1. Background and Overview

Data mining tries to find hidden structure in large, high-dimensional datasets.

Interesting structure can arise in regression analysis, discriminant analysis, cluster analysis, or more exotic situations, such as multidimensional scaling.

Classic applications include:

- Regression models for climate change, wine price, cost of software development.
- Classification models for fraudulent credit card transactions and good credit risk.
- Cluster analyses for market segmentation and microarray data.
- Multidimensional scaling analyses for document retrieval systems.

Data mining grew at the interface of statistics, computer science, and database management.

- Statistical work began in the 1980s, with the invention of CART, and expanded through increased research on visualization, nonparametric regression, and data quality.
- Computer scientists coined the term data mining, pioneered neural nets, and developed a body of ad hoc techniques.
- Database scientists developed SQL, relational databases, and other key tools.

Data mining has become an important research area, and was the topic of a year-long program at the Statistical and Applied Mathematical Sciences Institute, 2003-2004. That program was the impetus to the development of this shortcourse.

1.1 Example: Nonparametric Regression

Nonparametric regression is a natural way to introduce the main ideas in data mining. The core model is

$$y = f(\boldsymbol{x}) + \epsilon$$

where $\boldsymbol{x} \in \mathbb{R}^p$ and f is unknown. The emphasis is upon estimating the function f.

In real problems one observes $\{y_i, \boldsymbol{x}_i\}$, $i = 1, \ldots, n$. We assume that:

- the $\boldsymbol{x}_i$ are measured without error
- the ϵ_i are i.i.d. with mean zero
- the variance of the ϵ_i values is an unknown constant σ.

These are the minor assumptions, and can be weakened in customary ways.

The main assumption regards the class of functions to which f belongs. Common assumptions include:

- f is in a Sobolev space (essentially, these are functions with bounded derivatives)
- f has a bounded number of discontinuities.

For now, we require only that f be vaguely smooth.

The problem of estimating f becomes vastly harder as p, the dimension of $\boldsymbol{x}$, increases. This is called the Curse of Dimensionality (COD). The term was coined by Richard Bellman in the context of approximation theory (*Adaptive Control Processes*, 1961, Princeton University Press).

In order to minimize or evade the COD, data miners have invented many computer-intensive techniques. Some of these include: MARS, CART, Projection Pursuit Regression, Loess, random forests, support vector machines.

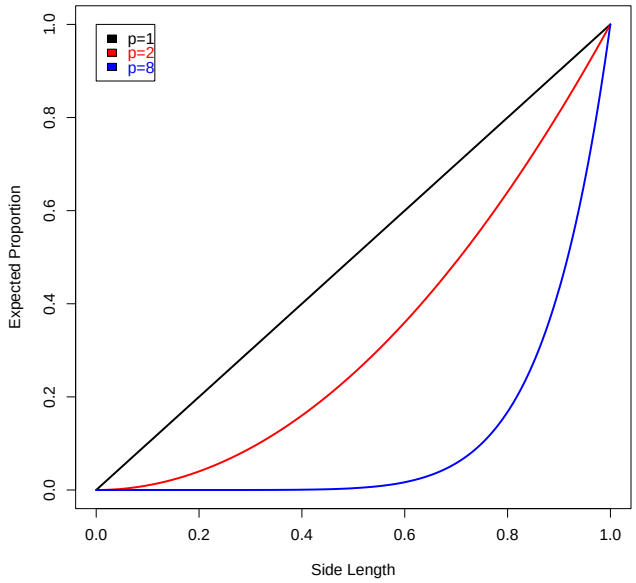
The COD applies to all multivariate analyses that choose not to impose strong modeling assumptions (e.g., that the relationship between $\boldsymbol{x}$ and $\mathbb{E}[Y]$ is linear, or that f belongs to a particular parametric family of curves).

Although we use regression as our example, the COD applies equally to classification, cluster analysis, and multidimensional scaling.

In terms of the sample size n and dimension p, the COD has three nearly equivalent descriptions:

- For fixed n, as p increases, the data become sparse.
- As p increases, the number of possible models explodes.
- For large p, most datasets are multicollinear (or concurve, which is a nonparametric generalization).

To explain the sparsity description of the COD, assume that n points are uniformly distributed in the unit cube in $\mathbb{R}^p$. What is the side-length ℓ of a subcube that is expected to contain a fraction d of the data? Ans: $\ell = \sqrt[p]{d}$



This means that for large p, the amount of local information that is available to fit bumps and wiggles in f is too small.

