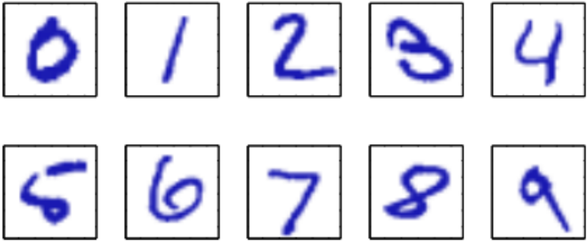
PRML Chapter I : Introduction

孙蕴哲

Sep 25th,2019

Introduction



- Training Set & Test Set & validation set
- Supervised learning & Unsupervised learning & Weakly-supervised learning &Semi-supervised learning
- Pre-process
- Classfication & Regression
- reinforcement learning

- Train Set & Test Set & **Validation Set**

- Train Set & Test Set & **Validation Set**
  - Train :A set of examples used for learning, which is to fit the parameters [i.e., weights] of the classifier.
  - Test:A set of examples used only to assess the performance [generalization] of a fully specified classifier.
  - **Validation**:A set of examples used to tune the parameters [i.e., architecture, not weights] of a classifier, for example to choose the number of hidden units in a neural network.

- Train Set & Test Set & **Validation Set**
  - Train :A set of examples used for learning, which is to fit the parameters [i.e., weights] of the classifier.
  - Test:A set of examples used only to assess the performance [generalization] of a fully specified classifier.
  - **Validation**:A set of examples used to tune the parameters [i.e., architecture, not weights] of a classifier, for example to choose the number of hidden units in a neural network.
- Supervised learning & Unsupervised learning & **weakly-supervised learning** & **Semi-supervised**

- Train Set & Test Set & **Validation Set**
  - Train :A set of examples used for learning, which is to fit the parameters [i.e., weights] of the classifier.
  - Test:A set of examples used only to assess the performance [generalization] of a fully specified classifier.
  - **Validation**:A set of examples used to tune the parameters [i.e., architecture, not weights] of a classifier, for example to choose the number of hidden units in a neural network.
- Supervised learning & Unsupervised learning & **weakly-supervised learning** & **Semi-supervised**
  - Supervised learning:Applications in which the training data comprises examples of the input vectors along with their corresponding target vectors.$\{ X_{data}, Y_{label}\}$ is known in advance.
  - Unsupervised learning :The goal in such unsupervised learning problems may be to discover groups of similar examples within the data.

- **Weakly-supervised learning**:
  - incomplete supervision:One (usually very small) subset of the training set is labeled, while the other data is not labeled.
  - inexact supervision: Image has only coarse-grained labels.
  - inaccurate supervision:The labels given by the model are not always true values.

- **Weakly-supervised learning**:
  - incomplete supervision:One (usually very small) subset of the training set is labeled, while the other data is not labeled.
  - inexact supervision: Image has only coarse-grained labels.
  - inaccurate supervision:The labels given by the model are not always true values.
- **Semi-supervised learning**:
  - Semi-supervised learning is to enable learners to use unlabeled samples to improve learning performance without relying on external interaction.

- Pre-process

- Pre-process
  - For most practical applications, the original input variables are typically preprocessed to transform them into some new space of variables where, it is hoped, the pattern recognition problem will be easier to solve.
  - Speed up computation.

- Pre-process
  - For most practical applications, the original input variables are typically preprocessed to transform them into some new space of variables where, it is hoped, the pattern recognition problem will be easier to solve.
  - Speed up computation.
- Classfication & Regression

- Pre-process
  - For most practical applications, the original input variables are typically preprocessed to transform them into some new space of variables where, it is hoped, the pattern recognition problem will be easier to solve.
  - Speed up computation.
- Classfication & Regression
  - **Quantitative** output is called regression, or **continuous** variable prediction.—Temperature prediction
  - **Qualitative** output is called classification, or **discrete** variable prediction. —Weather forecast

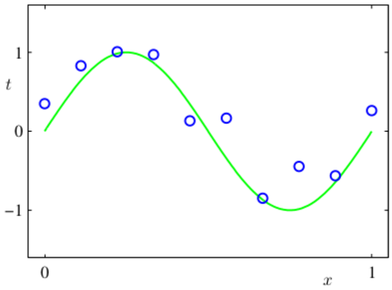
- Pre-process
  - For most practical applications, the original input variables are typically preprocessed to transform them into some new space of variables where, it is hoped, the pattern recognition problem will be easier to solve.
  - Speed up computation.
- Classfication & Regression
  - **Quantitative** output is called regression, or **continuous** variable prediction.—Temperature prediction
  - **Qualitative** output is called classification, or **discrete** variable prediction. —Weather forecast
- Reinforcement learning

- Pre-process
  - For most practical applications, the original input variables are typically preprocessed to transform them into some new space of variables where, it is hoped, the pattern recognition problem will be easier to solve.
  - Speed up computation.
- Classfication & Regression
  - **Quantitative** output is called regression, or **continuous** variable prediction.—Temperature prediction
  - **Qualitative** output is called classification, or **discrete** variable prediction. —Weather forecast
- Reinforcement learning

  The technique of reinforcement learning is concerned with the problem of finding suitable actions to take in a given situation in order to maximize a reward.

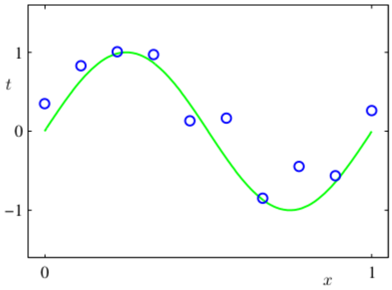
Section 1.1 Example:Polynomial Curve Fitting

Example:Polynomial Curve Fitting



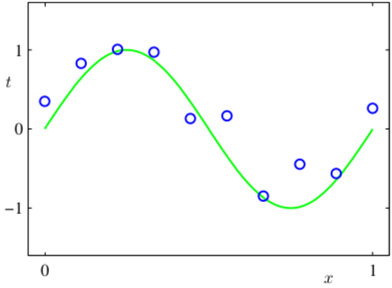
Our goal is to train a function $y(x)$ which takes a vector X as input and produce the prediction of Y.

