8. Issues with Bases

Nonparametric regression often tries to fit a model of the form

$$f(\boldsymbol{x}) = \sum_{j=1}^{M} \beta_j h_j(\boldsymbol{x}) + \epsilon$$

where the h_j functions may pick out specific components of $\boldsymbol{x}$ (multiple linear regression), or be powers of components of $\boldsymbol{x}$ (polynomial regression), or be prespecified transformations of components of $\boldsymbol{x}$ (nonlinear regression).

Many functions f cannot be represented by the kinds of h_j listed above. But if the set $\{h_j\}$ is an orthogonal basis for a space of functions that contains f, then we can exploit many standard strategies from linear regression.

8.1 Hilbert Spaces

Usually we are concerned with spaces of functions such as $\mathcal{L}^2[a,b]$, the Hilbert space of all real-valued functions defined on the interval $[a,b]$ that are square-integrable:

$$\int_a^b f(x)\,dx < \infty.$$

This definition extends to functions on $\mathbb{R}^p$.

Hilbert spaces have an inner product. For $\mathcal{L}^2[a,b]$ it is

$$< f, g >= \int_a^b f(x)g(x)\,dx.$$

The inner product defines a norm $\|f\|$, given by $< f, g >^{1/2}$, which is essentially a metric on the space of functions.

There are additional issues for a Hilbert space, such as completeness (i.e., the space contains the limit of all Cauchy sequences), but we can ignore those.

A set of functions $\{h_j\}$ in a Hilbert space is mutually orthogonal if for all $j \neq k$, $< h_j, h_k >= 0$.

Additionally, if $\|h_j\| = 1$ for all j, then the set is orthonormal.

If $\{h_j\}$ is an orthonormal basis for a space $\mathcal{H}$ then every function $f \in \mathcal{H}$ can be uniquely written as:

$$f(x) = \sum_{j=1}^{\infty} \beta_j h_j(x)$$

where

$$\beta_j =< f, h_j > .$$

Some famous orthogonal bases include the Fourier bases, consisting of $\{\cos nx, \sin nx\}$, wavelets, Legendre polynomials, and Hermite polynomials.

If one has an orthonormal basis for the space in which f lives, then several nice things happen:

1. Since f is a linear function of the basis elements, then simple regression methods allow us to estimate the coefficients β_j.
2. Since the basis is orthogonal, the estimates of different β_j coefficients are independent.
3. Since the set is a basis, there is no identifiability problem; each function in $\mathcal{H}$ is uniquely expressed as a weighted sum of basis elements.

But not all orthonormal bases are equally good. If one can find a basis that includes f itself, that would be ideal. Second best is a basis in which only a few elements in the representation of f have non-zero coefficients. But this problem is tautological since we do not know f.

One approach to choosing a basis is to use Gram-Schmidt orthogonalization to construct a basis set such that the first few elements correspond to the kinds of functions that statisticians expect to encounter.

This approach is often practical if the right kind of domain knowledge is available. This is often the case in audio signal processing; Fourier series are natural ways to represent vibration.

But in general, one must pick basis set without regard to f. A common criterion is that the influence of the basis elements be local. ¿From this standpoint, polynomials and trigonometric functions are bad, because their support is the whole line, but splines and wavelets are good, because their support is essentially compact.

Given an orthonormal basis, one can try estimating all of the $\{\beta_j\}$. But one quickly runs out of data—even if f is exactly equal to one of the basis elements, noise ensures that all of the other elements will make some contribution to the fit.

To address this problem one can:

- Restrict the set of functions, so it is no longer a basis (e.g., linear regression);
- select only those basis elements that seem to contribute significantly to the fit of the model (e.g., variable selection methods, or greedy fitters like MARS, CART, and boosting);
- regularize, to restrict the values of the coefficients (e.g., through ridge regression or shrinkage or thesholding).

The first technique is problematic, especially since it prevents flexiblity in fitting nonlinear functions.

The second technique is important, especially when one has large p but small n. But it is often insufficient, and the theory for aggressive variable selection is not well-developed yet.

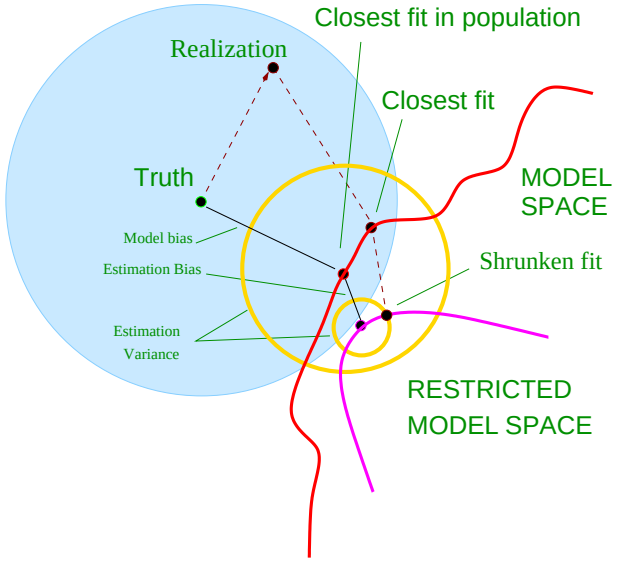


Figure 7.2: *Schematic of the behavior of bias and variance. The model space is the set of all possible predictions from the model, with the "closest fit" labeled with a black dot. The model bias from the truth is shown, along with the variance, indicated by the large yellow circle centered at the black dot labelled "closest fit in population". A shrunken or regularized fit is also shown, having additional estimation bias, but smaller prediction error due to its decreased variance.*

8.2 Ridge Regression

Ridge regression is an old idea, used originally to protect against multicollinearity. It shrinks the coefficients in a regression towards zero (and each other) by imposing a penalty on the sum of the squares of the coefficients.

$$\hat{\boldsymbol{\beta}} = \mathbf{argmin}_{\boldsymbol{\beta}}\{\sum_{I=1}^{n}(y_I - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij})^2 + \lambda\sum_{j=1}^{p}\beta_j^2\}$$

where $\lambda \geq 0$ is a penalty parameter that controls the amount of shrinkage.

Recall Stein's result that when estimating a multivariate normal mean with squared error loss, the sample average is inadmissible and can be improved by shinking the estimator towards $\mathbf{0}$ (or any other value).

Neural net methods now often do similar shrinkage on the weights at each node, Thereby improving predictive squared accuracy.

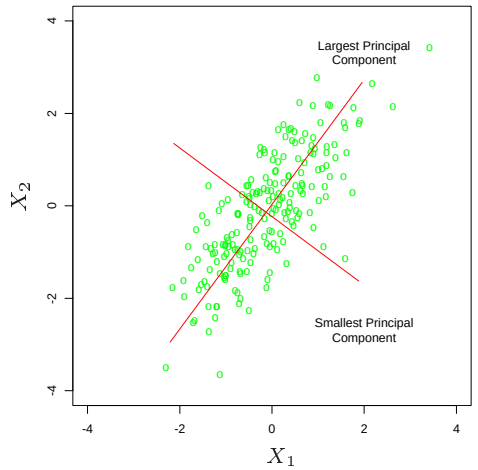


Figure 3.8: *Principal components of some input data points. The largest principal component is the direction that maximizes the variance of the projected data, and the smallest principal component minimizes that variance. Ridge regression projects* $\mathbf{y}$ *onto these components, and then shrinks the coefficients of the low-variance components more than the high-variance components.*

Ridge regression methods are not equivariant under scaling, so one normally standardizes the x_{ij} sample values wrt j so that each has unit variance.

Traditional ridge regression solves

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X}'\boldsymbol{X} + \lambda \boldsymbol{I})^{-1}\boldsymbol{X}'\boldsymbol{y}$$

so the name derives from the stabilization of the inverse matrix obtained by adding a constant to the diagonal.

Note that this increases the trace of the hat matrix, which corresponds to the degrees of freedom used in fitting the model.

Ridge regression (and shrinkage methods in general) can also be obtained as the Bayesian posterior mode for a suitable prior. In the multivariate normal case, the prior assumes independent normal distributions for each β_j, with common variance.

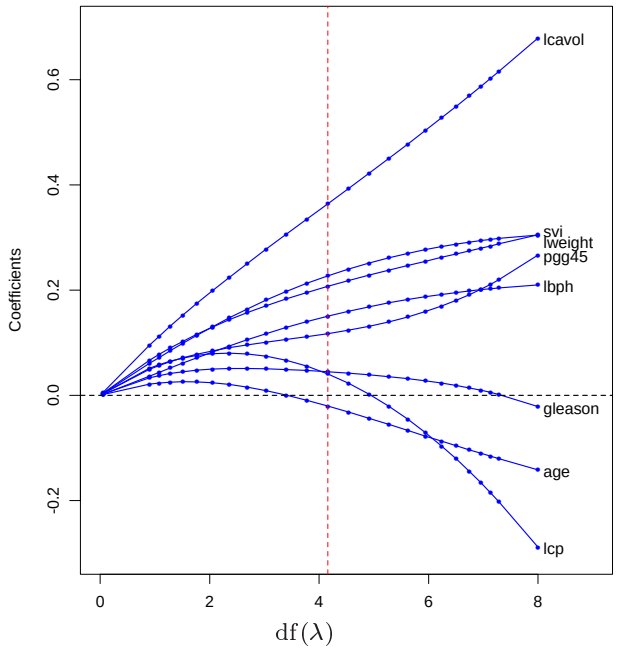


Figure 3.7: *Profiles of ridge coefficients for the prostate cancer example, as tuning parameter* λ *is varied. Coefficients are plotted versus* $\mathrm{df}(\lambda)$*, the effective degrees of freedom. A vertical line is drawn at* $\mathrm{df} = 4.16$*, the value chosen by cross-validation.*

8.3 The Lasso

The Lasso method is analogous to ridge regression. The Lasso estimate is given by