To explain the model explosion description of the COD, suppose we restrict attention to just linear models of degree 2 or fewer. For $p = 1$ these are:

$$\mathbb{E}[Y] = \beta_0 \qquad \mathbb{E}[Y] = \beta_1 x_1 \qquad \mathbb{E}[Y] = \beta_2 x_1^2$$
$$\mathbb{E}[Y] = \beta_0 + \beta_1 x_1 \qquad \mathbb{E}[Y] = \beta_0 + \beta_2 x_1^2 \qquad \mathbb{E}[Y] = \beta_1 x_1 + \beta_2 x_1^2$$
$$\mathbb{E}[Y] = \beta_0 + \beta_1 x_1 + \beta_2 x_1^2$$

For $p = 2$ this set is extended to include expressions with the terms $\alpha_1 x_2$, $\alpha_2 x_2^2$, and $\gamma_{12} x_1 x_2$.

For general p, combinatorics shows that the number of possible models is

$$2^{1+2p+\binom{p}{2}} - 1.$$

This increases superexponentially in p, and there is not enough sample to enable the data to discriminate among these models.

To explain the multicollinearity description of the COD, recall that multicollinearity occurs when the explanatory values concentrate on an affine subspace in $\mathbb{R}^p$. And multicollinearity implies that the predictive value of the fitted model breaks down quickly as one moves away from the subspace in which the data concentrate.

For large p, the number of possible subspaces is enormous $(2^p - 2)$, and so the probability that a sample of fixed size n concentrates on an affine shift of one of them, just by chance, is large.

Concurvity is the nonparametric analogue of multicollinearity, and it occurs when the data concentrate on some smooth manifold within $\mathbb{R}^p$. Since the number of smooth manifolds is larger than the number of affine shifts, the nonparametric version of the problem is worse.

Recently, several researchers have obtained results that purport to evade the COD.

- Barron (1994; *Machine Learning*, **14**, 115-133) shows that in a technical sense which we describe in 3.4, neural networks avoid the COD.
- Zhao and Atkeson (1991; *NIPS'91*, **4**, 936-943) show that in a sense similar to Barron's, Projection Pursuit Regression can evade the COD.
- Wozniakowski (1991; *Bulletin of the American Mathematical Society*, N.S., **24**, 184-194) uses a modification of Hammersley points (chosen to minimize the discrepancy from the uniform distribution in a Kolmogorov-Smirnov test) to dodge the COD in the context of multivariate integration.

None of these results has much practical significance to data miners. The results are *very* asymptotic, and rely upon some awkward fine print.

1.2 Two Tools

In regression, classification, and (often) clustering, there are two key tasks:

- Assessment of model fit.
- Estimation of uncertainty.

The first problem is handled by some variant of cross-validation. The second problem is handled by the bootstrap.

Both cross-validation and bootstrapping are key methodologies in data mining, and we review them briefly.

1.2.1 Cross-Validation

To assess model fit in complex, computer-intensive situations, the ideal strategy is to hold out a random portion of the data, fit a model to the rest, then use the fitted model to predict the response values from the values of the explanatory variables in the hold-out sample.

This allows a straightforward estimate of predicted mean squared error (PMSE) for regression, or predictive classification error, or some similar fit criterion. But this wastes data.

Also, we usually need to compare fits among *many* models. If the same hold-out sample is re-used, then the comparisons are not independent and (worse) the model selection process will tend to choose a model the overfits the hold-out sample, causing spurious optimism.

Cross-validation is a procedure that balances the need to use data to select a model and the need to use data to assess prediction.

Specifically, v-fold cross-validation is as follows:

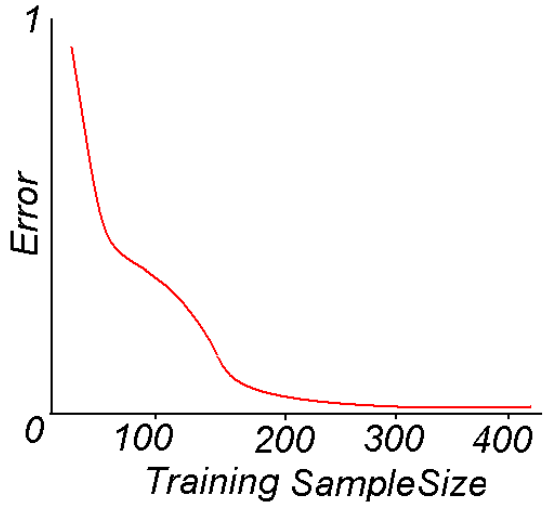
- randomly divide the sample into v portions;
- for $i = 1, \ldots, v$, hold out portion i and fit the model from the rest of the data;
- for $i = 1, \ldots, v$, use the fitted model to predict the hold-out sample;
- average the PMSE over the v different fits.