Example:Polynomial Curve Fitting



Our goal is to train a function $y(x)$ which takes a vector X as input and produce the prediction of Y.
$y(x, w) = w_0 + w_1 x + w_2 x^2 + \ldots w_M x^M = \sum_{j=0}^{M} w_j x^j$

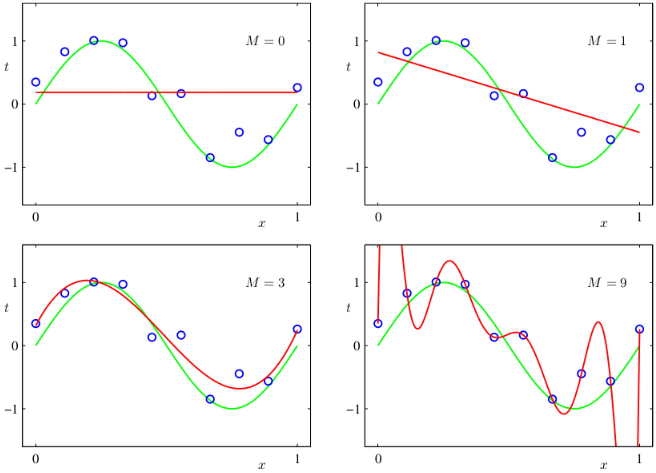
Example:Polynomial Curve Fitting

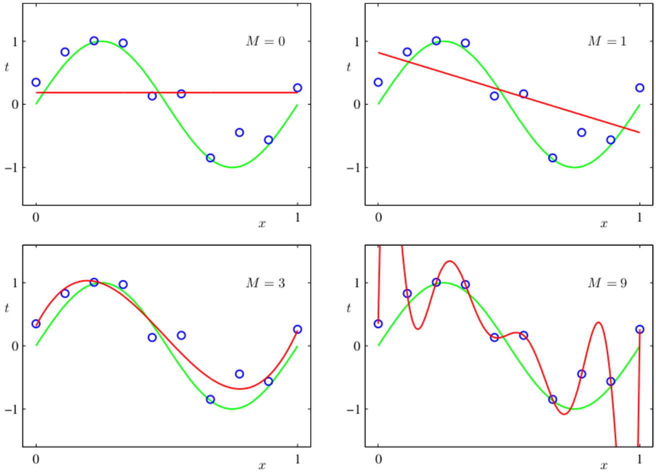


Our goal is to train a function $y(x)$ which takes a vector X as input and produce the prediction of Y.

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

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


M = 0
M = 1
M = 3
M = 9
t
x
1
0
−1
0
1



Model comparsion(Model Selection), **Over-fitting**, **Regularization**, a **Penalty Term** to the error function

	$M=0$	$M=1$	$M=3$	$M=9$
w_0^*	0.19	0.82	0.31	0.35
w_1^*		-1.27	7.99	232.37
w_2^*			-25.43	-5321.83
w_3^*			17.37	48568.31
w_4^*				-231639.30
w_5^*				640042.26
w_6^*				-1061800.52
w_7^*				1042400.18
w_8^*				-557682.99
w_9^*				125201.43

Error Function: $E(w) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n, w) - t_n\}^2 + \frac{\lambda}{2}\|w\|^2$

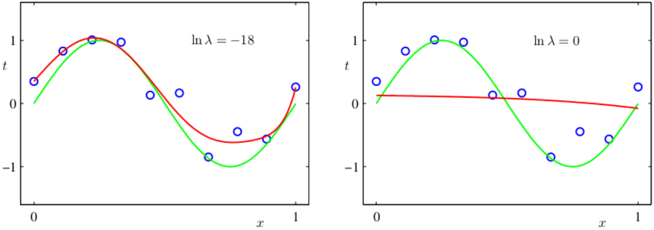
	$M=0$	$M=1$	$M=3$	$M=9$
w_0^*	0.19	0.82	0.31	0.35
w_1^*		-1.27	7.99	232.37
w_2^*			-25.43	-5321.83
w_3^*			17.37	48568.31
w_4^*				-231639.30
w_5^*				640042.26
w_6^*				-1061800.52
w_7^*				1042400.18
w_8^*				-557682.99
w_9^*				125201.43

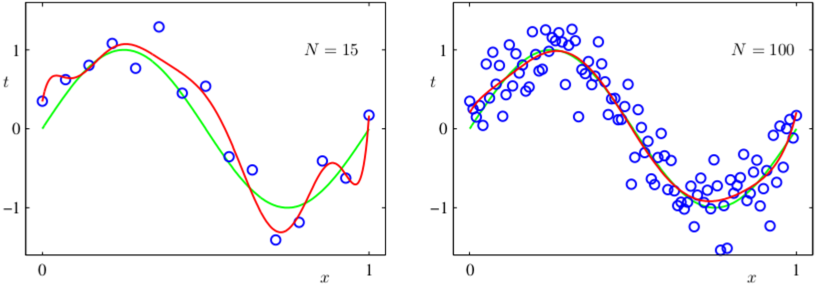
Error Function: $E(w) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n, w) - t_n\}^2 + \frac{\lambda}{2}\|w\|^2$
$\|w\|^2 = w^T w = {w_0}^2 + {w_1}^2 + ... + {w_M}^2$

	$M=0$	$M=1$	$M=3$	$M=9$
$w_0^\star$	0.19	0.82	0.31	0.35
$w_1^\star$		-1.27	7.99	232.37
$w_2^\star$			-25.43	-5321.83
$w_3^\star$			17.37	48568.31
$w_4^\star$				-231639.30
$w_5^\star$				640042.26
$w_6^\star$				-1061800.52
$w_7^\star$				1042400.18
$w_8^\star$				-557682.99
$w_9^\star$				125201.43

Error Function: $E(w) = \frac{1}{2}\sum_{n=1}^{N}\{y(x_n, w) - t_n\}^2 + \frac{\lambda}{2}\|w\|^2$

$\|w\|^2 = w^T w = w_0{}^2 + w_1{}^2 + ... + w_M{}^2$





- We see that increasing the size of the data set reduces the over-fitting problem.

Section 1.2 Probability Theory

Probability densities & Expectations and covariances

- joint probability : $P(X = x_i, Y = y_j)$
- marginal probability: $P(X = x_i) = \sum_{j=1}^{L} P(X = x_i, Y = y_j)$
- conditional probability : $P(Y = y_j | X = x_i)$
- **sum rule**: $P(X) = \sum_Y P(X, Y)$
- **product rule**: $P(X, Y) = P(Y|X)P(X)$
- probability densities : $p(x \in (a, b)) = \int_a^b p(x)dx$
- expections: $\mathbb{E}[f] = \sum_x p(x)f(x)$ & $\mathbb{E}[f] = \int p(x)f(x)dx$
- variances: $var[f] = \mathbb{E}[f(x)^2] - \mathbb{E}[f(x)]^2$
- covariance: $cov[x, y] = [\mathbb{E}_{x,y}\{x - \mathbb{E}[x]\}\mathbb{E}\{y - \mathbb{E}[y]\}]$

Bayesian probabilities

So far in this chapter, we have viewed probabilities in terms of the frequencies of random, repeatable events. We shall refer to this as the **classical** or **frequentist** interpretation of probability.

