Statistics

Jacob Denson

January 25, 2023

Table Of Contents

Chapter 1

Statistical Models

Statistics is the theory of inferring features of some probability distribution μ on a set S (normally S is discrete, or equal to $\mathbf{R}^d$ for some d), given a number of independent samples drawn from this probability distribution. In general, it is very difficult to determine properties of μ, so we often work with some assumptions on the distribution μ, a *statistical model*. More precisely, a statistic model $\mathcal{F}$ is a subset of the space of all distributions on a set S. We might write such a set as

$$\mathcal{F} = \{\mu_\theta : \theta \in \Theta\},$$

where Θ is the *parameter space* of the model. If Θ can be seen naturally as a subset of $\mathbf{R}^n$ for some n, we say this is a *parametric model*.

Example. *In many statistical problems, we commonly assume the distribution is normal. That is, we work over the statistical model*

$$\{N(\mu, \Sigma) : \mu \in \mathbf{R}^n, \Sigma \in \mathbf{R}^{n \times n}\}.$$

Thus a normal distribution in $\mathbf{R}^n$ is specifiable by $n^2 + n$ parameters.

Example. *A standard class in nonparametric statistics is the class of* all distributions. *But some methods might assume some minor regularity about the distribution. For instance, we might work in the model $\mathcal{F}_{ABS}$, which consists of all probability measures μ which are absolutely continuous (i.e. that have a density function). Or perhaps some additional regularity, i.e. that μ lies in some function space, i.e. a Sobolev space.*

A *statistical functional* is any function of the distribution μ. Examples include the *expected value* of the distribution, the *variance*, or the *median* (only defined for probability measures without atoms). A *statistic* is a function of the data drawn from this distribution. We can now formally describe the problem of statistics: given a statistical functional, find a statistic which best approximates this functional. This is the problem of *point estimation*. The methods involved depend on the statistical functional employed.

For notational purposes, in statistics we often consider a multitude of distributions μ over the same set S. We therefore introduce the notation $\mathbf{P}_\mu$, $\mathbf{E}_\mu$, and $\mathbf{V}_\mu$, to denote the probabilistic quantities obtained by using the distribution μ. For parametric problems, we might also use the notation $\mathbf{P}_\theta$, $\mathbf{E}_\theta$, and $\mathbf{V}_\theta$.

One of the most important point estimation problem is the problem of *regression*, *prediction*, and *classification*, we are given pairs

$$(X_1, Y_1), \dots, (X_N, Y_N)$$

drawn independently from some product distribution μ. Given μ and some *error*, or *loss function*, we have a given *regression function* $Y = R(X)$, which best approximates the relationship between X and Y given by the distribution μ. Finding a good approximation of R via the data is called *regression*, or *curve estimation*. Given a new input X that has not yet been observed, we can then *predict* from the approximate regression function what the best estimate for the output Y should be.

Related to point estimation is the computation of *confidence intervals*. Given a statistical functional $\theta \in \mathbf{R}$, the problem is to compute an interval C from observed data $X_1, \dots, X_N$ such that for each μ,

$$\mathbf{P}_\mu(\theta \in C) \geqslant 1 - \alpha$$

The interval C is then called an *α-confidence interval*. More generally, if θ is a vector we can consider a *confidence set*. It is standard, but not necessary, to take $\alpha = 0.05$, and unless specified we will take this as the choice of α.

Remark. Under the frequentist interpretation of probability, a confidence interval *does not* give a probability associated with the quantity θ, since we assign no probability distribution to the distributions μ we consider. Instead, one can interpret a confidence interval as follows: each time we perform a confidence interval computation, we should be expected to be right 95% of the time.

Another problem is *hypothesis testing*. Here we have a particular fact about a distribution, called the *null hypothesis*. Given some data, the goal of hypothesis testing is to determine whether we should reject the null hypothesis, or if we have grounds to still believe in the null hypothesis.

Chapter 2

Point Estimation

Let us consider point estimation. We consider a statistical functional θ of some distribution μ, from which we observe data $X_1, \dots, X_N$. We must now find a statistic $\widehat{\theta}_N$, called the *estimator*, which is our best guess of θ. We say the estimator $\widehat{\theta}$ is *unbiased* if $\mathbf{E}_\mu[\widehat{\theta} - \theta_N] = 0$ for any distribution μ in our statistical model. It is consistant if $\widehat{\theta}_N$ converges to θ in probaibility as $N \to \infty$, for any μ.