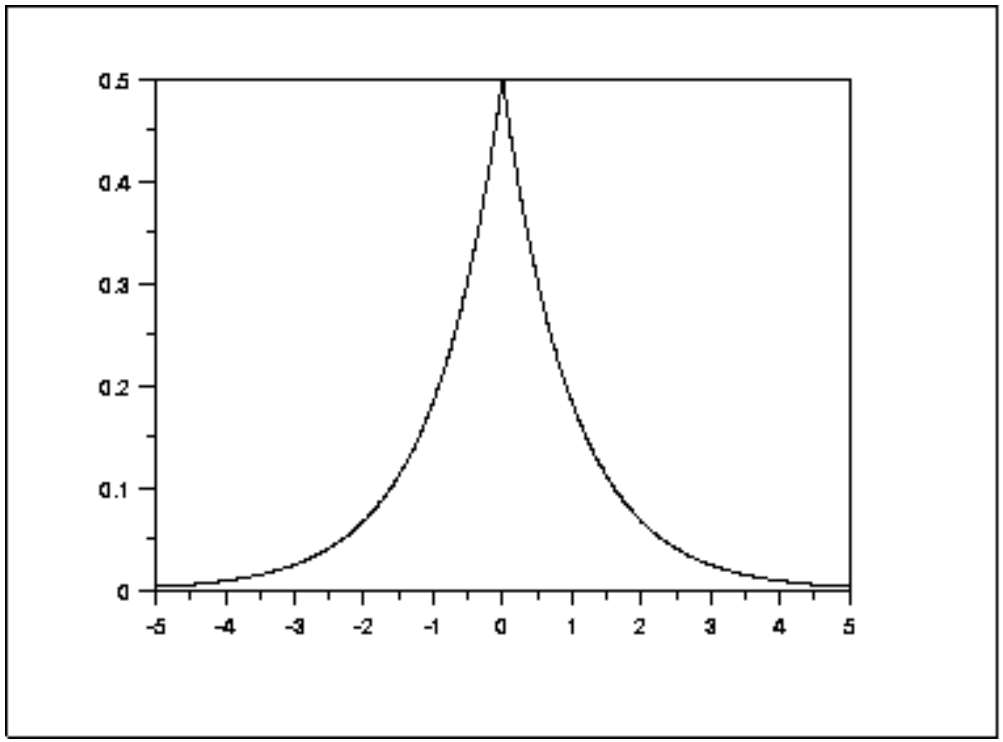
$$\hat{\boldsymbol{\beta}} = \mathbf{argmin}_{\boldsymbol{\beta}}\{\sum_{I=1}^{n}(y_I - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij})^2\}$$

subject to the constraint that $\sum_{j=1}^{p} |\beta_j| \leq s$.

The Lasso replaces the quadratic penalty in ridge regression by a penalty on the sum of the absolute values of the β_j terms.

If s is larger than the sum of the absolute values of the least squares estimators, then the Lasso agrees with OLS. If s is small, then many of the β_j terms are driven to 0, so it is performing variable selection.

The Lasso corresponds to a Bayesian method in which the prior on each parameter is Laplacian.



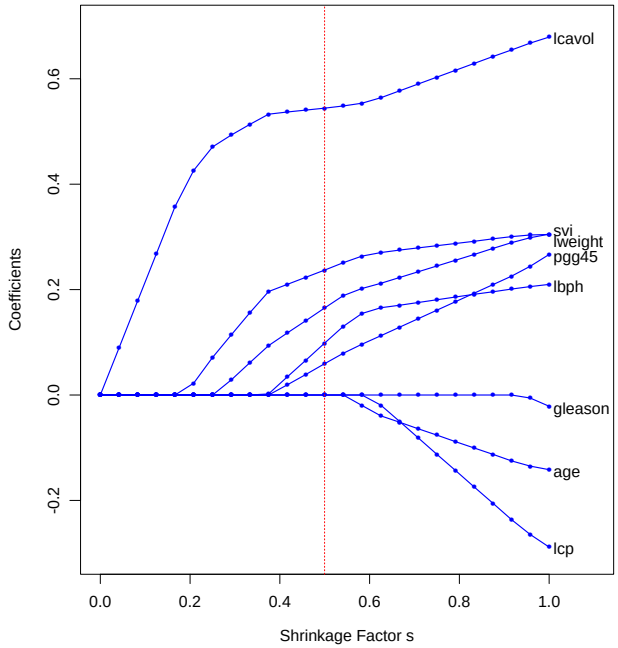


Figure 3.9: *Profiles of lasso coefficients, as tuning parameter t is varied. Coefficients are plotted versus $s = t/\sum_1^p |\hat{\beta}_j|$. A vertical line is drawn at $s = 0.5$, the value chosen by cross-validation. Compare Figure 3.7 on page 7; the lasso profiles hit zero, while those for ridge do not.*

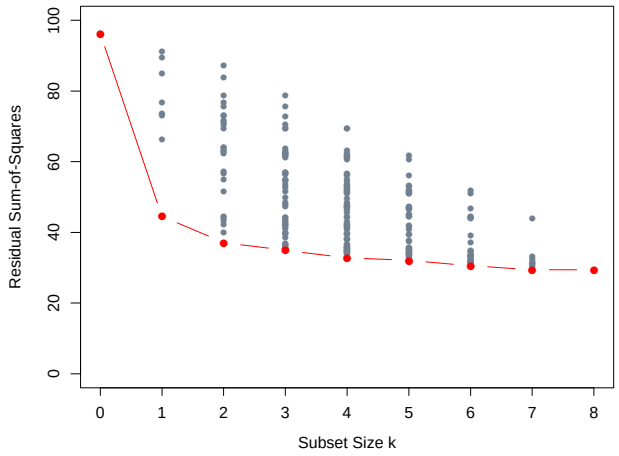


Figure 3.5: *All possible subset models for the prostate cancer example. At each subset size is shown the residual sum-of-squares for each model of that size.*

8.4 LARS

LARS was introduced in 2003 ("Least Angle Regression," Efron, Hastie, Johnstone, and Tibshirani, Annals of Statistics). It is closely related the the Lasso and forward stagewise modeling (the strategy that generated the boosting algorithm).

LARS was motivated by consideration of Forward Selection in multiple linear regression. Recall that Forward selection starts with no variables in the model, then adds the variable that best predicts the response. It then looks at the residuals from that fit, and adds the variable that best predicts those residuals. The process continues until no remaining variable is significantly associated with the residuals.

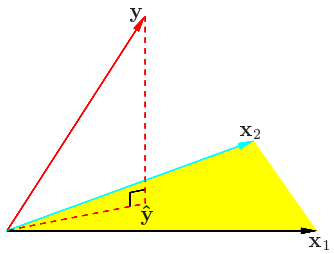


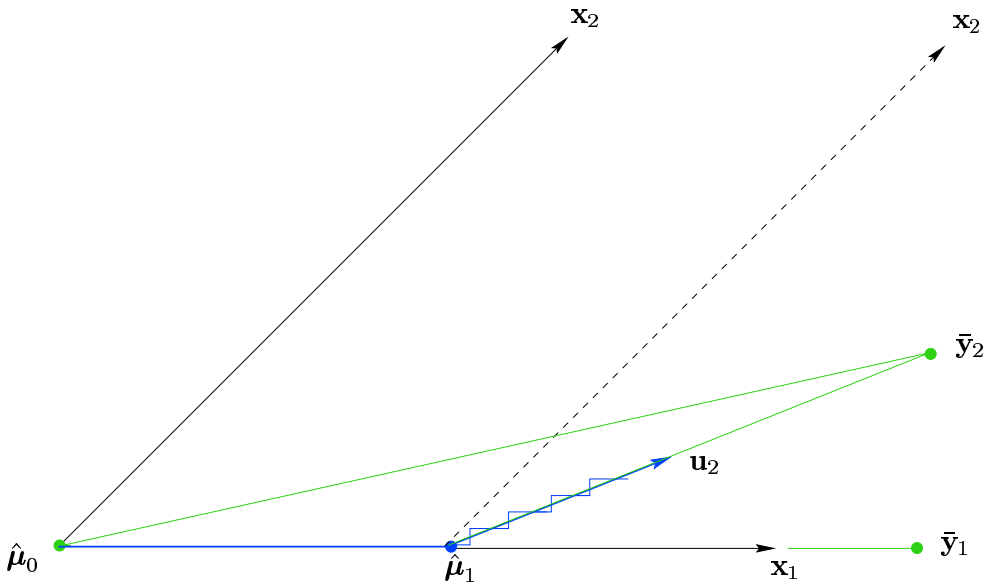
Figure 3.2: *The N-dimensional geometry of least squares regression with two predictors. The outcome vector* $\mathbf{y}$ *is orthogonally projected onto the hyperplane spanned by the input vectors* $\mathbf{x}_1$ *and* $\mathbf{x}_2$*. The projection* $\hat{\mathbf{y}}$ *represents the vector of the least squares predictions*

The geometry of regression is key. Note that linear regression estimates $\boldsymbol{\beta}$ by projecting the $\boldsymbol{Y}$ onto the subspace spanned by the explanatory variables: $\hat{\boldsymbol{\beta}} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{Y}$ so $\hat{\boldsymbol{Y}} = \boldsymbol{H}\boldsymbol{Y} = \boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{Y}$. Here $\boldsymbol{H}$ (the "hat matrix") is a projection matrix. See Christiansen (1987) for details.

The LARS strategy starts with all $\hat{\boldsymbol{\beta}}$ coefficients equal to zero. It then takes the largest "step" possible in the direction of the predictor most correlated with $\boldsymbol{Y}$. The step ends when some other predictor has as much correlation with the current residual. LARS then steps in the direction that bisects the angle between the two most-correlated predictors, and continues until a third variable becomes as correlated with the residuals as the first two. LARS then moves equiangularly between those three variables, and so on.

LARS differs from Forward Selection in that at the end of the first step, Forward Selection would continue to move in the direction of the first predictor, until the correlation between it and the current residuals was reduced to zero.

LARS is closely related to the LASSO and to Forward Stagewise modeling, although this is not obvious.



The LARS algorithm for $p = 2$ predictors. The $\bar{y}_2$ is the projection of y on the space spanned by $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, the columns of the $\boldsymbol{X}$ matrix. The LARS prediction at step k is $\hat{\boldsymbol{\mu}}_k = \boldsymbol{X}\hat{\boldsymbol{\beta}}_k^L$ where $\hat{\boldsymbol{\beta}}_k^L$ is the LARS estimate at step k.