One repeats these steps (including the random division of the sample!) each time a new model is assessed.

The choice of v requires judgment. If $v = n$, then cross-validation has low bias but possibly high variance, and computation is lengthy. If v is small, say 4, then bias can be large. A common choice is 10-fold cross-validation.

Leave-one-out cross-validation takes $v = n$, and calculates the predicted value for each observation using all of the other data. It is almost unbiased for the true predictive error. But the variance can be large, because the samples are so similar, and it is lengthy to calculate since it requires n runs.

It is tricky to now what value of v to use. If one wants to minimize mean squared error, then one must balance the variance in the estimate against the bias. One strategy is to plot the error as a function of the size of the training sample—when the curve levels off, there is no need to increase v.



Cross-validation is not perfect—some dependency remains, and the process can absorb a lot of computer time. Many of the data mining techniques use computational shortcuts to approximate cross-validation.

For example, in a regression model where the estimated values have linear dependence on the observed values, or $\hat{\boldsymbol{y}} = \boldsymbol{H}\boldsymbol{y}$, then often

$$n^{-1}\sum_{i=1}^{n}[y_i - \hat{f}^{-1}(\boldsymbol{x}_i)]^2 = n^{-1}\sum_{i=1}^{n}\left[\frac{y_i - \hat{f}(\boldsymbol{x}_i)}{1 - h_{ii}}\right]^2$$

where $\hat{f}^{-1}(\boldsymbol{x}_i)$ is the leave-one-out cross-validation estimate of f at $\boldsymbol{x}_i$. This reduces computational time by requiring only one calculation of $\hat{f}$.

To avoid calculating h_{ii}, Generalized Cross-Validation (GCV) estimates the total squared error as

$$n^{-1}\sum_{i=1}^{n}\left[\frac{y_i - \hat{f}(\boldsymbol{x}_i)}{1 - h_{ii}}\right]^2 = n^{-1}\sum_{i=1}^{n}\left[\frac{y_i - \hat{f}(\boldsymbol{x}_i)}{1 - \mathbf{tr}(\boldsymbol{H})/n}\right]^2 .$$

1.2.2 The Bootstrap

The bootstrap is a popular tool for setting approximate confidence regions on estimated quantities when the underlying distribution is unknown. It relies upon samples drawn from the empirical cumulative distribution function (ecdf).

Let $X_1, \ldots, X_n$ be iid with cdf F. Then

$$\hat{F}_n(x) = \frac{1}{n} \sum_{i=1}^{n} I_{(-\infty, x]}(X_i)$$

is the ecdf. The ecdf is bounded between 0 and 1 with jumps of size n^{-1} at each observation.

The ecdf is a nonparametric estimator of the true cdf. It is the basis for Kolmogorov's goodness-of-fit test, and plays a key role in many aspects of statistical theory.

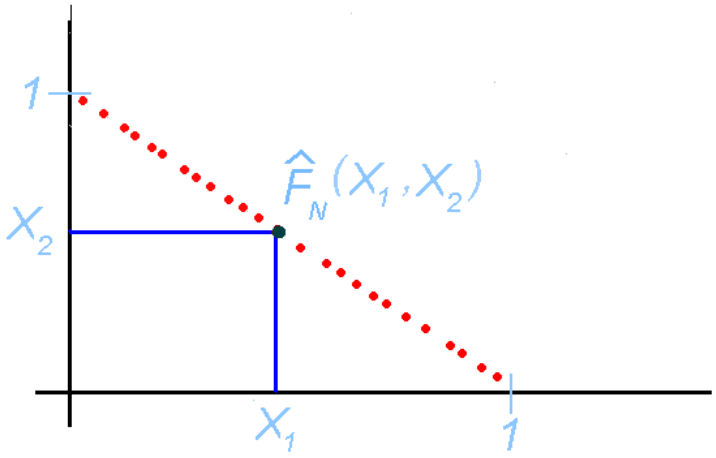
The Glivenko-Cantelli theorem implies

$$\mathbb{P}[\limsup_n |\hat{F}_n(x) - F(x)| < \epsilon] = 1 \textbf{ a.s.}$$

This fails in higher dimensions, but convergence in distribution holds, i.e., for each continuity point $\boldsymbol{x}$ in $\mathbb{R}^p$,