As a function of random data, the estimator $\widehat{\theta}$ is a random variable. It's distribution is called the *sampling distribution*. The standard deviation of the estimator is called the *standard error* SE_N. Often, this error will depend on μ, and will thus be unknown. But we can often find an *estimator* for the standard error, which will be denoted by $\widehat{\text{SE}}_N$.

To determine how good an estimator is, we produce a *loss function* L, and then to determine the *expected loss*

$$\mathbf{E}_\mu[L(\theta, \widehat{\theta}_N)].$$

The function L is normally selected so that if $\widehat{\theta}_N = \theta$ then no loss in incurred, and more loss is incurred the 'further away' $\widehat{\theta}_N$ is from being correct. There are several natural choices for the loss function:

- The most common is the loss function

  $$L(\theta_1, \theta_2) = (\theta_1 - \theta_2)^2.$$

  The resulting expected loss is the *mean square error*, denoted MSE. A nice feature of this formula is the *bias variance* decomposition, i.e.

for any estimator, if we define

$$\text{BIAS} = \mathbf{E}[\widehat{\theta}_N - \Theta_N],$$

then

$$\text{MSE} = \text{BIAS}^2 + \text{SE}^2.$$

- Another common choice is the L^1 loss function

$$L(\theta_1, \theta_2) = |\theta_1 - \theta_2|.$$

- For discrete problems, another choice is the $0 - 1$ loss function

$$L(\theta_1, \theta_2) = \mathbf{I}(\theta_1 \neq \theta_2).$$

Given a loss function, our goal is often therefore to find an estimator which minimizes the expected loss over all other possible estimators.

Expanding out the mean square error, we conclude that

$$\text{MSE} = \text{BIAS}^2 + \text{SE}^2,$$

where $\text{BIAS} = \mathbf{E}_\mu[\widehat{\theta}_N] - \theta$. In particular, we conclude that if $\text{BIAS} \to 0$ and $\text{SE} \to 0$ as $N \to \infty$, then $\widehat{\theta}_N$ is a consistent estimator, since then $\widehat{\theta}_N$ converges in the L^2 norm to θ for any distribution μ, and convergence in L^2 implies convergence in probability.

Example. *Consider $X_1, \dots, X_N \sim$ Bernoulli(p), where $p \in [0,1]$ is unknown. A natural estimator to use is*

$$\widehat{p}_N = \frac{X_1 + \dots + X_N}{N}.$$

It is simple to check via linearity of expectation that $\widehat{p}_N$ is unbiased. It's standard error is

$$SE = \sqrt{\frac{p(1-p)}{N}}.$$

A natural estimator for this standard error is therefore

$$\widehat{SE}_N = \sqrt{\frac{\widehat{p}_N(1-\widehat{p}_N)}{N}}.$$

For any value p, the standard error converges to zero, so this is a consistent estimator.

A common method for point estimation, especially in parametric problems where the point we are estimating determines the distribution under study, is the *maximum likelihood estimator*. We begin by assuming the statistical model we are working in is contained in the space of all distributions which are either continuous or are discrete. Given data $X_1, \dots, X_N$, we define the *likelihood function*

$$L(\mu) = f(X_1) \cdots f(X_N)$$

where f is the density function of μ. To find the maximum likelihood estimator, we find

$$\mu_* = \arg\max_{\mu \in \mathcal{F}} (L(\mu)),$$

the 'most likely' distribution to generate a given dataset. We then set $\widehat{\theta}_N = \theta^*$, where θ^* is the value of the statistical function θ we are estimating which is associated with the distribution μ^*. To compute μ^*, it is often easier to work with the *log likelihood*

$$l(\mu) = \log f(X_1) + \log f(X_2) + \dots + \log f(X_N),$$

since the distribution maximizing likelihood also maximizes the log-likelihood.