Bayesian probabilities

- **Prior probability**： cause −> effect
- **Posterior probability**： effect −> cause

Bayesian probabilities

- **Prior probability**： cause −> effect
- **Posterior probability**： effect −> cause

定义

$p(Y|X) = \frac{p(X|Y)p(Y)}{p(X)}$,$p(X) = \sum_Y p(X|Y)p(Y)$

Bayesian probabilities

- Teacher : Student = 1:1
  The probability of a teacher coming to the office on weekends is 0.2
  The probability of a student coming to the office on weekends is 0.6
- There's a man in the office on weekends, so what's the probability that he's a student?
- W–> T/S, X–> coming to office
- $p(W_1) = 0.5$ $p(W_2) = 0.5$

Bayesian probabilities

- Teacher : Student = 1:1
  The probability of a teacher coming to the office on weekends is 0.2
  The probability of a student coming to the office on weekends is 0.6
- There's a man in the office on weekends, so what's the probability that he's a student?
- W–> T/S, X–> coming to office
- $p(W_1) = 0.5$ $p(W_2) = 0.5$
- $p(X|W_1) = 0.2$,$p(X|W_2) = 0.6$

Bayesian probabilities

- Teacher : Student = 1:1
  The probability of a teacher coming to the office on weekends is 0.2
  The probability of a student coming to the office on weekends is 0.6
- There's a man in the office on weekends, so what's the probability that he's a student?
- W–> T/S, X–> coming to office
- $p(W_1) = 0.5$ $p(W_2) = 0.5$
- $p(X|W_1) = 0.2, p(X|W_2) = 0.6$
- $p(X) = p(X|W_1)p(W_1) + p(X|W_2)p(W_2) = 0.4$

Bayesian probabilities

- Teacher : Student = 1:1
  The probability of a teacher coming to the office on weekends is 0.2
  The probability of a student coming to the office on weekends is 0.6
- There's a man in the office on weekends, so what's the probability that he's a student?
- W–> T/S, X–> coming to office
- $p(W_1) = 0.5 \ p(W_2) = 0.5$
- $p(X|W_1) = 0.2, p(X|W_2) = 0.6$
- $p(X) = p(X|W_1)p(W_1) + p(X|W_2)p(W_2) = 0.4$
- $p(W_1|X) = \frac{p(X|W_1)P(W_1)}{p(X)} = 0.25$

Bayesian probabilities

- Teacher : Student = 1:1
  The probability of a teacher coming to the office on weekends is 0.2
  The probability of a student coming to the office on weekends is 0.6
- There's a man in the office on weekends, so what's the probability that he's a student?
- W–> T/S, X–> coming to office
- $p(W_1) = 0.5 \; p(W_2) = 0.5$
- $p(X|W_1) = 0.2, p(X|W_2) = 0.6$
- $p(X) = p(X|W_1)p(W_1) + p(X|W_2)p(W_2) = 0.4$
- $p(W_1|X) = \frac{p(X|W_1)P(W_1)}{p(X)} = 0.25$
- $p(W_2|X) = \frac{p(X|W_2)P(W_2)}{p(X)} = 0.75$

The Gaussian distribution

The most important probability distributions for continuous variables, called the normal or Gaussian distribution.

定义

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

The Gaussian distribution

The most important probability distributions for continuous variables, called the normal or Gaussian distribution.

定义

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

$$x = \{x_1, x_2 ... x_N\}$$

Independent and Identically Distributed

$$p(x_1, ...x_N|\mu,\sigma^2) = \prod_{n=1}^{N} N(x_n|\mu,\sigma^2)$$

The Gaussian distribution

The log likelihood function can be written in the form

$$\ln p(\mathbf{x}|\mu, \sigma^2) = -\frac{1}{2\sigma^2}\sum_{n=1}^{N}(x_n - \mu)^2 - \frac{N}{2}\ln\sigma^2 - \frac{N}{2}\ln(2\pi)$$

The Gaussian distribution

The log likelihood function can be written in the form

$$lnp(\mathbf{x}|\mu,\sigma^2) = -\frac{1}{2\sigma^2}\sum_{n=1}^{N}(x_n-\mu)^2 - \frac{N}{2}ln\sigma^2 - \frac{N}{2}ln(2\pi)$$

Maximizing with respect to μ:

$$\mu_{ML} = \frac{1}{N}\sum_{n=1}^{N}x_n$$

The Gaussian distribution

The log likelihood function can be written in the form

$$\mathit{lnp}(\mathbf{x}|\mu,\sigma^2) = -\frac{1}{2\sigma^2}\sum_{n=1}^{N}(x_n-\mu)^2 - \frac{N}{2}\mathit{ln}\sigma^2 - \frac{N}{2}\mathit{ln}(2\pi)$$

Maximizing with respect to μ:

$$\mu_{ML} = \frac{1}{N}\sum_{n=1}^{N} x_n$$

Maximizing with respect to σ^2:

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

The Gaussian distribution

Limitations of Maximum Likelihood Estimation: **bias**

The Gaussian distribution

Limitations of Maximum Likelihood Estimation: **bias**
Ex:

$$
\begin{aligned}
E(\mu_{ML}) &= E(\frac{1}{N}\sum_{n=1}^{N} x_n) \\
&= \frac{1}{N}E(\sum_{n=1}^{N} x_n) \\
&= \frac{1}{N}\sum_{n=1}^{N} E(x_n) \\
&= \frac{1}{N} * (N * \mu) = \mu
\end{aligned}
$$

The Gaussian distribution

Limitations of Maximum Likelihood Estimation: **bias**

Ex:

$$
\begin{aligned}
E(\mu_{ML}) &= E(\frac{1}{N}\sum_{n=1}^{N} x_n) \\
&= \frac{1}{N}E(\sum_{n=1}^{N} x_n) \\
&= \frac{1}{N}\sum_{n=1}^{N} E(x_n) \\
&= \frac{1}{N} * (N * \mu) = \mu
\end{aligned}
$$

Var:

$$E(\sigma_{ML}^2) = \frac{N-1}{N}\sigma^2$$

The Gaussian distribution

- So that on average the maximum likelihood estimate will obtain the correct mean but will underestimate the true variance by a factor $\frac{N-1}{N}$.