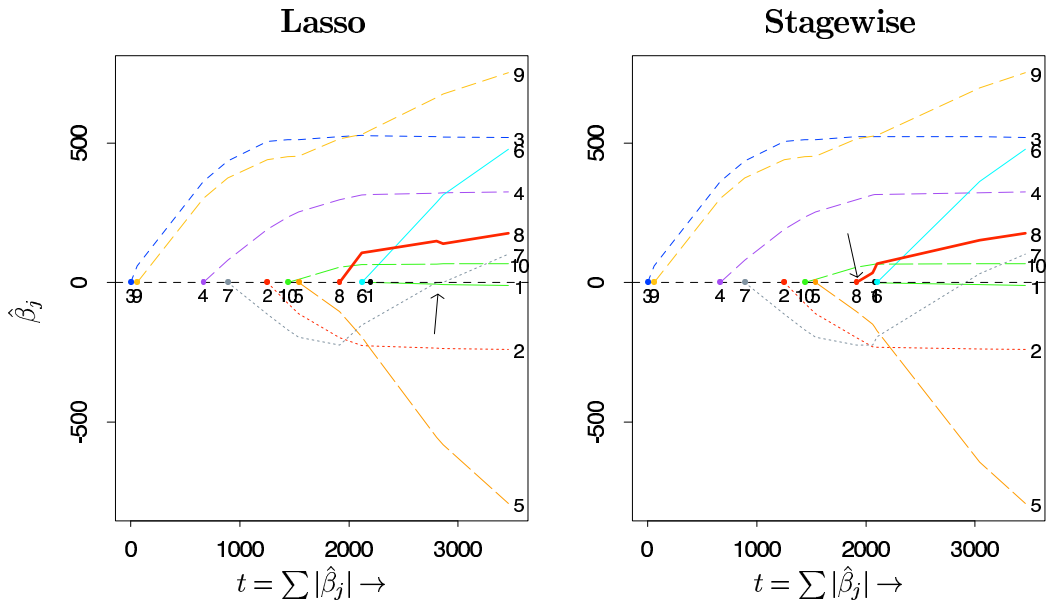
Beginning at $\boldsymbol{\mu}_0 = \mathbf{0}$, the residual vector $\bar{\boldsymbol{y}} - \hat{\boldsymbol{\mu}}_0$ has larger correlation with $\boldsymbol{x}_1$ than $\boldsymbol{x}_2$. So LARS steps in the direction of $\boldsymbol{x}_1$ as far as it can until the current residuals are as correlated with $\boldsymbol{x}_2$ as with $\boldsymbol{x}_1$.

That first step gives two estimates, $\hat{\boldsymbol{\mu}}_1$ and, implicitly, $\hat{\boldsymbol{\beta}}_1^L$. Here $\hat{\boldsymbol{\mu}}_1 = \hat{\boldsymbol{\mu}}_0 + \hat{\gamma}_1 \boldsymbol{x}_1$ where $\hat{\gamma}_1$ is chosen so that $\bar{\boldsymbol{y}}_2 - \hat{\boldsymbol{\mu}}_1$ bisects the angle between $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$.

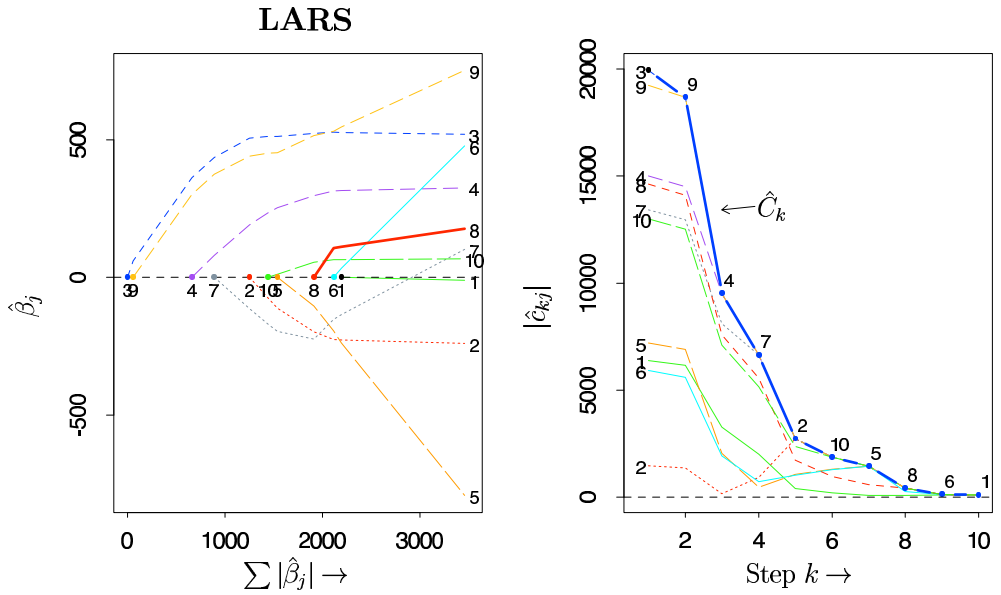
The second step moves along the unit vector $\boldsymbol{u}_2$ that bisects the angle. It goes distance $\hat{\gamma}_2$, and terminates at the OLS estimator $\bar{\boldsymbol{y}}_2$.

Some comments:

- For p explantory variables, the LARS estimate arrives at the OLS estimate on step p.
- The Forward Selection algorithm would, on the first step, proceed along the $\boldsymbol{x}_1$ direction until it reaches the projection of $\bar{\boldsymbol{y}}_2$ onto the $\boldsymbol{x}_1$ subspace at $\bar{\boldsymbol{y}}_1$, as shown on the graph.
- A Stagewise selection procedure fits at each step a model that improves, in one variable, the fit. This is shown by the staircase in the figure. It necessarily takes small steps, and thus is computationally intensive.
- If two predictors are perfectly correlated, LARS would use both and put equal weight upon them.
- The Lasso can be obtained as a slight modification of the LARS algorithm (see section 3.1 in the Efron et al. paper).
- LARS, the Lasso, and Forward Stagewise regression give very similar answers.



Lasso and Stagewise regression estimates for the diabetes study. Note that both are automatically parsimonious.



LARS regression estimates for the diabetes study. Note that it is also automatically parsimonious.

8.5 Bayesian Methods

Suppose there are p explanatory variables and one considers a model of the form:

$$\mathcal{M}_i : \boldsymbol{Y} = \boldsymbol{X}_i^* \boldsymbol{\beta}_i^* + \boldsymbol{\epsilon}$$

where $\boldsymbol{\epsilon} \sim N(\mathbf{0}, \sigma^2 \boldsymbol{I})$ and $i = 1, \ldots, 2^p$, indexing all possible choices of inclusion or exclusion among the p explanatory variables.

Assume there is an intercept: each $\boldsymbol{\beta}_i^* = (\beta_0, \boldsymbol{\beta}_i')$, and thus let $\boldsymbol{X}_i$ be the submatrix of $\boldsymbol{X}_i^*$ corresponding to $\boldsymbol{\beta}_i$, and let the length of $\boldsymbol{\beta}_i$ be $d(i) + 1$. Denote the density corresponding to model $\mathcal{M}_i$ by $f_i(\boldsymbol{y} | \boldsymbol{\beta}_i^*, \sigma^2)$.

For this model, standard choices for the prior on the parameters in $\mathcal{M}_i$ include:

- The g-priors. Here:

$$\pi_i^g(\boldsymbol{\beta}_i, \sigma^2) = \frac{1}{\sigma^2} N(\boldsymbol{\beta}_i | \mathbf{0}, cn\sigma^2 (\boldsymbol{X}'_i \boldsymbol{X}_i)^{-1})$$

  where c is fixed (typically to 1, or estimated through empirical Bayes).

- Zellner-Siow priors. The intercept model has prior $\pi(\beta_0, \sigma^2) = 1/\sigma^2$, while the prior for all other $\mathcal{M}_i$ is:

$$\begin{aligned} \pi_i^{ZS}(\boldsymbol{\beta}_i, \sigma^2) &= \pi_i^g(\boldsymbol{\beta}_i, \sigma^2 | c) \\ \pi_i^{ZS}(c) &\sim \textbf{InverseGamma}(c | 0.5, 0.5). \end{aligned}$$

The **marginal density** under $\mathcal{M}_i$ is

$$m_i(\boldsymbol{Y}) = \int f_i(\boldsymbol{y} | \boldsymbol{\beta}_i, \sigma^2) \pi(\boldsymbol{\beta}_i, \sigma^2) d\boldsymbol{\beta}_i d\sigma^2.$$

If all models $\mathcal{M}_i$ have equal prior probability, then the posterior probability of a model is

$$P(\mathcal{M}_i|\boldsymbol{Y}) = \frac{m_i(\boldsymbol{y})}{\sum_k m_k(\boldsymbol{Y})}.$$

Recall the posterior inclusion probability for the ith variable:

$$\begin{aligned} q_i &= P(\beta_i \in \textbf{ correct model } |\boldsymbol{Y}) \\ &= \sum_k P(\mathcal{M}_k|\boldsymbol{Y}) I_{\beta_i \in \mathcal{M}_k}. \end{aligned}$$

The **median probability model** is the model consisting of all and only those variables whose posterior inclusion probability is at least 1/2.

Usually, the median probability model is superior to the maximum posterior probability model for prediction. (But the Bayesian model average is better than both, if one can use an ensemble method.)

Theorem: (Barbieri and Berger, 2004, Annals of Statistics. Consider a sequence of nested linear models. If

- prediction is wanted at "future covariates like the past",
- the posterior mean under $\mathcal{M}_i$ satisfies $\tilde{\boldsymbol{\beta}}_i = b\hat{\boldsymbol{\beta}}_i$, where $\hat{\boldsymbol{\beta}}_i$ is the OLS estimate,

then the best single model for prediction under squared error loss is the median probability model.

The second condition is satisfied if one uses either noninformative priors for the model parameters or if one uses g-type priors with the same constant $c > 0$ for each model (and any prior on σ^2).

Example: Polynomial regression. Here $\mathcal{M}_i$ is

$$y = \sum_{j=0}^{i} \beta_j x^j + \epsilon.$$

Model	0	1	2	3	4	5	6
$\mathrm{P}(\mathcal{M}_i \mid \boldsymbol{Y})$	≈ 0	.06	.22	.29	.38	.05	≈ 0

Covariate j	0	1	2	3	4	5	6
P(x^j is in model$\mid \boldsymbol{Y}$)	≈ 1	≈ 1	.94	.72	.33	.05	≈ 0

The correct model was $\mathcal{M}_3$, which is the median probability model. But the MAP model is $\mathcal{M}_4$.

8.6 Overcompleteness

Statisticians traditionally have used an orthogonal basis of functions $\{h_j\}$ for estimating functions f:

$$\hat{f}(\boldsymbol{x}) = \sum_{j=1}^{M} \hat{\beta}_j h_j(\boldsymbol{x}).$$

This choice is largely motivated by the independence of the coefficient estimates and the ability to do asymptotic theory.

But computer scientists have found that using larger sets of functions than are available in the orthogonal basis can improve performance. See Wolfe, Godsill, and Ng (2004; *Journal of the Royal Statistical Society, Series B*, **66**, 1-15).

This larger set is called a frame. A frame contains a basis for the space of interest (e.g., $\mathcal{L}^2[a,b]$), but may also other functions. Formally, a frame is a set of functions $\{h_j : j \in J\}$ with the property that there are constants $A, B > 0$ such that

$$A\|f\|^2 \leq \sum_{j\in J} |< f, h_j >|^2 \leq B\|f\|^2 \quad \forall f \in \mathcal{H}.$$

A frame that is just an ordinary basis has $A = B = 1$. A frame that is the union of two ordinary bases has $A = B = 2$. Some frames have uncountable cardinality.

Nontrivial frames contain at least one non-zero element which can be written as a linear combination of the other elements in the frame. This contrasts with the situation for basis sets.