Example. *Consider $X_1, \dots, X_N \sim$ Bernoulli(p), where p is unknown, and suppose we want to estimate p. We will find that the estimator we saw above is the maximum likelihood estimator for this example. Indeed, we have*

$$\begin{aligned} l(p) &= \sum_{i=1}^N \log \mathbf{P}_p(X_i) \\ &= \sum_{i=1}^N \log \left(p^{X_i} (1-p)^{1-X_i} \right) \\ &= \sum_{i=1}^N X_i \log(p) + (1 - X_i) \log(1-p). \end{aligned}$$

The derivative of l as a function of p is equal to

$$\sum_{i=1}^N \frac{X_i}{p} - \frac{1 - X_i}{1-p} = \frac{1}{1-p} \left(\frac{S}{p} - N \right),$$

where $S = \sum X_i$. Setting this quantity equal to zero gives that we should set $\widehat{p}_N = S/N$, which was the estimator we considered before.

Example. *Consider $X_1, \ldots, X_N \sim N(\mu, \sigma^2)$, where μ and σ are unknown. The problem is to estimate $\theta = (\mu, \sigma)$. The likelihood function is*

$$L(\theta) = \frac{1}{\sigma^N (2\pi)^{N/2}} \exp\left(-\frac{1}{2} \sum_{i=1}^{N} \left(\frac{X_i - \mu}{\sigma} \right)^2 \right),$$

and thus the log likelihood is proportional to

$$l(\theta) = -N \log(\sigma) - \frac{1}{2} \sum_{i=1}^{N} \left(\frac{X_i - \mu}{\sigma} \right)^2.$$

Taking partial derivatives in σ and μ gives that the maximum likelihood parameters μ^* and σ^* satisfy

$$\mu^* = \frac{X_1 + \cdots + X_N}{N}$$

and

$$\sigma^* = \left(\frac{(X_1 - \mu)^2 + \cdots + (X_N - \mu)^2}{N} \right)^{1/2},$$

which are the natural estimators we might choose given the observed data.

Example. *Consider $X_1, \ldots, X_N \sim$ Uniform$([0, \theta])$, for some $\theta \in [0, \infty)$. What is the maximum likelihood estimator of θ? The likelihood function is*

$$L(\theta) = \theta^{-N} \mathbf{I}(\max(X_1, \ldots, X_N) \leqslant \theta).$$

This function is equal to zero for $\theta \in [0, \max(X_1, \ldots, X_N))$, and decreases for values bigger than $\max(X_1, \ldots, X_N)$. Thus we conclude that the maximum likelihood estimator for θ is

$$\widehat{\theta}_N = \max(X_1, \ldots, X_N).$$

Example. *Consider a practical example. Let us suppose that there are N fish in a lake, a quantity unknown to us, and which we would like to estimate. We capture A fish, paint them with red spots, and then release them back into the wild. We then capture B fish, and observe that X of them are red. Let us use this information to compute the maximum likelihood estimate of N. The only random quantity here is X, the other quantities A and B determining the* *statistical model we use. Given N, the distribution of X is a hypergeometric distribution, and so the likelihood function is*

$$L(N) = \frac{\binom{A}{X}\binom{N-A}{n-X}}{\binom{N}{B}},$$

To determine the maximum quantity here, we compute there for $N > A$,

$$\frac{L(N)}{L(N-1)} = \frac{N-A}{N-A-B+X}\frac{N-B}{N}.$$

This quantity is greater than one if $N < AB/X$, and smaller than one if $N > AB/X$. The maximum likelihood estimator for N is therefore the greatest integer less than or equal to AB/X, i.e.

$$\widehat{N} = \left\lfloor \frac{AB}{X} \right\rfloor.$$

As an example, if we capture 1000 fish and mark them with red spots, capture another set of 1000 fish, and observe that 100 have red spots, we should guess that there are 10000 fish in the lake.

Chapter 3

Regression

The most basic task of statistics is regression. Given a sequence of independent experiments $(X_1, Y_1), \dots, (X_N, Y_N)$ drawn from some distribution (X, Y), we want to determine the a function f such that $f(X)$ approximates Y. To measure the approximations success, for each value (x, y), we consider a *loss function* $L(y, f(x)) \geqslant 0$. The *estimated prediction error* is then defined to be $\text{EPE}(f) = \mathbf{E}(L(Y, f(X)))$. The goal of *regression* is to find a function f which minimizes the estimated prediction error. In parametric problems, the goal is to find the function f lying in a given finite dimensional class of candidate functions minimized estimated prediction error. In non parametric problems, we must find the function f from an infinite dimensional class of candidate functions.