The Gaussian distribution

- So that on average the maximum likelihood estimate will obtain the correct mean but will underestimate the true variance by a factor $\frac{N-1}{N}$.
- Note that the bias of the maximum likelihood solution becomes less significant as the number N of data points increases, and in the limit $N \to \infty$ the maximum likelihood solution for the variance equals the true variance of the distribution that generated the data.

Curve fitting re-visited

We have seen how the problem of polynomial curve fitting can be expressed in terms of error minimization.

Curve fitting re-visited

We have seen how the problem of polynomial curve fitting can be expressed in terms of error minimization.

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

Curve fitting re-visited

We have seen how the problem of polynomial curve fitting can be expressed in terms of error minimization.

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

Here we return to the curve fitting example and view it from a probabilistic perspective, thereby gaining some insights into error functions and regularization, as well as taking us towards a full Bayesian treatment.

Curve fitting re-visited

We have seen how the problem of polynomial curve fitting can be expressed in terms of error minimization.

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

Here we return to the curve fitting example and view it from a probabilistic perspective, thereby gaining some insights into error functions and regularization, as well as taking us towards a full Bayesian treatment.

- trainind data: $\mathbf{x} = (x_1, ...x_N)$ and t $= (t_1, ...t_N)$
- it is assumed that t is Gaussian:

Curve fitting re-visited

We have seen how the problem of polynomial curve fitting can be expressed in terms of error minimization.

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

Here we return to the curve fitting example and view it from a probabilistic perspective, thereby gaining some insights into error functions and regularization, as well as taking us towards a full Bayesian treatment.

- trainind data: $\mathbf{x} = (x_1, ...x_N)$ and t $= (t_1, ...t_N)$
- it is assumed that t is Gaussian:
  $p(t|x, w, \beta) = N(t|y(x, w), \beta^{-1})$

Curve fitting re-visited

We have seen how the problem of polynomial curve fitting can be expressed in terms of error minimization.

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

Here we return to the curve fitting example and view it from a probabilistic perspective, thereby gaining some insights into error functions and regularization, as well as taking us towards a full Bayesian treatment.

- trainind data: $\mathbf{x} = (x_1, ...x_N)$ and $\mathrm{t} = (t_1, ...t_N)$
- it is assumed that t is Gaussian: $p(t|x, w, \beta) = N(t|y(x, w), \beta^{-1})$
- **precision parameter** β corresponding to the inverse variance of the distribution.

Curve fitting re-visited

$$p(\mathbf{t} \mid \mathbf{x}, \boldsymbol{w}, \beta) = \prod_{n=1}^{N} \mathcal{N}(t_n \mid y(x_n, \boldsymbol{w}), \beta^{-1})$$

Curve fitting re-visited

$$p(\mathbf{t} \mid \mathbf{x}, \boldsymbol{w}, \beta) = \prod_{n=1}^{N} \mathcal{N}(t_n \mid y(x_n, \boldsymbol{w}), \beta^{-1})$$

$$\ln p(\mathbf{t} \mid \mathbf{x}, \boldsymbol{w}, \beta) = -\frac{\beta}{2} \sum_{n=1}^{N} \{y(x_n, \boldsymbol{w}) - t_n\}^2 + \frac{N}{2} \ln \beta - \frac{N}{2} \ln(2\pi)$$

Curve fitting re-visited

$$p(\mathbf{t} \mid \mathbf{x}, \boldsymbol{w}, \beta) = \prod_{n=1}^{N} \mathcal{N}(t_n \mid y(x_n, \boldsymbol{w}), \beta^{-1})$$

$$\ln p(\mathbf{t} \mid \mathbf{x}, \boldsymbol{w}, \beta) = -\frac{\beta}{2} \sum_{n=1}^{N} \{y(x_n, \boldsymbol{w}) - t_n\}^2 + \frac{N}{2} \ln \beta - \frac{N}{2} \ln(2\pi)$$

$$\frac{1}{\beta_{ML}} = \frac{1}{N} \sum_{n=1}^{N} \{y(x_n, \boldsymbol{w}_{ML}) - t_n\}^2$$

Curve fitting re-visited

$$p(\mathbf{t} \mid \mathbf{x}, \boldsymbol{w}, \beta) = \prod_{n=1}^{N} \mathcal{N}(t_n \mid y(x_n, \boldsymbol{w}), \beta^{-1})$$

$$\ln p(\mathbf{t} \mid \mathbf{x}, \boldsymbol{w}, \beta) = -\frac{\beta}{2} \sum_{n=1}^{N} \{y(x_n, \boldsymbol{w}) - t_n\}^2 + \frac{N}{2} \ln \beta - \frac{N}{2} \ln(2\pi)$$

$$\frac{1}{\beta_{ML}} = \frac{1}{N} \sum_{n=1}^{N} \{y(x_n, \boldsymbol{w}_{ML}) - t_n\}^2$$

$$p(t \mid x, \boldsymbol{w}_{ML}, \beta_{ML}) = \mathcal{N}(t \mid y(x, \boldsymbol{w}_{ML}), \beta_{ML}^{-1})$$

Curve fitting re-visited

- The polynomial coefficients are treated as random variables with a Gaussian distribution taken over a vector of dimension M+1:

$$p(\boldsymbol{w} \mid \alpha) = \mathcal{N}(\boldsymbol{w} \mid \mathbf{0}, \alpha^{-1}\boldsymbol{I}) = \left(\frac{\alpha}{2\pi}\right)^{\frac{M+1}{2}} \exp\left\{-\frac{\alpha}{2}\boldsymbol{w}^T\boldsymbol{w}\right\}$$

Curve fitting re-visited

- The polynomial coefficients are treated as random variables with a Gaussian distribution taken over a vector of dimension M+1:

$$p(\boldsymbol{w} \mid \alpha) = \mathcal{N}(\boldsymbol{w} \mid \mathbf{0}, \alpha^{-1}\boldsymbol{I}) = \left(\frac{\alpha}{2\pi}\right)^{\frac{M+1}{2}} \exp\left\{-\frac{\alpha}{2}\boldsymbol{w}^T\boldsymbol{w}\right\}$$