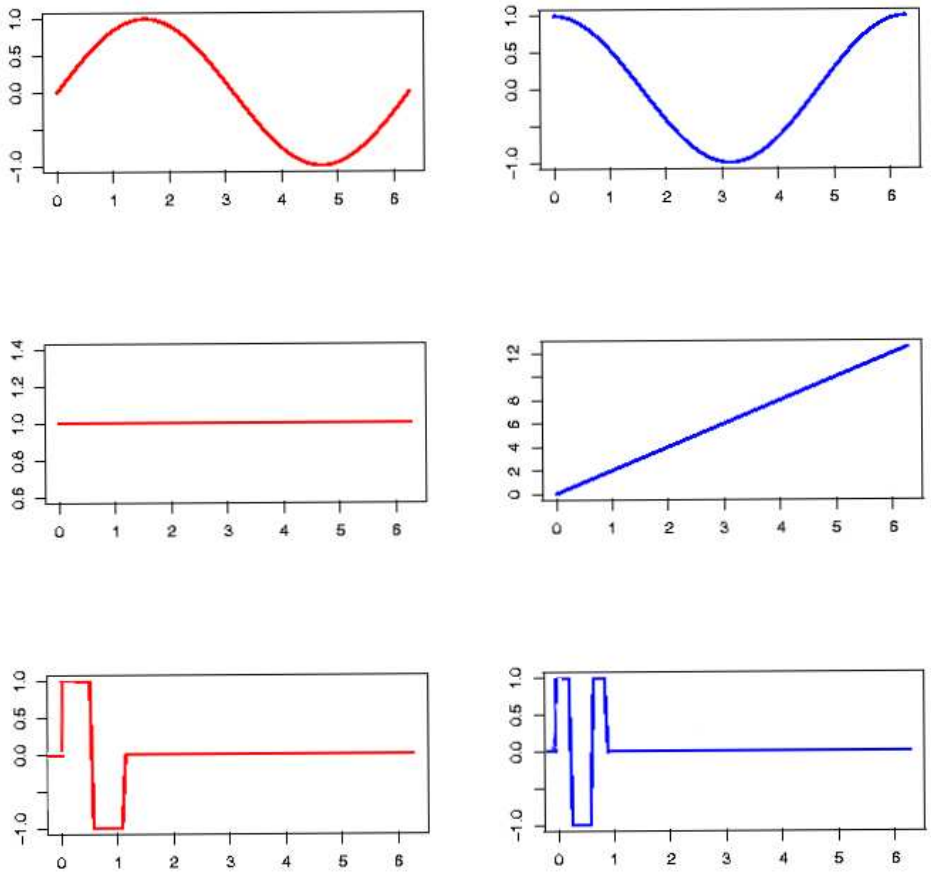
The advantage of frames is that their greater size allows the possibility of finding a *very* parsimonious representation for f.

One wants a criterion for comparing the performance of different frames used for function estimation.

Consider estimating functions in $\mathcal{L}^2[a, b]$. If one forms an overcomplete frame as the union of two basis sets, is it better to take the union of a Fourier basis and a Hermite polynomial basis, or the union of a Fourier basis and the Haar basis?

One imagines that it would be desirable to combine bases that contain very different functions. From this perspective, the Fourier basis, which is smooth, might be more effectively combined with the rough Haar basis than the smooth Hermite polynomial basis. Thus the resulting frame would contain elements that allow parsimonious representation of both smooth and rough functions.

The following figure shows the first two elements of these three bases.


1.0
0.5
0.0
-1.0
0 1 2 3 4 5 6
1.4
1.2
1.0
0.8
0.6
12
8
6
4
2
0

One proposal for a criterion that compares two frames depends upon the parsimony of the approximating functions.

For a given frame $F = \{ h_j : j \in J \}$, a function f, and a tolerance ϵ^*, consider the set of all approximating functions

$$S(f, \epsilon^*) = \{ \hat{f} : \hat{f} = \sum_{j=1}^{k} \beta_j h_j \textbf{ and } \|f - \hat{f}\| < \epsilon^* \}.$$

Each $\hat{f} \in S(f, \epsilon^*)$ has a certain number of terms in the sum. The most parsimonious approximation is the one that uses the fewest terms h_j. Let

$$k(f, \epsilon^*) = \inf\{j : \hat{f} \in S(f, \epsilon^*)\}.$$

Some functions $f \in L^2[0, 1]$ will be hard to approximate with elements in the frame F, and for these functions $k(f, \epsilon^*)$ will be large (and maybe infinite).

To handle the worst case for estimation, let

$$k(\epsilon^*) = \sup_{f \in L^2[0,1]} \{ k(f, \epsilon^*) \}.$$

This is the number of terms needed to approximate the most difficult function in $L^2[0,1]$ to within ϵ^* using only frame elements.

For some frames and values ϵ^*, $k(\epsilon^*)$ will be infinite. But there are many cases in which it is finite. For example, if F contains an ϵ^*-net, this trivially ensures that $k(\epsilon^*) = 1$.

For non-trivial cases in which frames F and G are unions of bases, one would like to say that frame F is better than frame G at level ϵ^* if $k_F(\epsilon^*) < k_G(\epsilon^*)$, where the subscript indicates the frame.

If there exists some γ such that the inequality holds for all $\epsilon^* < \gamma$, then one could broadly claim that F is better than G.

9. Wavelets

A wavelet is a function that looks like a localized wiggle. Special collections of wavelets can be used to obtain approximations of functions by separating then information at different scales.

The stunning success of wavelets has spurred explosive growth in research. Donoho and Johnstone (1994; *Biometrika*, **81**, 425-455) led the way in statistics, showing that wavelets can achieve local asymptotic minimaxity in approximating thick classes of functions.

Local asymptotic minimaxity is a technical property, but it ensures that the estimates are asymptotically minimax with respect to a large family of loss functions in a large class of functions. Such estimates probably evade the COD, in the same technical sense that neural nets do.

In its simplest form, a wavelet representation begins with a single function ψ, often called th mother wavelet.

We define a collection of wavelets—called a wavelet basis—by dilation and translation of the mother wavelet ψ:

$$\psi_{jk}(t) = 2^{j/2}\psi(2^j t - k)$$

for $j, k \in \boldsymbol{Z}$, the set of integers.

One obtains an overcomplete frame if one allows j, k to take values in $\mathbb{R}$ rather than $\boldsymbol{Z}$. This is called an *undecimated* wavelet basis.

Each ψ_{jk} has a characteristic resolution scale (determined by j) and is roughly centered on the location $k/2^j$. It turns out that if ψ is properly chosen, any "reasonable" function can be represented by an infinite linear combination of the ψ_{jk} functions.

Wavelets have three key features:

- Wavelets provide *sparse* representations of a broad class of functions and signals,
- Wavelets can achieve very good *localization* in both time and frequency,
- There are *fast algorithms* for computing wavelet representations in practice.

Sparsity means that most of coefficients in the linear combination are nearly zero. A single wavelet basis can provide sparse representations of many spaces of functions at the same time.

Good localization means that both the wavelet function ψ *and* its Fourier transform have small support; i.e., the functions are essentially zero outside some compact domain.

Regarding speed, wavelet representations can be computed in $\mathcal{O}(n \log n)$ and sometimes $\mathcal{O}(n)$ operations (the FFT is $\mathcal{O}(n \log n)$).

9.1 Constructing Wavelets

We begin with a mathematical description of a smooth localized wiggle.

Let $\mathcal{L}^p(\mathbb{R})$ denote the space of measurable complex-valued functions f on the real numbers $\mathbb{R}$ such that

$$\|f\|_p = \left[\int |f(x)|^p\right]^{1/p} dx < \infty.$$

Here $\mathcal{L}^2(\mathbb{R})$ is a Hilbert space with inner product defined by

$$< f, g >= \int f(x)\bar{g}(x)\, dx,$$

where $\bar{g}$ is the complex conjugate of g.

Recall that a complex function $g(x) = u(x) + iv(x)$ has conjugate $\bar{g}(x) = u(x) - iv(x)$ where $u(x)$ and $v(x)$ are real-valued functions and i is $\sqrt{-1}$. The modulus of $g(x)$ is $\sqrt{u^2(x) + v^2(x)}$.

For integers $D, M \geq 0$, suppose that $\psi \in \mathcal{L}^2(\mathbb{R})$ satisfies the following for $d = 0, \ldots, D$ and $m = 0, \ldots, M$:

1. The derivative $\psi^{(d)}$ exists and is in $\mathcal{L}^\infty(\mathbb{R})$,
2. The modulus $|\psi^{(d)}|$ is rapidly decreasing as $|x| \to \infty$, and
3. The mth moment of ψ vanishes, i.e., $\int x^m \, \psi(x) \, dt = 0$.

Then we say that ψ is a basic wavelet of regularity (D, M).

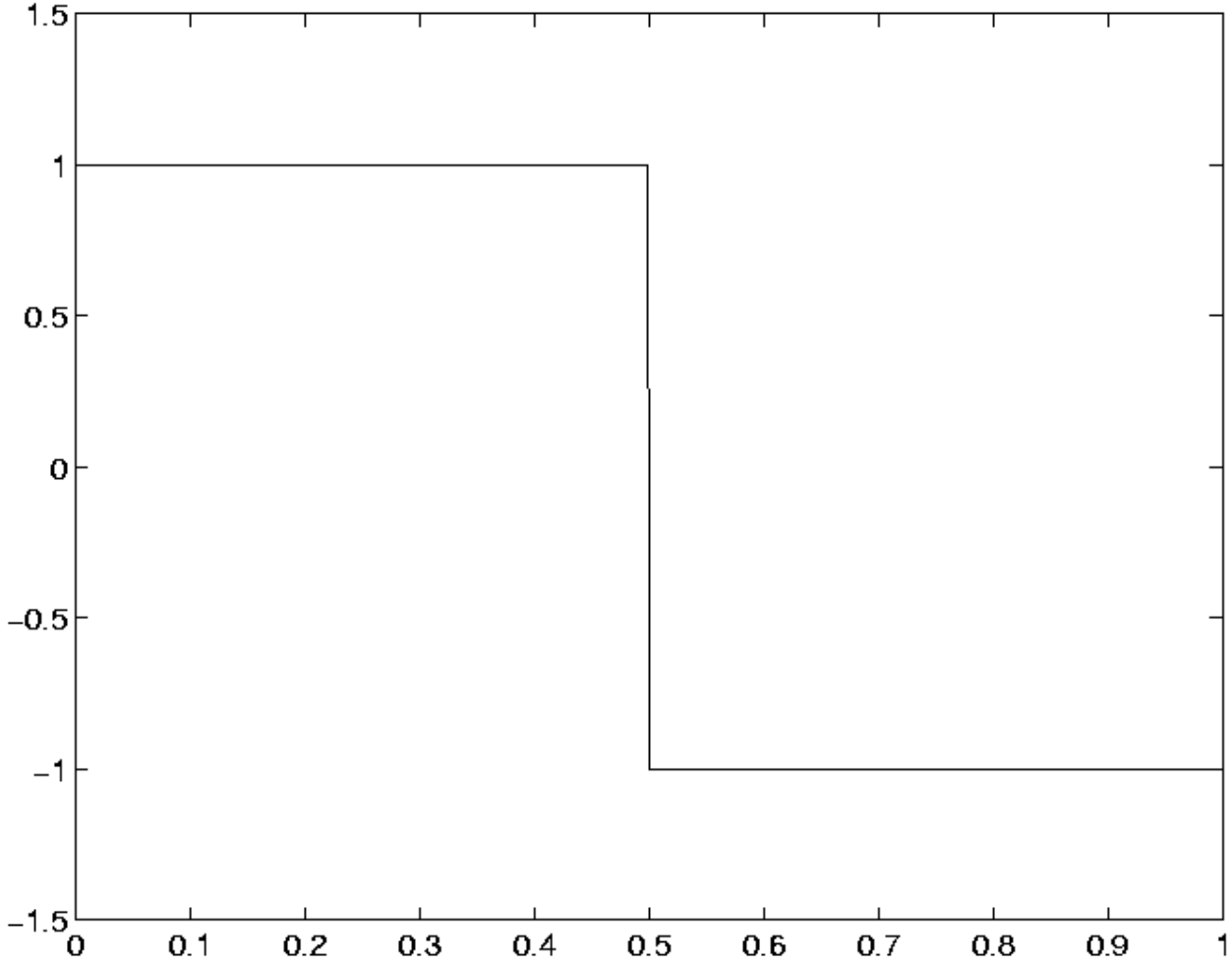
Condition 1 implies that ψ is smooth, and conditions 1 and 2 together imply that ψ is localized in time and frequency.