Example. *In most cases, the most analytically convenient loss functions is the* square error loss

$$L(y, f(x)) = (y - f(x))^2.$$

We then have

$$EPE(f) = \mathbf{E}(L(Y, f(X))) = \mathbf{E}((Y - f(X))^2) = \mathbf{E}(\mathbf{E}(Y - f(X))^2 | X)).$$

In this case, the expected prediction error is known as the mean square error, *denoted $MSE(f)$. Thus regression reduces to minizing $(Y - f(X))^2$ pointwise, given X. The minimizer of this quantity is the conditional expectation*

$$f(X) = \mathbf{E}(Y|X).$$

This is because if f is any function, then $\mathbf{E}(f(X)|X) = f(X)$, and so

$$\mathbf{E}((Y - \mathbf{E}(Y|X))f(X)|X) = \mathbf{E}(Y|X)f(X) - \mathbf{E}(Y|X)f(X) = 0.$$

Thus $Y - \mathbf{E}(Y|X)$ is orthogonal to the subspace of random variables measurable with respect to the sigma algebra generated by X. Thus we can apply the Pythagorean theorem to conclude that

$$\begin{aligned} MSE(f) = \mathbf{E}\left((Y - f(X))^2|\right) &= \mathbf{E}\left((Y - \mathbf{E}(Y|X))^2\right) + \mathbf{E}\left((\mathbf{E}(Y|X) - f(X))^2\right) \\ &\geqslant \mathbf{E}\left((Y - \mathbf{E}(Y|X))^2\right) = MSE(\mathbf{E}(Y|X)). \end{aligned}$$

Thus regression with respect to squared loss is equivalent to estimating the conditional expectation of one variable with respect to one another. The decomposition above using Pythagoras' theorem is very useful, which is really what makes the square error loss most useful in a given situation.

Example. *Another standard loss function is the L^1 loss function, given by*

$$L(y, f(x)) = |y - f(x)|.$$

As with the squared loss it suffices to choose $f(X)$ which pointwise minimizes

$$\mathbf{E}\left(|Y - f(X)||X\right).$$

Fix y. We note that if $\varepsilon > 0$, then

$$|Y - y + \varepsilon| - |Y - y| = \begin{cases} \varepsilon & : Y \geqslant y \\ -\varepsilon & : Y \leqslant y - \varepsilon \\ 2(Y - y) + \varepsilon & : y - \varepsilon < Y < y \end{cases}$$

Thus

$$\begin{aligned} &|\mathbf{E}\left(|Y - y + \varepsilon| - |Y - y||X\right) - \varepsilon[\mathbf{P}(Y \geqslant y|X) - \mathbf{P}(Y \leqslant y - \varepsilon|X)]| \\ &\quad \leqslant \varepsilon\,\mathbf{P}(y - \varepsilon < Y < y|X). \end{aligned}$$

Provided we are working with a regular probability measure, this means that for each ω, the function $y \mapsto \mathbf{E}(|Y - y||X = X(\omega))$ is right differentiable, with derivative $\mathbf{P}(Y \geqslant y|X = X(\omega)) - \mathbf{P}(Y < y|X = X(\omega))$. In particular, a choice of y which minimizes $\mathbf{E}(|Y - y||X = X(\omega))$ must satisfy

$$\mathbf{P}(Y \geqslant y|X = X(\omega)) = \mathbf{P}(Y < y|X = X(\omega)) = 0.5.$$

If $f(X)$ is a function such that almost surely,

$$\mathbf{P}(Y \geqslant f(X)|X) = \mathbf{P}(Y < f(X)|X) = 0.5$$

then we say it is a conditional median, *and it will be a pointwise minimizer of the expected loss. We normally denote a conditional median by* $\mathbf{M}(Y|X)$. *Unlike the conditional expectation, the conditional median need not be unique if the underlying distribution of X is not continuous.*

Example. *Suppose that the values of Y lie in some discrete set of values. The problem of regression in this setting is normally called* classification. *A natural loss function to use here is the 0-1 loss function*

$$L(y, f(x)) = \mathbf{I}(y \neq f(x)).$$

As with the previous examples, the regression function in this setting can be easily proved to be

$$\mathbf{B}(Y|X) = \arg\min \mathbf{P}(Y = y|X).$$

This is known as the Bayes classifier. *The value EPE*$(\mathbf{B}(Y|X))$ *is known as the* Bayes rate.