- from Bayes we get the posterior probability:
  $p(w|\mathbf{x},\mathbf{t},\alpha,\beta) \propto p(\mathbf{t}|\mathbf{x},w,\beta)p(w|\alpha)$

Curve fitting re-visited

- The polynomial coefficients are treated as random variables with a Gaussian distribution taken over a vector of dimension M+1:

$$p(\boldsymbol{w} \mid \alpha) = \mathcal{N}(\boldsymbol{w} \mid \mathbf{0}, \alpha^{-1}\boldsymbol{I}) = \left(\frac{\alpha}{2\pi}\right)^{\frac{M+1}{2}} \exp\left\{-\frac{\alpha}{2}\boldsymbol{w}^T\boldsymbol{w}\right\}$$

- from Bayes we get the posterior probability: $p(w|\mathbf{x},\mathbf{t},\alpha,\beta) \propto p(\mathbf{t}|\mathbf{x},w,\beta)p(w|\alpha)$
- **maximum posterior** or **MAP**.We take the negative logarithm,we throw out constant terms and we get:

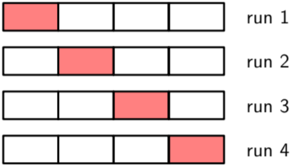
$$\frac{\beta}{2}\sum_{n=1}^{N}\{t_n - y(x_n,\boldsymbol{w})\}^2 + \frac{\alpha}{2}\boldsymbol{w}^\top\boldsymbol{w}$$

Section 1.3 Model Selection

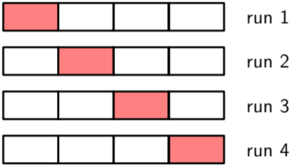
Model Selection : Cross validation

- With regularized least squares, the regularization coefficient λ also controls the effective complexity of the model.
- For more complex models, such as mixture distributions or neural networks there may be multiple parameters governing complexity.
- we need to determine the values of such parameters, and the principal objective in doing so is usually to achieve the best predictive performance on new data.
- Furthermore, as well as finding the appropriate values for complexity parameters within a given model, we may wish to consider a range of different types of model in order to find the best one for our particular application.
- It may be necessary to keep aside a third test set on which the performance of the selected model is finally evaluated.

Cross Validation



Cross Validation

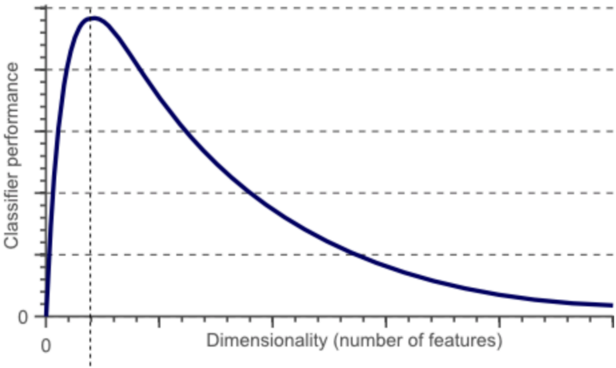


- When data is particularly scarce, it may be appropriate to consider the case $S = N$, where N is the total number of data points, which gives the **leave-one-out** technique.
- One major drawback of cross-validation is that the number of training runs that must be performed is increased by a factor of S.

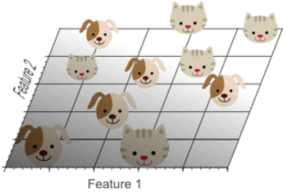
Section 1.4 The Curse of Dimensionality

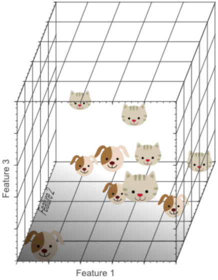
The Curse of Dimensionality

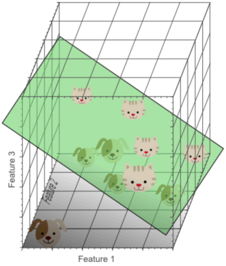
When reading machine learning papers, we often see that some writers refer to "curse of dimensionality", What kind of "disaster" is it? what is its importance in classification.

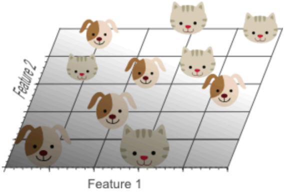


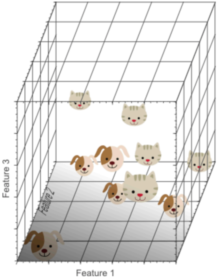
the Curse of Dimensionality

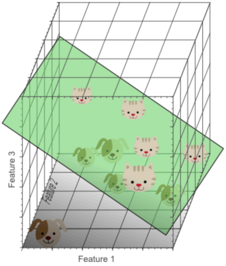












- **Sample density**
- **Feature space**

Section 1.5 Decision Theory

Decision Theory

- The decision problem:
  - **classification problem**
  - given x,predict t according to a probablistic model $p(x, t)$

Decision Theory

- The decision problem:
  - **classification problem**
  - given x,predict t according to a probablistic model $p(x,t)$
- Two basic requirements:
  - The probability distribution of the population of each category is known.
  - The number of categories to be classified is certain.

Decision Theory

- The decision problem:
  - **classification problem**
  - given x,predict t according to a probablistic model $p(x,t)$
- Two basic requirements:
  - The probability distribution of the population of each category is known.
  - The number of categories to be classified is certain.
- Important quantity:$p(C_k|x)$

$$p(C_k|x) = \frac{p(x,C_k)}{p(x)} = \frac{p(x,C_k)}{\sum_{k=1}^{2} p(x,C_k)}$$

- Intution:choose k that maximizes $p(C_k|x)$

Decision Theory-Minimizing the misclassification rate

- **Decision region:** $R_i = \{\ x : \mathit{pred}(x) = C_i\ \}$
- **Decision Boundaries or Decision Surfaces**： The boundaries between decision regions

Decision Theory-Minimizing the misclassification rate

- **Decision region:** $R_i = \{\ x : \textit{pred}(x) = C_i\ \}$
- **Decision Boundaries or Decision Surfaces**： The boundaries between decision regions
- Note that each decision region need not be contiguous but could comprise some number of disjoint regions.