Condition 2 implies that ψ is highly concentrated in the x domain by requiring that it decrease faster than any inverse polynomial outside some compact set.

Condition 1 implies that the Fourier transform of ψ is quite concentrated as well, decreasing faster than the reciprocal of a D-degree polynomial for high frequency.

Condition 3 forces ψ to be "wiggly" since the integral of ψ with any polynomial up to degree M must be zero.

The standard example of a wavelet basis is the Haar Basis. Let $\psi = 1_{[0,1/2)} - 1_{[1/2,1)}$ where 1_A is the indicator function of the set A. Then the mother wavelet is:



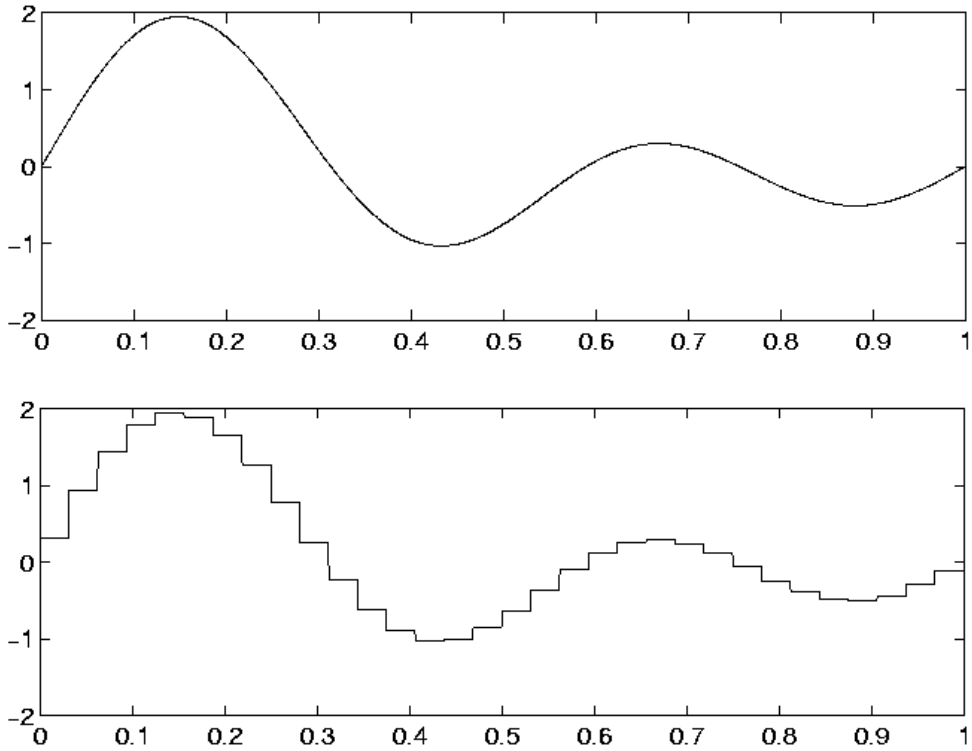
This function is not differentiable, but it has compact support and integrates any constant to zero. Thus ψ is a basic wavelet of regularity $(0, 0)$.

If we define ψ_{jk} from ψ by translation and dilation, we obtain a doubly-infinite set of functions with the same properties as ψ.

Moreover, any two ψ_{jk} are orthonormal in the sense that $< \psi_{j,k}, \psi_{j',k'} >= \delta_{jj'}\delta_{kk'}$ where δ is a Kronecker delta.

These ψ_{jk} form an orthonormal basis for $\mathcal{L}^2(\mathbb{R})$. any function in $\mathcal{L}^2(\mathbb{R})$ can be well-approximated by $\sum_{j,k} \beta_{j,k}\psi_{jk}$. (As a first step in seeing this, note that it is sufficient to be able to approximate any function that is piecewise constant on dyadic intervals.)

Since the Haar basis consists of step functions, an approximation of a smooth function by a finite number of Haar wavelets is necessarily ragged.



The construction of smoother variants of the Haar basis leads to more general wavelets.

A wavelet basis is especially useful if it can extract interpretable information about $f \in \mathcal{L}^2(\mathbb{R})$ through the inner product $\beta =< f, \psi >$.

If ψ is well-localized in $\mathbb{R}$, then a β coefficient essentially reflects the behavior of f on a particular part of its domain (unlike the case in linear regression).

Similarly, if ψ is well-localized in frequency, then β essentially reflects particular frequency components of f as well.

Suppose that ψ is concentrated in a small interval and that in that interval f can be approximated by a Taylor series; then, since the inner product with ψ cancels all polynomials up to degree M, β reflects only the higher-order behavior in f. (Another way to state this is that if two functions differ only by a polynomial of degree M or smaller, their β coefficient will be the same.)

For a basis $\{\psi_{jk}\}$, the index j represents resolution scale; as j increases, ψ_{jk} becomes concentrated in a decreasing set.

The index k represents location; as k changes, the function is shifted along the line. Hence, the map $(j,k) \mapsto \beta_{j,k} =< f, \psi_{jk} >$ provides information about f at every dyadic scale and location.

The wavelet representation

$$f(x) = \sum_{j \in Z} \sum_{k \in Z} \beta_{jk} \psi_{jk}(x)$$

is called the homogenous equation. It expresses f in terms of functions that all integrate to 0. (There is no paradox here, since convergence in $\mathcal{L}^2(\mathbb{R})$ and convergence in $\mathcal{L}^1(\mathbb{R})$ are not the same thing.)

9.2 The Father Wavelet

It is often useful to re-express the homogenous equation in an equivalent but inhomogenous form. This form separates "coarse" structure from "fine" structure.

To do this, select a convenient resolution level J_0. Resolution levels $j \leq J_0$ capture the coarse structure of f and levels $j > J_0$ capture the fine structure.

One approximates the fine-resolution structure of f by a linear combination of the ψ_{jk} at $j \geq J_0$, and one approximates the coarse-resolution structure by a combination of the functions $\phi_k(t) = \phi(t-k)$ for $k \in \boldsymbol{Z}$, where ϕ is called the father wavelet(or scaling function).

The inhomogenous representation for $f \in \mathcal{L}^2(\mathbb{R})$ is then

$$f = \sum_{k \in \boldsymbol{Z}} < f, \phi_k > \phi_k + \sum_{j \geq J_0} \sum_{k \in \boldsymbol{Z}} < f, \psi_{jk} > \psi_{jk}.$$

This father wavelet ϕ is closely related to the mother wavelet ψ:

- one can choose ϕ so that $< \phi, \psi >= 0$,
- the father wavelet satisfies the first two requirements of a basic wavelet with the same indices of regularity as ψ.

The difference between the inhomogenous and homogenous representations lie in the first sum. This aggregates the information in the low resolution levels through the specially constructed function ϕ rather than distributing it among the ψ_{jk} terms.

The prime advantage of this inhomogenous representation is that the coarse/fine dichotomoy is intuitively appealing and useful. This is especially the case if the noise in the function estimation can be thought of as high-frequency, while the signal in the problem is low-frequency.

For the Haar Basis, the father wavelet is the indicator of the unit interval, $\phi = 1_{[0,1)}$.

When $J_0 = 0$, the inhomogenous representation takes the form

$$f = \sum_{k \in \boldsymbol{Z}} \alpha_k \, \phi_k \; + \sum_{j \geq 0} \sum_{k \in \boldsymbol{Z}} \beta_{j,k} \, \psi_{jk},$$

where $\alpha_k =< f, \phi_k >$ and $\beta_{j,k} =< f, \psi_{jk} >$.

The coefficients α_k are just integrals of f over intervals of the form $[k, k+1)$; the remaining structure in f is represented as fluctuations within those intervals.

Note that ϕ is in $\mathcal{L}^2(\mathbb{R})$, and consequently can be written in terms of the ψ_{jk}:

$$\phi = \frac{1}{2} \, \psi_{-1,0} + \frac{1}{2} \sum_{j=2}^{\infty} 2^{-j/2} \psi_{-j,0}.$$

This uses only the coarsest ψ_{jk} terms, as expected.

The relationship between ψ and ϕ goes the other way as well: $\psi = 1_{[0,1/2)} - 1_{[1/2,1)}$ by definition, which is a linear combination of translated and dilated ϕ terms.

9.3 Multiresolution Analysis

A key idea in wavelets is that of successive refinement. If one can approximate at several levels of accuracy, then the differences between successive approximations characterize the refinements needed to move from one level to another.

Multiresolution analysis (MRA) formalizes this notion for approximations that are related by a translation and dilation. An MRA is a sequence of nested approximation spaces for a containing class of functions.