Even in these examples, where we can calculate an explicit formula for the regression function, in practice we cannot compute the regression function from sample data. Thus we must come up with an approximation $\widehat{f}$ of the regression function $f(x)$, where for each x, $\widehat{f}(x)$ is a random variable determined by the data $(X_1, Y_1), \dots, (X_N, Y_N)$. Such a random variable is known as a *statistic*.

There is a very useful decomposition result for the expected prediction error of $\widehat{f}$, where $f(x) = \mathbf{E}(Y|X = x)$. We can write $Y = f(X) + \varepsilon$, where ε has mean zero and variance σ^2. If we define the *bias*

$$\text{Bias}\,(x) = \mathbf{E}\left(\widehat{f}(x)\right) - f(x).$$

then

$$\begin{aligned} \text{MSE}(f) &= \mathbf{E}[(Y - f(X))^2] + \mathbf{E}[(f(X) - \widehat{f}(X)]^2) \\ &= \sigma^2 + \mathbf{V}[\widehat{f}] + \mathbf{E}\left[\text{Bias}(X)^2\right]. \end{aligned}$$

This is referred to as the *Bias-Variance decomposition*. The error σ^2 is unavoidable for any function $\widehat{f}$, whereas to minimize the mean square error, we must make a tradeoff between making the variance of our estimator small, and the bias small.

3.1 Parametric Regression

In parametric statistics, we are given independent and identically distributed data $\{(X_1, Y_1), \dots, (X_N, Y_N)\}$. We do not know the distribution they are drawn from, but we *do* know the distribution lies in some

3.2 Linear Regression

In some case, we assume our regression functions take the form $\beta^* X$, where $\beta^* \in (\mathbf{R}^n)^*$. Given the data $(X_1, Y_1), \dots, (X_k, Y_k)$, we determine the best estimate $\widehat{\beta}$ of β^* by evaluating it against the loss function $\mathcal{L}(\beta) = \mathbf{E}[(Y - \beta X)^2]$. Of course, we cannot calculate $\mathcal{L}$ directly, but we may estimate it with our samples. Because the loss function is a differentiable function of β, we may take derivatives to determine β:

$$\nabla \mathcal{L}(\beta) = 2\mathbf{E}[(Y - \beta X)X^T] = 2\mathbf{E}(YX^T) - 2\beta \mathbf{E}(XX^T)$$

This is optimized when the derivative of this function is zero. i.e., when

$$\beta \mathbf{E}(XX^T) = \mathbf{E}(YX^T)$$

Assuming $\mathbf{E}(XX^T)$ is invertible, we may invert, and determine that the optimal value β^* can be calculated as

$$\beta^* = \mathbf{E}(YX^T)\mathbf{E}(XX^T)^{-1}$$

Now if we only have the samples (X_i, Y_i), we may approximate this value by forming the conglomerate matrices $\mathbf{X} = (X_1 | X_2 | \dots | X_n)$ and $\mathbf{Y} = (Y_1, \dots, Y_n)$, and calculating $\widehat{\beta} = \mathbf{Y}\mathbf{X}^T(\mathbf{X}\mathbf{X}^T)^{-1}$. This minimizes the error over the training data set $\sum (Y_i - \beta X_i)^2 = \|\mathbf{y} - \beta \mathbf{X}\|^2$.

How do we estimate how accurate our prediction is. First, assume that each Y_i is independant, with the same variance σ^2. Then

$$\mathbf{V}(\widehat{\beta}) = \mathbf{V}(\mathbf{Y}\mathbf{X}^T(\mathbf{X}\mathbf{X}^T)^{-1})$$

INSERT THEORETICAL ESTIMATES, Gauss Markov theorem, etc.