Decision Theory-Minimizing the misclassification rate

- **Decision region:** $R_i = \{\ x : \textit{pred}(x) = C_i\ \}$
- **Decision Boundaries or Decision Surfaces**: The boundaries between decision regions
- Note that each decision region need not be contiguous but could comprise some number of disjoint regions.
- $p(mistake) = p(x \in R_1, C_2) + p(x \in R_2, C_1)$
  $= \int_{R_1} p(x, C_2)dx + \int_{R_2} p(x, C_1)dx$

Decision Theory-Minimizing the misclassification rate

- **Decision region:** $R_i = \{\ x : \textit{pred}(x) = C_i\ \}$
- **Decision Boundaries or Decision Surfaces**： The boundaries between decision regions
- Note that each decision region need not be contiguous but could comprise some number of disjoint regions.
- $p(mistake) = p(x \in R_1, C_2) + p(x \in R_2, C_1)$
  $= \int_{R_1} p(x, C_2)dx + \int_{R_2} p(x, C_1)dx$
- In order to minimize,affect x to $\mathcal{R}_1$ if:

$$
\begin{aligned}
& p(x, C_1) > p(x, C_2) \\
& \Leftrightarrow p(C_1|x)p(x) > p(C_2|x)p(x) \\
& \Leftrightarrow p(C_1|x) > p(C_2|x)
\end{aligned}
$$

Decision Theory-Minimizing the misclassification rate

- **Decision region:** $R_i = \{\ x : pred(x) = C_i\ \}$
- **Decision Boundaries or Decision Surfaces**: The boundaries between decision regions
- Note that each decision region need not be contiguous but could comprise some number of disjoint regions.
- $p(mistake) = p(x \in R_1, C_2) + p(x \in R_2, C_1)$
  $= \int_{R_1} p(x, C_2)dx + \int_{R_2} p(x, C_1)dx$
- In order to minimize,affect x to $\mathcal{R}_1$ if:

$$
\begin{aligned}
& p(x, C_1) > p(x, C_2) \\
& \Leftrightarrow p(C_1|x)p(x) > p(C_2|x)p(x) \\
& \Leftrightarrow p(C_1|x) > p(C_2|x)
\end{aligned}
$$

- $p(correct) = p(x \in R_x, C_x)$

Decision Theory-Minimizing the expected loss

- Suppose that, for a new value of x, the true class is C_k and that we assign x to class C_j (where j may or may not be equal to k).
- In so doing, we incur some level of loss that we denote by L_{kj} , which we can view as the k, j element of a loss matrix.
- **Cost**/**Loss** of a decision:L_{kj} = predict C_j while truth is C_k

$$\begin{array}{cc} & \begin{array}{cc} \text{cancer} & \text{normal} \end{array} \\ \begin{array}{c} \text{cancer} \\ \text{normal} \end{array} & \begin{pmatrix} 0 & 1000 \\ 1 & 0 \end{pmatrix} \end{array}$$

$$\mathbb{E}(L) = \sum_k \sum_j \int_{R_j} L_{kj} p(\mathbf{x}, C_k) dx$$

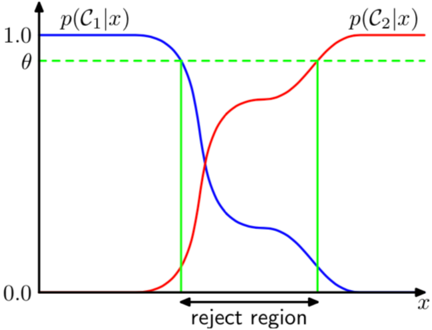
$$\mathbb{E}[L] = \int_{R2} L_{1,2} p(x, C_1) + \int_{R1} L_{2,1} p(x, C_2)$$

$$\sum L_{kj} p(C_k|x)$$

The reject option

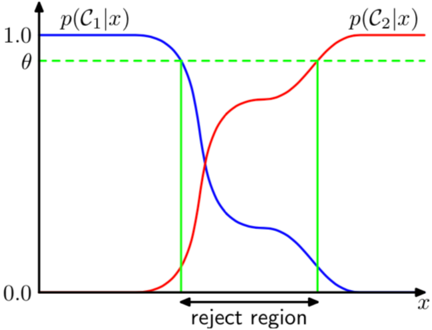
- For the 0/1 loss,pred(x) $=$ $argmax_k p(C_k|x)$
  - Note: K classes–> $1/K \leq maxp(C_k|x) \leq 1$
- When max $p(C_k|x)$ –>1/K the confidence in the prediction decreases

- For the 0/1 loss,pred(x) = $argmax_k p(C_k|x)$
  - Note: K classes–> $1/K \leq maxp(C_k|x) \leq 1$
- When max $p(C_k|x)$ –>$1/K$ the confidence in the prediction decreases
- **Reject option**



The reject option

- For the 0/1 loss,pred(x) = $argmax_k p(C_k|x)$
  - Note: K classes–> $1/K \leq maxp(C_k|x) \leq 1$
- When max $p(C_k|x)$ –>1/K the confidence in the prediction decreases
- **Reject option**



- Motivation:switch between automatic/human decision

Inference and decision

2 (or 3) different approaches to the decision problem:

Inference and decision

2 (or 3) different approaches to the decision problem:

1. rely on probabilistic model,with 2 flavours:
   1. generative
      - ★ use a generative model to infer $p(x|C_k)$
      - ★ combine with priors $p(C_k)$ to get $p(x, C_k)$ and eventually $p(C_k|x)$

Inference and decision

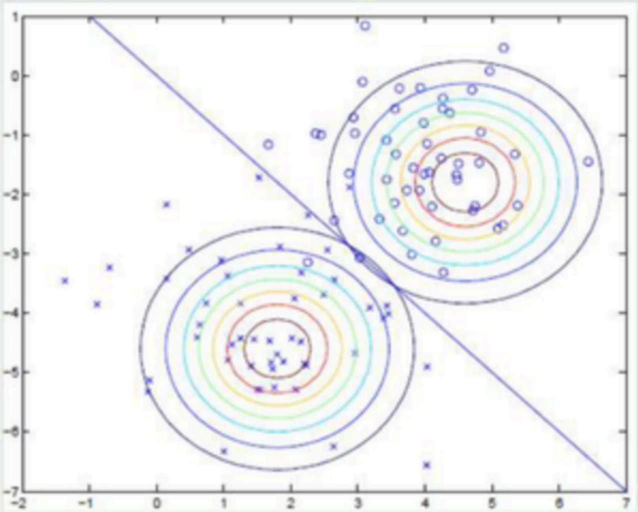
2 (or 3) different approaches to the decision problem:

1. rely on probabilistic model,with 2 flavours:
   1. generative
      - ★ use a generative model to infer $p(x|C_k)$
      - ★ combine with priors $p(C_k)$ to get $p(x, C_k)$ and eventually $p(C_k|x)$
   2. discriminative : infer directly $p(C_k|x)$
      - ★ this is sufficient for the decision problem

Inference and decision

2 (or 3) different approaches to the decision problem:

1. rely on probabilistic model,with 2 flavours:
   1. generative
      - ★ use a generative model to infer $p(x|C_k)$
      - ★ combine with priors $p(C_k)$ to get $p(x, C_k)$ and eventually $p(C_k|x)$
   2. discriminative : infer directly $p(C_k|x)$
      - ★ this is sufficient for the decision problem
2. learn a discriminant funtion $f(x)$
   - directly map input to class labels
   - for binary classification, $f(x)$ is typically defined as the sign $(+1/\text{-}1)$ of an auxiliary funtion


1
0
−1
−2
−3
−4
−5
−6
−7
−2
−1
0
1
2
3
4
5
6
7

Inference and decision

- probabilistic generative models:
  - pros:access to $p(x)$ $->$ easy detection of outliers
  - cons:estimation the joint probability $p(x, C_k)$ can be computational and data demanding.
- probabilistic discrimative models:
  - pros:less demanding than the generative approach
- discriminant functions:
  - pros:a single learning problem
  - cons:no access to $p(C_k|x)$

Loss functions for regression

- The regression setting:quantitative target t $\in R$
- Typical regression loss-function : $L(t, y(x)) = (y(x) - t)^2$
  - the **squared loss**
- The decision problem = minimize the expected loss:

$$\mathbb{E}[L] = \int_X \int_{\mathcal{R}} L(t, y(x)) p(x, t) dx dt$$

- Note: general class of loss functions $L(x, y(x)) = |y(x) - t|^2$

Section 1.6 Information Theory

Information Theory-Entropy

- Consider a discrete random variable X
- We want to define a measure $h(x)$ of **superise/information** of observing $X = x$

Information Theory-Entropy

- Consider a discrete random variable X
- We want to define a measure $h(x)$ of **superise/information** of observing $X = x$
- Natrual requirements:
  - if $p(x)$ is low (resp. high),$h(x)$ shoule be high(resp.low)
  - if X and Y are unrelated,$h(x, y)$ shoule be $h(x)+h(y)$
  - this leads to $h(x) = -\log p(x)$

Information Theory-Entropy

- Consider a discrete random variable X
- We want to define a measure $h(x)$ of **superise/information** of observing $X = x$
- Natrual requirements:
  - if $p(x)$ is low (resp. high),$h(x)$ shoule be high(resp.low)
  - if X and Y are unrelated,$h(x, y)$ shoule be $h(x)+h(y)$
  - this leads to $h(x) = -\log p(x)$
- **Entropy** of the variable X:

$$H[X] = E[h(x)] = -\sum_x p(x) \log(p(x)) \text{ bits}$$

Information Theory-Entropy

- Consider a discrete random variable X
- We want to define a measure $h(x)$ of **superise/information** of observing $X = x$
- Natrual requirements:
  - if $p(x)$ is low (resp. high),$h(x)$ shoule be high(resp.low)
  - if X and Y are unrelated,$h(x, y)$ shoule be $h(x)+h(y)$
  - this leads to $h(x) = -\log p(x)$
- **Entropy** of the variable X:

$$H[X] = E[h(x)] = -\sum_x p(x)\log(p(x)) \text{ bits}$$

(Convention: $p\log p = 0$ if p=0, $\log$–>$\ln$(in nats))

Information Theory-Entropy

- Consider a random variable x having 8 possible states, each of which is equally likely. In order to communicate the value of x to a receiver, we would need to transmit a message of length 3 bits.

Information Theory-Entropy

- Consider a random variable x having 8 possible states, each of which is equally likely. In order to communicate the value of x to a receiver, we would need to transmit a message of length 3 bits.

$$H[x] = -8 * \frac{1}{8} \log \frac{1}{8} = 3 \text{ bits}$$

Information Theory-Entropy

- Consider a random variable x having 8 possible states, each of which is equally likely. In order to communicate the value of x to a receiver, we would need to transmit a message of length 3 bits.

$$H[x] = -8*\frac{1}{8}\log\frac{1}{8} = 3 \text{ bits}$$

- $\{a, b, c, d, e, f, g, h\}$: $\{\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{64}, \frac{1}{64}, \frac{1}{64}, \frac{1}{64}\}$

$$H[x] = -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{4}\log_2\frac{1}{4} - \frac{1}{8}\log_2\frac{1}{8} - \frac{1}{16}log_2\frac{1}{16} - \frac{4}{64}\log_2\frac{1}{64} = 2 \text{ bits}$$

Information Theory-Entropy

- Consider a random variable x having 8 possible states, each of which is equally likely. In order to communicate the value of x to a receiver, we would need to transmit a message of length 3 bits.

$$H[x] = -8*\frac{1}{8}\log\frac{1}{8} = 3 \text{ bits}$$

- $\{a, b, c, d, e, f, g, h\}$: $\{\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{64}, \frac{1}{64}, \frac{1}{64}, \frac{1}{64}\}$

$$H[x] = -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{4}\log_2\frac{1}{4} - \frac{1}{8}\log_2\frac{1}{8} - \frac{1}{16}log_2\frac{1}{16} - \frac{4}{64}\log_2\frac{1}{64} = 2 \text{ bits}$$

- Consider how we would transmit the identity of the variable' s state to a receiver:
    - 0, 10, 110, 1110, 111100, 111101, 111110, 111111
    - average code length: $\frac{1}{2}x1 + \frac{1}{4}x2 + \frac{1}{8}x3 + \frac{1}{16}x4 + 4x\frac{1}{64}x6$ =2 bits

Information Theory-Entropy

- Consider a random variable x having 8 possible states, each of which is equally likely. In order to communicate the value of x to a receiver, we would need to transmit a message of length 3 bits.

$$H[x] = -8{*}\tfrac{1}{8}\log\tfrac{1}{8} = 3 \text{ bits}$$

- $\{a, b, c, d, e, f, g, h\}$: $\{\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{64}, \frac{1}{64}, \frac{1}{64}, \frac{1}{64}\}$

$$H[x] = -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{4}\log_2\frac{1}{4} - \frac{1}{8}\log_2\frac{1}{8} - \frac{1}{16}log_2\frac{1}{16} - \frac{4}{64}\log_2\frac{1}{64} = 2 \text{ bits}$$

- Consider how we would transmit the identity of the variable' s state to a receiver:
  - 0, 10, 110, 1110, 111100, 111101, 111110, 111111
  - average code length: $\frac{1}{2}x1 + \frac{1}{4}x2 + \frac{1}{8}x3 + \frac{1}{16}x4 + 4x\frac{1}{64}x6$ =2 bits
- The **noiseless coding theorementropy** is a lower bound on the number of bits needed to transmit the state of a random variable.

