T Bell Telephone laboratories in 1941. His paper 'A Mathematical Theory of Communication' published in the *Bell System Technical Journal* in 1948 laid the foundations for modern information theory. This paper introduced the word 'bit', and his concept that information could be sent as a stream of 1s and 0s paved the way for the communications revolution. It is said that von Neumann recommended to Shannon that he use the term entropy, not only because of its similarity to the quantity used in physics, but also because "nobody knows what entropy really is, so in any discussion you will always have an advantage".

Figure 5:

1.6.1 Relative entropy and mutual infromation

relative entropy or Pullback-Libeler divergence *KL divergence*.

$$KL[p(x)||q(x)] = -\int p(x) \ln \frac{q(x)}{p(x)} dx \tag{1.113}$$

which is non negative and take zero only if $p(x) = q(x)$, which can be proved by *Jensen's inequality*:

$$f(E(x)) \leqslant E(f(x)), \quad f \quad concave$$

We can try to approximate one unknown distribution p using some parametric distribution $q(x|\theta)$, which can be relied by minimizing the *KL divergence*. But we can not do it directly, for we don't know the true distribution $p(x)$, we only have finite data drawn from it. So the expectation with respect to $p(x)$ can be approximated by sum over these points:

$$KL(p||q) \simeq \sum_{n=1}^{N} \{-\ln(q(x_n|\theta) + \ln(p(x_n)\} \tag{1.119}$$

Thus we see that minimizing this KL divergence is equivalent to maximizing the likelihood function as the first term show on the above equation.

```
entropy <- function(x, f="Gaussian") {
  x1 <- unique(x)
  N  <- length(x)
  if(is.null(f)){
    t <- sapply(1:length(x1), function(i) sum(x1[i]==x))/N
    H <- -sum(t*log(t))
  }

  if(f == "Gaussian") H <- 0.5+log(2*pi*var(x))/2
  if(f == "uniform")  H <- log(length(x1))
  return(H)
}

x <- round(rnorm(100),1)
entropy(x, f="Gaussian")
```

```
## [1] 1.353561
```

```r
entropy(x, f="uniform")
```

```
## [1] 3.637586
```