3.3 Additive Models

A generalized additive model has a regression function of the form

$$\mathbf{E}(Y|X) = \alpha + f_1(X^1) + \cdots + f_n(X^n)$$

where the f_i are unspecified smooth (C^∞) functions, and $X = (X^1, \ldots, X^n)$ is a random vector. To fit an additive model, given a sample $(X_1, Y_1), \ldots, (X_m, Y_m)$, we take as a cost function the penalized sum of squares to find the constant α and functions f_i,

$$\sum_{i=1}^{m} \left(Y_i - \alpha - \sum_{j=1}^{n} f_j(X_i^j) \right)^2 + \sum_{j=1}^{m} \lambda_j \int (f_j'')^2$$

Where the $\lambda_j \geqslant 0$ are arbitrary parameters. The minimizer of this cost function is not unique – it is standard convention to require that $\sum_{i,j} f_i(X_i^j) = 0$. One can apply an iterative cubic smoothing spline solution to find this minimum.

3.4 Tree-Based Models

Tree based methods partition the feature space , and then fit a simple model (normally a constant) into each one. If $\mathcal{S}$ is such a space, and we partition it into $S_1, \ldots, S_n$, each with a model $f_1, \ldots, f_n$, then are model is

$$\mathbf{E}(Y|X) = \sum_{i=1}^{n} f_i(X)[X \in S_i]$$

If S_i has already been decided, and we are using constants for the f_i, then the best choice of constants (according to the least squares cost function) given a sample $(X_1, Y_1), \ldots, (X_n, Y_n)$ are just the mean values of the Y_j with $X_i \in S_i$. The questions remains, however, on how to choose our partitions.

To find optimal partitions, we assume our feature space is $\mathbf{R}^n$, and our regions formed by 'binary splits'. We start with the whole space, pick a 'splitting coordinate' i and 'splitting point' $t \in \mathbf{R}$, and partition our region into two sets $A = \{x \in \mathbf{R}^n : x_i < t\}$ and $B = \{x \in \mathbf{R}^n : x_i \geqslant t\}$. We then recursively partition A and B up in this manner, until we are satisfied with our splits.

Finding the best choice of partition using the method above is generally computationally infeasible. We shall proceed with a greedy approximation. Given a region A containing features $X_1, \ldots, X_n$, we seek to find a splitting variable i and split point t which minimize the cost function

$$\operatorname*{arg\,min}_{i,t} \min_{a} \sum_{X_j^i \leqslant t} (Y_j - a)^2 + \min_{b} \sum_{X_j^i > t} (Y_j - b)^2$$

Given i and t, the minimum values of a and b are just obtained by taking the mean of the results Y_j. By doing a linear scan on each coordinate, it is fairly simple to find i and t. Then we recursively perform this greedy approach on the subpartitions.

Now when do we stop splitting? If we split far enough, then we will only have very few examples in each subregion, and we will have overfitted our training data! Furthermore, it will be very difficult to interpret the model we have created.

Chapter 4

Neural Networks

Neural Networks arise from the solution of a certain model, known as the Projection Pursuit Regression model. Assume we have an input vector $X \in \mathbf{R}^n$, with target Y. The projection pursuit regression model has the form

$$f(X) = \sum_{i=1}^{M} g_i(\beta_i X)$$

Where the g_i are unspecified, and $\beta_i \in (\mathbf{R}^n)^*$. This is an additive model, but in the features $V_i = \beta_i X$. Each $g_i(\beta_i X)$ is called a ridge function in $\mathbf{R}^n$.

This model is really general. For instance, the product of the coordinates can be written

$$X_1 X_2 = \frac{(X_1 + X_2)^2 - (X_1 - X_2)^2}{4}$$

In fact, if we let M be large enough, for appropriate choices of g_i can approximate arbitrary continuous functions on $\mathbf{R}^n$ (this model is a universal approximator). Unfortunately, this means this model will be hard to fit exactly, and thus the model is better for estimating data rather than obtaining an understandable model.

Given some data $(X_1, Y_1), \dots, (X_n, Y_n)$ from the regression model, we thus seek the minimize the error

$$\sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{M} g_j(\beta_j x_i) \right)$$

as a choice of g_j and β_j. We need to impose constraints on g_j to prevent overfitting.