For the containing class $\mathcal{L}^2(\mathbb{R})$, MRA is a nested sequence of closed subspaces

$$\cdots \subset V_{-2} \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \subset \cdots$$

such that

1. $\mathbf{clos}\left\{\bigcup_{j \in \boldsymbol{Z}} V_j\right\} = \mathcal{L}^2(\mathbb{R})$
2. $\bigcap_{j \in \boldsymbol{Z}} V_j = \{0\}$
3. $f \in V_j \Leftrightarrow f(2^{-j}\cdot) \in V_0 \quad \forall\, j \in \boldsymbol{Z}$
4. $f \in V_0 \Leftrightarrow f(\cdot - k) \in V_0 \quad \forall\, k \in \boldsymbol{Z}.$

Condition 1 ensures that the approximation spaces (V_j) are sufficient to approximate any function in $\mathcal{L}^2(\mathbb{R})$.

There are many sequences of spaces satisfying Conditions 1, 2 and 4; the name "multiresolution" is derived from Condition 3 which implies that all the V_j are dyadically scaled versions of a common space V_0. By Condition 4, V_0 is invariant under integer translations.

When generating orthonormal wavelet bases, we also require that the space V_0 of the MRA contains a function ϕ such that the integer translations $\{\phi_{0,n}\}$ form an orthonormal basis for V_0. For simplicity, we will focus almost exclusively on orthonormal bases here.

Since $V_0 \subset V_1$, we can define its orthogonal complement W_0 in V_1, so $V_1 = V_0 \oplus W_0$. We can do likewise for every j, where $V_{j+1} = V_j \oplus W_j$.

The sequence of spaces $\{W_j\}_{j \in \mathbb{Z}}$ are mutually orthogonal and they are inherit Condition 3 from the V_j; i.e., $f \in W_j$ if and only if $f(2^{-j}\cdot) \in W_0$.

By themselves, these spaces provide a homogeneous representation of $\mathcal{L}^2(\mathbb{R})$, since $\mathcal{L}^2(\mathbb{R}) = \text{clos}\{\cup_{j \in \mathbb{Z}} W_j\}$. The W_j are the building blocks for successive refinement from one approximating space to another.

Given $f \in \mathcal{L}^2(\mathbb{R})$, the best approximation to f in any V_j is given by $P_j f$, where P_j is the orthogonal projection onto V_j. It follows that $Q_j = P_j - P_{j-1}$ is the orthogonal projection onto W_j.

Given a coarse approximation $P_0 f$, one can refine to the finer approximation $P_J f$ for any $J > 0$ by adding details from successive W_j spaces:

$$\begin{aligned} P_J f &= P_0 f + \sum_{j=1}^{J} P_j f - P_{j-1} f \\ &= P_0 f + \sum_{j=0}^{J-1} Q_j f, \end{aligned}$$

and as $J \to \infty$,

$$f = P_0 f + \sum_{j=0}^{\infty} Q_j f.$$

Successive refinement exactly mimics the inhomogenous wavelet representation. The coarse approximation $P_0 f$ corresponds to a linear combination of the $\phi_{0,k}$, and each $Q_j f$ for $j \geq 0$ corresponds to a linear combination of the span of the ψ_{jk}.

As an example, suppose V_j is the set of piecewise constant functions on intervals of the form $[k2^{-j}, (k+1)2^{-j})$. So V_0 is generated by integer translations of the function $\phi = 1_{[0,1)}$, the father wavelet for the Haar basis.

For $f \in \mathcal{L}^2(\mathbb{R})$, let $\alpha_{j,k} =< \phi_{j,k}, f >$. Then $P_0 f = \sum_k \alpha_{0,k}\phi_{0,k}$ and $P_1 f = \sum_k \alpha_{1,k}\phi_{1,k}$ are approximations to f that are piecewise constant on unit and half-unit intervals, respectively.

How does one refine the coarse approximation $P_0 f$ to the next higher resolution level?

We know $\alpha_{0,k} = \frac{1}{\sqrt{2}}(\alpha_{1,2k} + \alpha_{1,2k+1})$, but we also need to know the difference $\beta_{0,k} = \frac{1}{\sqrt{2}}(\alpha_{1,2k} - \alpha_{1,2k+1})$ between the integrals of f over the half-intervals $[k, k+1/2)$ and $[k+1/2, k+1)$. This $\beta_{0,k}$ is just the coefficient $< \psi_{0,k}, f >$ of the $(0,k)$ Haar wavelet $\psi_{0,k}$. The translations of the Haar mother wavelet form an orthonormal basis for the space W_0.

For a general MRA, how does one find the corresponding ψ?

It turns out that ψ and ϕ determine each other through the refinement relations among the spaces. So one can construct a function ψ whose integer translations (i.e., $\psi_{0,k}(t) = \psi(t-k)$) yield an orthonormal basis of W_0.

By construction, both ψ and ϕ are in V_1, and the $\psi_{0,k}$ and $\phi_{0,n}$ are orthogonal. It follows that both ϕ and ψ can be expressed as a linear combination of the $\phi_{1,n}$ with some constraints on the coefficients.

This reasoning leads to the following two-scale identities:

$$\begin{aligned} \phi(t) &= \sqrt{2} \sum_n g_n \, \phi(2t-n) \\ \psi(t) &= \sqrt{2} \sum_n h_n \, \phi(2t-n). \end{aligned}$$

The $\{h_n\}$ and $\{g_n\}$ satisfy $\sum_n |h_n|^2 = 1$ and $\sum_n |g_n|^2 = 1$. Orthogonality of the $\phi_k(t)$ and $\psi_{0,k}$ and the fact that V_1 is a direct sum of V_0 and W_0 force relationships among $\{g_n\}$ and $\{h_n\}$.

For a specific MRA, these relationships provide enough conditions to uniquely identify mother and father wavelets ϕ and ψ.

In the Haar case, the sequences are

$$\begin{aligned} \{g_n\} &= (\ldots, 0, 1/\sqrt{2}, 1/\sqrt{2}, 0, \ldots) \\ \{h_n\} &= (\ldots, 0, 1/\sqrt{2}, -1/\sqrt{2}, 0, \ldots). \end{aligned}$$

In general, it is more convenient to work with the two-scale identities in the Fourier domain so as to characterize the Fourier transforms ϕ^* and ψ^*.

The value of the two-scale identities is in the connections they imppose on the wavelet coefficients. By the two-scale identities,

$$\begin{aligned} \phi_{jk}(t) &= 2^{(j+1)/2} \sum_n g_n \phi(2^{j+1}t - 2k - n) \\ \psi_{jk}(t) &= 2^{(j+1)/2} \sum_n h_n \phi(2^{j+1}t - 2k - n). \end{aligned}$$

It follows that

$$\begin{aligned} < \phi_{jk}, f > &= \sum_n g_n \ < \phi_{j+1,2k+n}, f > \\ &= \sum_n g_{n-2k} \ < \phi_{j+1,n}, f > \end{aligned}$$

and

$$\begin{aligned} < \psi_{jk}, f > &= \sum_n h_n \ < \phi_{j+1,2k+n}, f > \\ &= \sum_n h_{n-2k} \ < \phi_{j+1,n}, f >. \end{aligned}$$

Each results from convolving the sequence of higher-resolution inner products with the reversed sequences $\{g_n\}$ and $\{h_n\}$ and then retaining only the even numbered components of the convolution.

So given the coefficients $< \phi_{J,k}, f >$ for $k \in \boldsymbol{Z}$ at some fixed resolution level J, one can compute the coefficients at all coarser levels by successively filtering and decimating the sequences. This is the heart of the Discrete Wavelet Transform (DWT).

10. Cluster Analysis

In data mining, cluster analysis is often called **structure discovery**. Traditional data methods attempt to cluster all of the cases, but in data mining is is often sufficient to just find some of the clusters.

Recent applications include:

- market segmentation (e.g., Claritas, Inc.)
- syndromic surveillance
- text retrieval
- microarray data

Classical statistics invented three kinds of cluster analysis, many of which were independently reinvented by computer scientists. These three families are:

- hierarchical agglomerative clustering, which developed in the biological sciences and is the most widely used strategy;
- k-means clustering, which is an algorithmic technique invented by MacQueen (1967; *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, University of California Press, 281-297);
- mixture models, which were recently proposed by Banfield and Raftery (1993; *Biometrics*, **49**, 803-821) and are gaining wide currency.

Of these three methods mixture models make the strongest assumptions about the data being a random sample from a population, but this is also the method which best supports inference.

10.1 Hierarchical Clustering

Hierarchical agglomerative clustering joins cases together according to a fixed set of rules.

0. One starts with a sample $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ of cases and a metric d on all possible sets of cases (including the singleton sets).

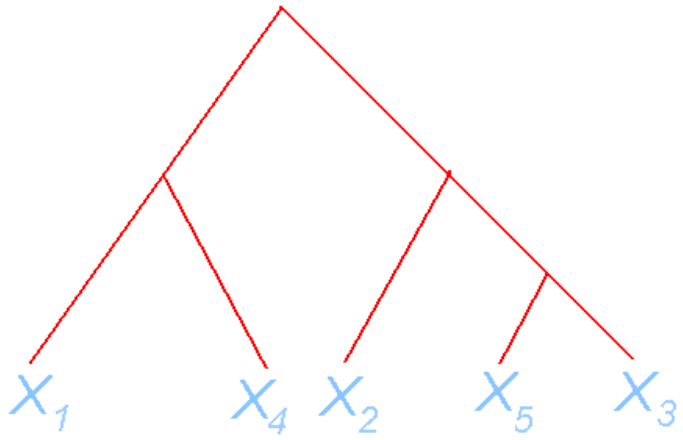
1. At the first step, one joins the two cases $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ that have the minimum distances among all possible pairs of cases.

2. At the second step, one joins either two singleton cases or a case $\boldsymbol{x}_k$ to $\{\boldsymbol{x}_i, \boldsymbol{x}_j\}$, according to whichever situation achieves minimum distance.

3. At subsequent steps one may be joining either pairs of cases, pairs of sets of cases, or a case to a set.

Sibson and Jardine (1971; *Mathematical Taxonomy*, Wiley) show that this approach uniquely satisfies certain theoretically desirable properties.

The result of a hierarchical agglomerative cluster analysis is often displayed as a tree.



The lengths of the edges in the binary tree shows the order in which cases or sets were joined together and the magnitude of the distance between them.

But it is hard to know when to stop growing a tree and declare the clusters. Milligan and Cooper (1985; *Psychometrika*, **50**, 159-179) like the cubic clustering criterion.