Information Theory-Entropy

- Differential entropy:

$$H[X] = -\int p(x) \ln p(x) dx$$

Relative entropy and mutual information

- Consider some unknown distribution $p(x)$, and suppose that we have modeled this using an approximating distribution $q(x)$.

Relative entropy and mutual information

- Consider some unknown distribution $p(x)$, and suppose that we have modeled this using an approximating distribution $q(x)$.
- If we use $q(x)$ to construct a coding scheme for the purpose of transmitting values of x to a receiver,
- then the average additional amount of information (in nats) required to specify the value of x as a result of using $q(x)$ instead of the true distribution $p(x)$ is given by:

Relative entropy and mutual information

- Consider some unknown distribution $p(x)$, and suppose that we have modeled this using an approximating distribution $q(x)$.
- If we use $q(x)$ to construct a coding scheme for the purpose of transmitting values of x to a receiver,
- then the average additional amount of information (in nats) required to specify the value of x as a result of using $q(x)$ instead of the true distribution $p(x)$ is given by:

$$\begin{aligned} \mathrm{KL}(p\|q) &= -\int p(\mathbf{x}) \ln q(\mathbf{x})\, \mathrm{d}\mathbf{x} - \left(-\int p(\mathbf{x}) \ln p(\mathbf{x})\, \mathrm{d}\mathbf{x}\right) \\ &= -\int p(\mathbf{x}) \ln \left\{\frac{q(\mathbf{x})}{p(\mathbf{x})}\right\} \mathrm{d}\mathbf{x}. \end{aligned}$$

- **relative entropy | Kullback-Leibler divergence**

Relative entropy and mutual information

- Consider some unknown distribution $p(x)$, and suppose that we have modeled this using an approximating distribution $q(x)$.
- If we use $q(x)$ to construct a coding scheme for the purpose of transmitting values of x to a receiver,
- then the average additional amount of information (in nats) required to specify the value of x as a result of using $q(x)$ instead of the true distribution $p(x)$ is given by:

$$\begin{aligned} \mathrm{KL}(p\|q) &= -\int p(\mathbf{x}) \ln q(\mathbf{x}) \, \mathrm{d}\mathbf{x} - \left(-\int p(\mathbf{x}) \ln p(\mathbf{x}) \, \mathrm{d}\mathbf{x}\right) \\ &= -\int p(\mathbf{x}) \ln \left\{ \frac{q(\mathbf{x})}{p(\mathbf{x})} \right\} \mathrm{d}\mathbf{x}. \end{aligned}$$

- **relative entropy | Kullback-Leibler divergence**
- $KL(p\|q) \not\equiv KL(q\|p)$

Relative entropy and mutual information

- Kullback-Leibler divergence satisfies $KL(\mathrm{p}\|\mathrm{q}) \geq 0$ with equality if, and only if, $p(x) = q(x)$.

Relative entropy and mutual information

- Kullback-Leibler divergence satisfies $KL(\mathrm{p}\|\mathrm{q}) \geq 0$ with equality if, and only if, $p(x) = q(x)$.
- A function os convex iff every cord lies above the function
- Any value of x in the interval from $x = a$ to $x = b$ can be written in the form $\lambda a + (1 - \lambda)b$ where $0 \leq \lambda \leq 1$.

$$f(\lambda a + (1 - \lambda)b) \leq \lambda f(a) + (1 - \lambda)f(b)$$

- strictly convex function:if the equality is satisfied only for $\lambda = 0$ and $\lambda = 1$.
- Jensen's inequality for convex functions:

$$E[f(x)] \geq f(E[x])$$

$$f\left(\int \boldsymbol{x}p(\boldsymbol{x})\ \mathrm{d}\boldsymbol{x}\right) \leq \int f(\boldsymbol{x})p(\boldsymbol{x})\ \mathrm{d}\boldsymbol{x}$$

- When applied to $KL(p\|q)$:

$$
\begin{aligned}
KL(p\|q) &= -\int p(x)\ln\frac{q(x)}{p(x)}dx \\
&> -\ln\int p(x)\times\frac{q(x)}{p(x)}dx \quad \text{(because } -\ln \text{ is stricly convex)} \\
&= -\ln\int q(x)dx = -\ln 1 = 0
\end{aligned}
$$

Moreover,straightforward to see that $KL(p\|p)$ =0

Mutual information

- Consider the joint distribution between two sets of variables x and y given by p(x, y).
- If the sets of variables are independent, p(x, y) = p(x)p(y).
- If the variables are not independent, we can gain some idea of whether they are ‘close’ to being independent by considering the Kullback-Leibler divergence between the joint distribution and the product of the marginals

$$\begin{aligned} \mathrm{I}[\mathbf{x}, \mathbf{y}] &\equiv \mathrm{KL}(p(\mathbf{x}, \mathbf{y}) \| p(\mathbf{x}) p(\mathbf{y})) \\ &= -\iint p(\mathbf{x}, \mathbf{y}) \ln \left(\frac{p(\mathbf{x}) p(\mathbf{y})}{p(\mathbf{x}, \mathbf{y})} \right) \mathrm{d}\mathbf{x}\, \mathrm{d}\mathbf{y} \end{aligned}$$

Mutual information

- From the properties of the Kullback-Leibler divergence, we see that I(x,y) $\geq$ 0 with equality if, and only if, x and y are independent.
- Using the sum and product rules of probability, we see that the mutual information is related to the conditional entropy through

$$I[\boldsymbol{x}, \boldsymbol{y}] = H[\boldsymbol{x}] - H[\boldsymbol{x} \mid \boldsymbol{y}] = H[\boldsymbol{y}] - H[\boldsymbol{y} \mid \boldsymbol{x}]$$

- Thus we can view the mutual information as the reduction in the uncertainty about x by virtue of being told the value of y (or vice versa). From a Bayesian perspective, we can view p(x) as the prior distribution for x and p(x|y) as the posterior distribution after we have observed new data y. The mutual information therefore represents the reduction in uncertainty about x as a consequence of the new observation y.