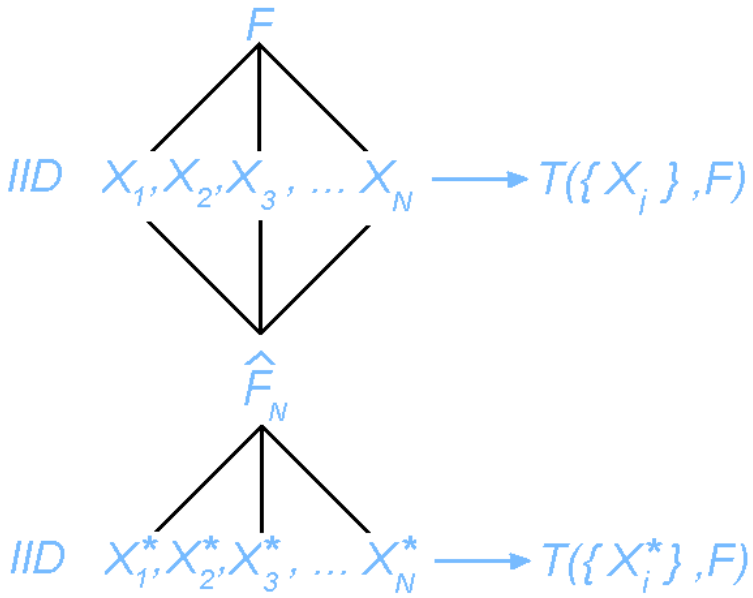
$$\lim_n \hat{F}_n(\boldsymbol{x}) = F(\boldsymbol{x}).$$

This is sufficient for bootstrap purposes.



Denote the estimate of a functional of F using the sample $\{\boldsymbol{X}_i\}$ by $T(\{\boldsymbol{X}_i\}, F)$. Here $T(\{\boldsymbol{X}_i\}, F)$ might be the population variance, or the ratio of the 7th moment to the 12th moment, or any other complex function for which one wants a confidence region.

To set a confidence region one needs to know how the sampling variation affects the estimator. The strategy behind the bootstrap reflects the reflexivity in its name. See Efron (1979; *Annals of Statistics*, **7**, 1-26)



Since $\hat{F}_n \to F$, the distribution of $T(\{X_i^*\}, \hat{F}_n)$ converges to the distribution of $T(\{X_i\}, F)$, the quantity of interest.

Suppose $X_1, \ldots, X_n$ are iid f and we want to find the distribution of

$$T(\{X_i\}, F) = \sqrt{n}\frac{\bar{X} - \mu}{s}$$

or, equivalently,

$$\mathbb{P}_F[\sqrt{n}\frac{\bar{X} - \mu}{s} \leq t] \quad \forall\, t \in \mathbb{R}.$$

The bootstrap approximation to this is

$$\mathbb{P}_{\hat{F}_n}[\sqrt{n}\frac{\bar{X}^* - \bar{X}}{s^*} \leq t] \quad \forall\, t \in \mathbb{R}$$

where $\bar{X}^*$ is the average of a random sample from $\hat{F}_n$ and s^* is its standard deviation. This can be numerically evaluated by repeated resamplings from $\hat{F}_n$.

Before Efron invented the bootstrap, statisticians used the Central Limit Theorem (CLT) to approximate the distribution of $T(\{X_i\}, F)$ by a standard normal distribution. So how good is the bootstrap? Is it better than the CLT?

To answer this, use an Edgeworth expansion argument as in Hall (1992; *The Bootstrap and Edgeworth Expansion*, Springer). Under reasonable technical conditions,

$$\mathbb{P}_F[\sqrt{n}\frac{\bar{X}-\mu}{s} \leq t] = \Phi(t) + n^{-1/2}p_1(t)\phi(t) + \ldots + n^{-j/2}p_j(t)\phi(t) + o(n^{-j/2})$$

where $\Phi(t)$ is the cdf of the standard normal, $\phi(t)$ is its density function, and the $p_j(t)$ functions are polynomials related to the Hermite polynomials and involve the $j+2$nd and lower moments of F.

The "oh" notation $o(h(n))$ means that the error gets small faster than $h(n)$; i.e.,

$$\lim_{n\to\infty} \textbf{error}/h(n) = 0.$$

If this happens in probability, we denote it by $o_p(h(n))$.

Recall that a pivot is a function of the data (and usually the unknown parameters) whose distribution does not depend upon the unknown parameters. For example,

$$T(\{X_i\}, F) = \sqrt{n}\frac{\bar{X} - \mu}{s}$$

is a pivot in the class of normal distributions, since this has the student's-t distribution for any μ and σ^2.