Suppose we have $M = 1$, and that $g = g_1$ is differentiable, and $\beta = \beta_i$ is the linear functional. Then, taking the initial terms around the taylor series,

$$g(\beta x_i) \approx g(\alpha x_i) + g'(\alpha x_i)(\alpha - \beta)x_i$$

$$\begin{aligned}\sum_{i=1}^{n}[y_i - g(\beta x_i)]^2 &\approx \sum_{i=1}^{n}[y_i - g(\alpha x_i) - g'(\alpha x_i)(\alpha - \beta)x_i]^2 \\ &= \sum_{i=1}^{n} g'(\alpha x_i)^2 \left[\alpha x_i + \frac{y_i - g(\alpha x_i)}{g'(\alpha x_i)} - \beta x_i\right]^2\end{aligned}$$

We can minimize the right-hand side by carrying out a least squares regression with target

$$\alpha x_i + \frac{y_i - g(\alpha x_i)}{g'(\alpha x_i)}$$

We can then iterate this regression until convergence. With more than one term in the model, we just perform forward stage-wise reggresion.

The projection pursuit regression model has not been widely used in the field of statistics, possibly due to the lack of computational resources when it was created. Nonetheless, it leads to the field of neural networks, which are much more useful.

4.1 Nets

There is a lot of mysticism surrounding neural networks (perhaps for the same reason 'the god particle' is so controversial) but they are really just non-linear statistical models. Here we will discuss the most basic kind of neural nets, the single hidden layer back-propogation network.

Suppose we are given a set of inputs $X = (X_1, \dots, X_n)$. A neural net creates layers of derived features $Z = (Z_1, \dots, Z_m)$ as affine transformations of the X_i, 'flattened' by some activation function σ. In the single layer approach, we have one layer of these derived features, and then these derived features are used to generate the target $Y = (Y_1, \dots, Y_k)$ as a function of the Z_i, again modified by an output function. In terms of formulas, our mathematical model is

$$Z = \sigma(\Lambda X + v) \qquad W = \Delta Z + w \qquad f(X) = g(W)$$

where Λ and Δ are linear transformations, and our regression function is f.

For regression, we normally choose not to modify our outputs via an output function (that is, we let $g = \mathbf{1}$). For classification, we need to choose an output function which results in reasonable results. The sigmoid function is often chosen as the activation, $\sigma(t) = (1 + e^{-t})^{-1}$. Sometime Gaussian radial basis functions are used, producing a radial basis function network. Note that if we let σ and the output regularization function be the identity, we obtain a linear model. Thus in this way, a neural network is a generalization of the linear model.

Chapter 5

Bayesian Networks

Let X,Y,Z be random variables. We say that X and Y are *conditionally independent* given Z, if for any measurable $A,B \subset \mathbf{R}$,

$$\mathbf{P}(X \in A, Y \in B | Y, Z) = \mathbf{P}(X \in A|Z)\mathbf{P}(Y \in B|Z)$$

This just means that once you know Z, you can gain no information about X from information about Y. Bayesian networks are a model of information which allow us to measure the conditional independence of random variables.

Given a set $X_1,\dots,X_n$ of random variables, suppose we form a directed, acyclic graph whose certices are the random variables. We say the variables are *Markov* with respect to the graph if for any random variable X_i,

$$\mathbf{P}(X_1 \in A_1,\dots,X_n \in A_n) = \prod_{i=1}^{n} \mathbf{P}(X_i \in A_i|\text{parents}(X_i))$$

In other words, this means exactly that X_i is independent of all variables once we condition on the parents of X_i.

Part I

Computational Learning Theory

Computational Learning Theory is the study of Machine Learning, from the perspective of theoretical computing science. In the study of computational learning theory, we specify learning models, which limit the *complexity* of the kinds of algorithms one can use to study data in machine learning. Since we can often obtain huge inputs in statistics, we often limit our study to data with low *sample complexity*, i.e. the minimum number of samples required to solve an algorithm, up to a certain degree of error. We might also naturally limit algorithms in *time complexity*, since, despite a problem having low sample complexity, there might not exist an efficient algorithm to analyze a given set of data. We might also limit the *hypothesis complexity*.