Some of the classic metrics for linking sets of cases are:

- Nearest-neighbor or single linkage. Here one has a metric d^* on $\mathbb{R}^p$ and defines

$$d(\mathcal{A}, \mathcal{B}) = \min_{a \in \mathcal{A}, b \in \mathcal{B}} d^*(a, b).$$

- Complete linkage. Here

$$d(\mathcal{A}, \mathcal{B}) = \max_{a \in \mathcal{A}, b \in \mathcal{B}} d^*(a, b).$$

- Centroid linkage. Here

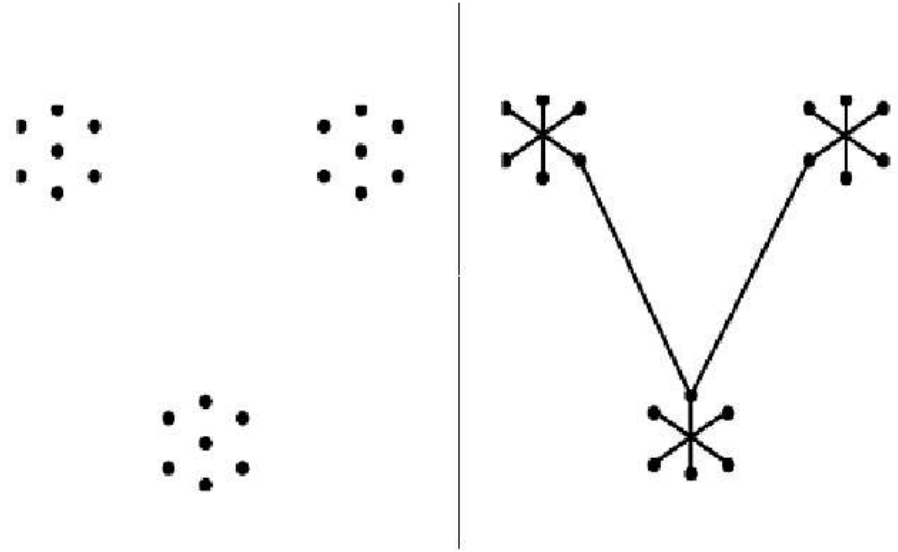
$$d(\mathcal{A}, \mathcal{B}) = d^*(\bar{a}, \bar{b})$$

  where $\bar{a}$ is the centroid (or average) of $\mathcal{A}$ and $\bar{b}$ is the centroid of $\mathcal{B}$.

- Many others have been used. Ward's metric looks at the ratio of between-set and within-set sums of squares, others use various kinds of weighting, etc.

Fisher and Van Ness (1971; *Biometrika*, **58**, 91-104) compare the properties of these metrics in a series of papers.

When the sample size is very large, and or p is large, then the fastest way to cluster cases uses single linkage and creates a minimal spanning tree. Then one just removes the longest edges to create clusters, as shown below.



There is an $\mathcal{O}(n^2)$ algorithm for this that was developed by Prim (1957; *Bell System Technical Journal*, **36**, 1389-1401). It can even be accelerated a little bit.

When p is large and most of the measurements are noise, then it is very difficult to discover true cluster structure. The signal that distinguishes the clusters gets lost in the chance variation among the spurious variables.

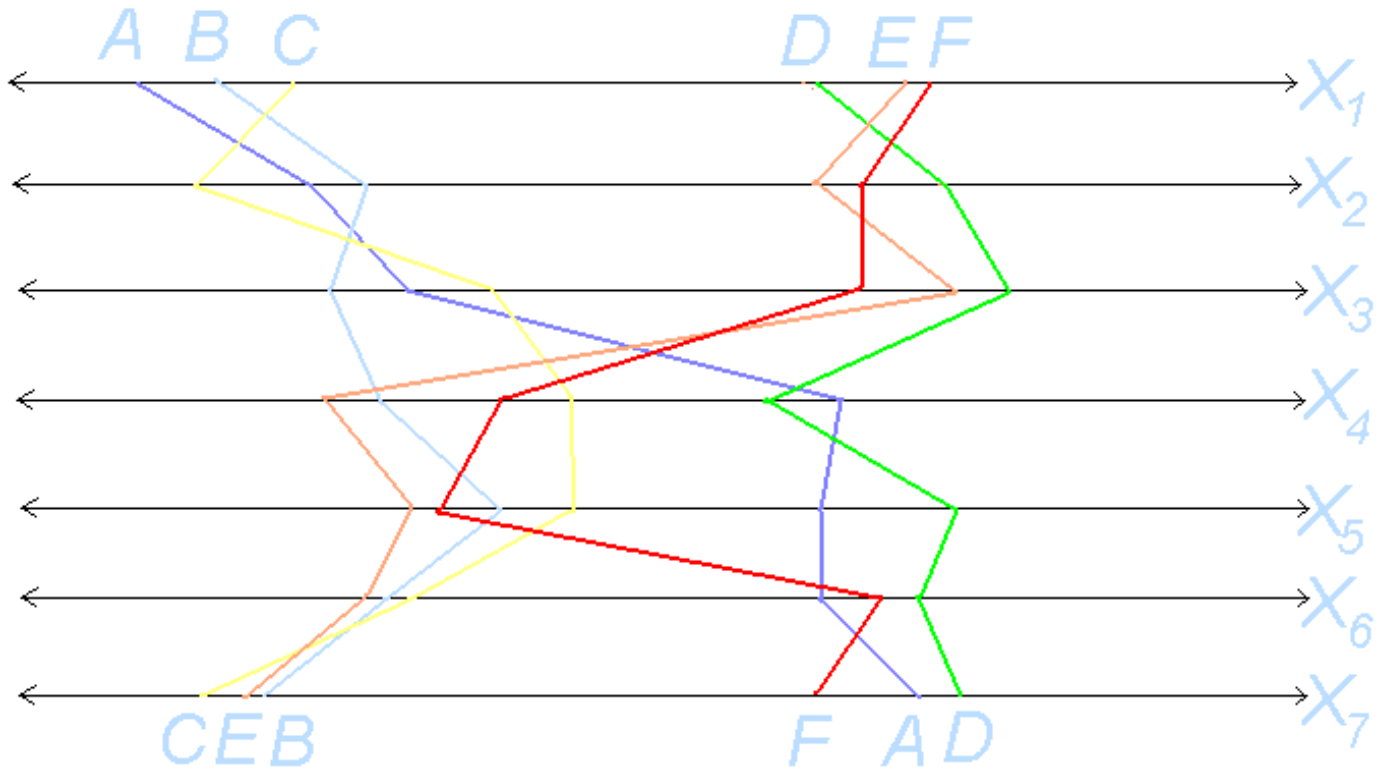
Graphical methods such as G-Gobi take too long to find interesting projections (via a Hermite polynomial function that measures "interestingness" according to the "gappiness" of the projection). See Swayne, Cook, and Buja (1998; *Journal of Computational and Graphical Statistics*, **7**, 113-130).

Visualization techniques probably become infeasible for $p > 10$ or so.

Another thing that can happen with large p is that there are multiple cluster structures. Here cases show strong clustering with respect to one subset of variables, and comparably strong, but distinct, clusters with respect to a different subset of the variables.

This situation gives rise to product structure in the clustering, which quickly becomes unmanageable.

Friedman and Meulman (2004, *Journal of the Royal Statistical Society, Series B*, to appear) discuss this problem and related issues. One strategy for visualizing multiple cluster structure is to use parallel coordinate plots, invented by Inselberg (1985; *The Visual Computer*, **1**, 69-91).



This shows that cases A, B, C and D, E, F have distinct cluster structure with respect to variables x_1, x_2, and x_3, while cases C, E, B and F, A, D Cluster on variables x_6 and x_7. There is no cluster structure on x_4 and x_5.

10.2 k-Means Clustering

In k-means clustering the analyst picks the number of clusters k and makes initial guesses about the cluster centers.

The algorithm starts at those k centers and absorbs nearby cases. Then it calculates a new cluster center (usually the mean) based upon the absorbed cases, and often a covariance matrix, and then absorbs more cases that are nearby, ususually in terms of Mahalanobis distance, to the current center:

$$d(\boldsymbol{x}_i, \bar{\boldsymbol{x}}_j) = [(\boldsymbol{x}_i - \bar{\boldsymbol{x}}_j)' \boldsymbol{S}^{-1} (\boldsymbol{x}_i - \bar{\boldsymbol{x}}_j)]^{1/2}$$

where $\boldsymbol{S}$ is the within-cluster covariance matrix and $\bar{\boldsymbol{x}}_j$ is the center of the current cluster j.

The process continues until all of the cases are assigned to one of the k clusters.

Smart computer scientists can do k-means clustering (or approximately this) very quickly. Andrew Moore presented methods at the NSA conference on streaming data in December 2002.

As in hierarchical agglomerative clustering, it is hard to know k. But one can do univariate search—try many values of k, and pick the one at the knee of some lack-of-fit curve, e.g., the ratio of the average within-cluster to between-cluster sum of squares.

The cluster centers move over time. If they move too much, this suggests that the clusters are unstable.

No unique solution is guaranteed for k-means clustering. In particular, if two starting points fall within the same true cluster, then it can be hard to discover new clusters.

10.2.1 Self-Organizing Maps

Kohonen (1989; *Self-Organization and Associative Memory*, Springer-Verlag) developed a procedure called Self-Organizing Maps or SOMs. They quickly become popular for visualizing complex data.

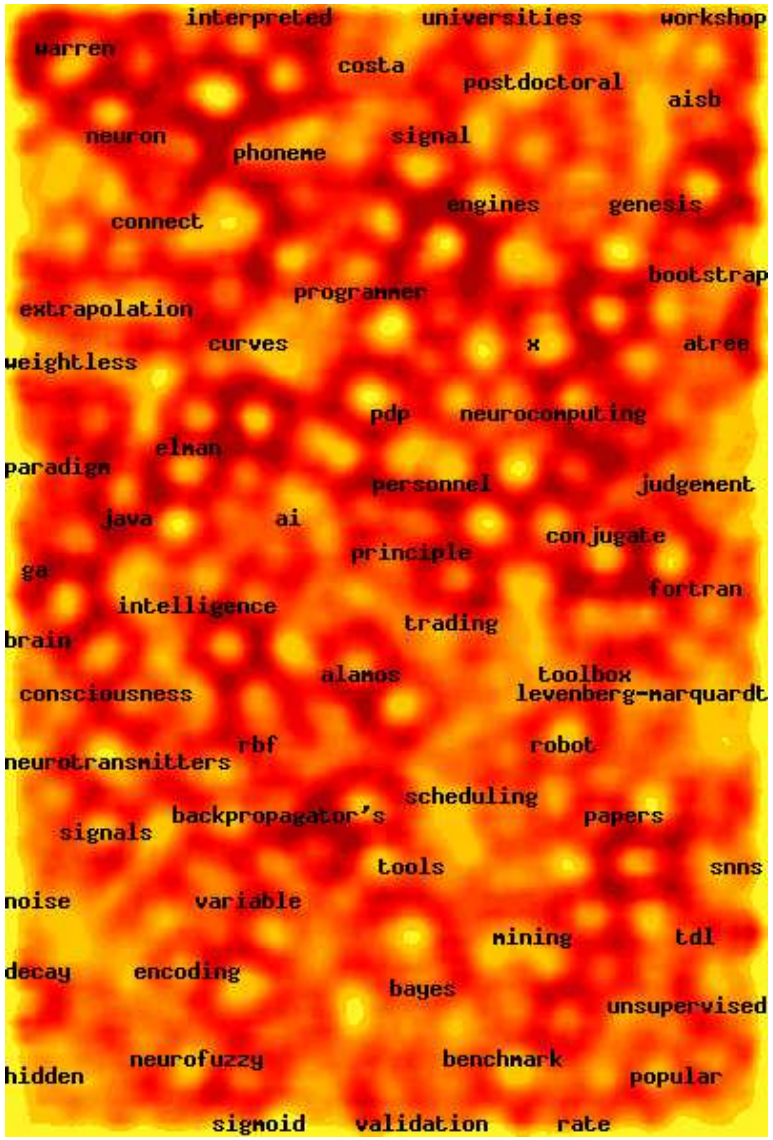
SOMs have become widely used in some aspects of business and information technology, but their underlying theory may be weak. There is no real way to specify uncertainty, and the method is highly susceptible to outliers.