And in the class of distributions for which the first two moments are finite, $T(\{X_i\}, F)$ is an asymptotic pivot, since its asymptotic distribution is the standard normal.

For functionals that are asymptotic pivots with standard normal distribution, the Edgeworth expansion implies

$$\begin{aligned} G(y) &= \mathbb{P}[T(\{X_i\}, F) \leq y] \\ &= \Phi(y) + n^{-1/2} p_1(y)\phi(y) + \mathcal{O}(n^{-1}) \end{aligned}$$

where $\mathcal{O}(n^{-1})$ means that the ratio of the absolute error to n^{-1} is bounded for all $n > M$.

The bootstrap estimate for $G(y)$ turns out to be

$$\begin{aligned} G^*(y) &= \mathbb{P}[T(\{X_i^*\}, \hat{F}_n) \leq y \,|\, \{X_i\}] \\ &= \Phi(y) + n^{-1/2}\hat{p}(y)\phi(y) + \mathcal{O}_p(n^{-1}) \end{aligned}$$

where

$$T(\{X_i^*\}, \hat{F}_n) = \sqrt{n}\frac{\bar{X}^* - \bar{X}}{s^*}$$

as before, and $\hat{p}(y)$ is obtained from $p(y)$ by replacing the $j+2$nd and lower moments of F by the corresponding moments of the ecdf.

The $\mathcal{O}_p(n^{-1})$ is a random variable that means the error term is $\mathcal{O}(n^{-1})$ in probability, or

$$\lim_{\lambda\to\infty} \limsup_{n\to\infty} \mathbb{P}[\frac{\textbf{error}}{n^{-1}} > \lambda] = 0.$$

One can show (in a course in asymptotics) that $\hat{p}(y) - p(y) = \mathcal{O}_p(n^{-1/2})$.

Thus

$$\begin{aligned} G^*(y) - G(y) &= n^{-1/2}\phi(y)[\hat{p}(y) - p(y)] + \mathcal{O}_p(n^{-1}) \\ &= \mathcal{O}_p(n^{-1}) \end{aligned}$$

since the first term of the sum is also $\mathcal{O}_p(n^{-1})$ and big-oh errors add.

So using a bootstrap approximation to an asymptotic pivot statistics incurs an error of order n^{-1}.

In contrast, recall that

$$\begin{aligned} G(y) - \Phi(y) &= n^{-1/2}p(y)\phi(y) + \mathcal{O}(n^-1) \\ &= \mathcal{O}(n^{-1/2}). \end{aligned}$$

So the CLT has error of order $n^{-1/2}$, and thus is asymptotically worse than the bootstrap.

Suppose we had bootstrapped a function that was not a pivot. For example, the percentile bootstrap (cf. Efron, 1982; *The Jackknife, the Bootstrap, and Other Resampling Plans*, SIAM, Philadelphia) would use the distribution of

$$U^* = \sqrt{n}(\bar{X}^* - \bar{X})$$

as a proxy when making uncertainty statements about $U = \bar{X} - \mu$.

In this case,

$$\begin{aligned} H(y) &= \mathbb{P}[U \leq y] \\ &= \mathbb{P}[\frac{1}{s}U \leq \frac{1}{s}y] \\ &= \mathbb{P}[T \leq y/s] \\ &= \Phi(y/s) + n^{-1/2}p(y/s) + \mathcal{O}(n^{-1}), \end{aligned}$$

which uses the Edgeworth expansion again.

Similarly,

$$\begin{aligned} H^*(y) &= \mathbb{P}[U^* \leq y \,|\, \{X_I\}] \\ &= \Phi(y/s^*) + n^{-1/2}\hat{p}(y/s^*)\phi(y/s^*) + \mathcal{O}(n^{-1}). \end{aligned}$$

From asymptotics, it can be shown that:

$$\begin{aligned} p(y/s) - \hat{p}(y/s^*) &= \mathcal{O}_p(n^{-1/2}) \\ s - s^* &= \mathcal{O}_p(n^{-1/2}). \end{aligned}$$

Thus

$$H(y) - H^*(y) = \Phi(y/s) - \Phi(y/s^*) + n^{-1/2}[p(y/s)\phi(y/s) - \hat{p}(y/s^*)\phi(y/s^*)] + \mathcal{O}_p(n^{-1}).$$

The second term has order $\mathcal{O}_p(n^{-1})$ but the first has order $\mathcal{O}_p(n^{-1/2})$.

So when the statistic is not an asymptotic pivot, the bootstrap and the CLT have the same asymptotics. It pays to bootstrap a studentized pivot.