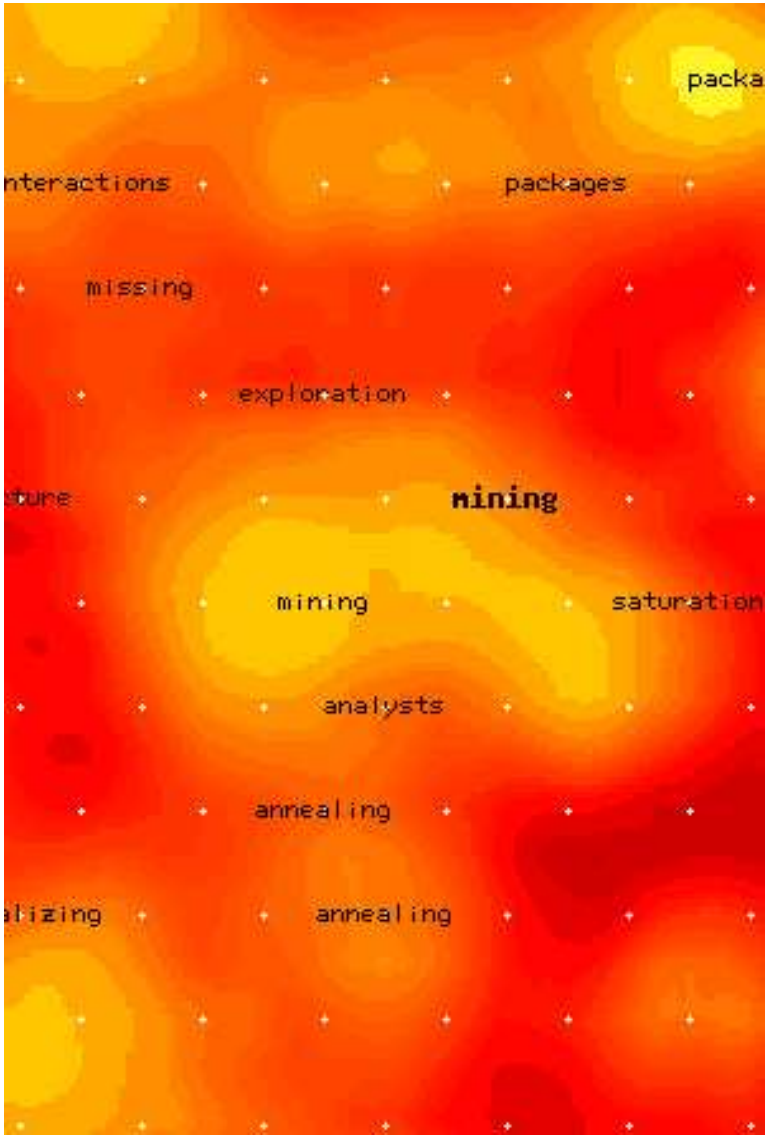
It turns out that SOMs can be viewed as k-means clustering with the constraint that the cluster centers have to lie in a plane.

SOMs are rather like multidimensional scaling. The intention is to produce a two-dimensional (sometimes three-dimensional) picture of high-dimensional data, one that puts similar observations close together in the visualization.

One starts with a set of prototypes, which are just the integer coordinates in the plane formed by the first two principal components of the data. This plane is then distorted, so as to pull the prototypes near to the data. Finally, the data are identified with their nearest point on the distorted surface.

The following two examples of SOMs are heatmaps showing the number of documents in the newsgroup corpus comp.ai.neural-nets with different kinds of subject matter, taken from the WEBSOM homepage. The first is the entire corpus (with well-populated nodes tagged with their keyword), and the second is a blow-up of the heatmap focused on the region around "mining".


interpreted
universities
workshop
warren
costa
postdoctoral
aisb
neuron
phoneme
signal
engines
genesis
connect
bootstrap
programmer
extrapolation
curves
x
atree
weightless
pdp
neuroconputing
elman
paradigm
personnel
judgement
java
ai
conjugate
principle
ga
fortran
intelligence
trading
brain
alamos
toolbox
consciousness
levenberg-marquardt
rbf
robot
neurotransmitters
scheduling
backpropagator's
papers
signals
tools
snns
noise
variable
mining
tdl
decay
encoding
bayes
unsupervised
neurofuzzy
benchmark
hidden
popular
sigmoid
validation
rate


packa
nteractions
packages
missing
exploration
ture
mining
mining
saturation
analysts
annealing
lizing
annealing

10.3 Mixture Models

Mixture models fit data by a weighted sum of pdfs:

$$f(\boldsymbol{x}) = \sum_{j=1}^{k} \pi_j g_j(\boldsymbol{x} \,|\, \boldsymbol{\theta}_j)$$

where the π_j are positive and sum to one and the g_j are density functions in a family indexed by $\boldsymbol{\theta}$.

The main technical difficulty with mixture models is identifiability. Both

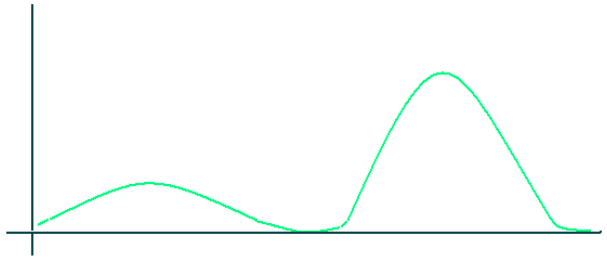
$$\textbf{Model 1} \quad : \quad \{\pi_1 = .5, g_1 = N(0,1);\ \pi_2 = .5, g_2 = N(0,1)\}$$
$$\textbf{Model 2} \quad : \quad \{\pi_1 = 1, g_1 = N(0,1);\ \pi_2 = 0, g_2 = N(-10,7)\}$$

describe exactly the same situation. But identifiability is only an issue at the "edges" of the model space—in most regions it is not a problem.

Bruce Lindsay (1995; *Mixture Models: Geometry, Theory, and Applications*, IMS) gives a convexity argument for solving the identifiability problem. But this has not yet worked its way into data mining practice.

Traditional mixture modeling ducks the identifiability question and uses the EM algorithm (cf. Dempster, Laird, and Rubin; 1977, *Journal of the Royal Statistical Society, Series B*, **39**, 1-22) for model fitting.

To illustrate this, consider how to fit a two-component Gaussian model to the following smoothed histogram.



The model for an observation is

$$f(x) = \pi\phi(x \,|\, \mu_1, \sigma_1^2) + (1-\pi)\phi_2(x \,|\, \mu_2, \sigma_2^2)$$

where ϕ_j is the normal density with parameters $\boldsymbol{\theta}_j = (\mu_j, \sigma_j^2)$.

Let $\boldsymbol{\theta} = (\pi, \mu_1, \mu_2, \sigma_1^2, \sigma_2^2)'$. A direct effort to maximize the log-likelihood leads to

$$\ell(\boldsymbol{\theta}, \boldsymbol{x}) = \sum_{i=1}^{n} \ln[\pi\phi_1(x_i \,|\, \mu_1, \sigma_1^2) + (1-\pi)\phi_2(x_i \,|\, \mu_2, \sigma_2^2)]$$

which turns out to be difficult to solve.

The EM algorithm alternates between an expectation step and a maximization step in order to converge on a solution for the maximum likelihood estimates in this problem.

Instead, assume that there are latent variables (unobserved variables) Δ_i that determine from which mixture component observation x_i came.

$$\Delta_i = \begin{cases} 0 & \textbf{if } x_i \sim \phi_2 \\ x_2^2 & \textbf{if } x_i \sim \phi_1 \end{cases}$$

Then we can write the following generative model:

$$\begin{aligned} Y_{1i} &= N(\mu_1, \sigma_1^2) \\ Y_{2i} &= N(\mu_2, \sigma_2^2) \\ X_i &= \Delta_i Y_{1i} + (1 - \Delta_i) Y_{2i} \end{aligned}$$

where $\mathbb{P}[\Delta_i = 1] = \pi$.

If we knew the $\{\Delta_i\}$ then we could get the mles for (μ_j, σ_j^2) separately for $j = 1, 2$, and this would be easy. But since we do not, we use the EM algorithm to obtain estimates by iterative alternation.

1. Make initial guesses for $\hat{\pi}$, $\hat{\mu}_1$, $\hat{\mu}_2$, $\hat{\sigma}_1^2$, and $\hat{\sigma}_2^2$. Usually we take $\hat{\pi} = .5$, the $\hat{\mu}_j$ are well-separated sample values, and the variance estimates are both set to the sample variance of the full dataset.

2. Expectation Step. Compute the expected values of the Δ_i terms (these are sometimes called "responsibilities"). The responsibity γ_i is:

$$\gamma_i = \frac{\hat{\pi}\phi_1(x_i \,|\, \hat{\mu}_1, \hat{\sigma}_1^2)}{\hat{\pi}\phi_1(x_i \,|\, \hat{\mu}_1, \hat{\sigma}_1^2) + (1-\hat{\pi})\phi_2(x_i \,|\, \hat{\mu}_2, \hat{\sigma}_2^2)}.$$

   Note that this is an approximation to the posterior probability the x_i comes from component ϕ_1.

3. Maximization Step. Compute the new estimates by weighting the sample:

$$\hat{\mu}_1 = \sum_{i=1}^n \gamma_i x_i / \sum_{i=1}^n \gamma_i \qquad \hat{\mu}_2 = \sum_{i=1}^n (1-\gamma_i) x_i / \sum_{i=1}^n (1-\gamma_i)$$
$$\hat{\sigma}_1^2 = \frac{\sum_{i=1}^n \gamma_i (x_i - \hat{\mu}_1)^2}{\sum_{i=1}^n \gamma_i} \qquad \hat{\sigma}_2^2 = \frac{\sum_{i=1}^n (1-\gamma_i)(x_i - \hat{\mu}_2)^2}{\sum_{i=1}^n (1-\gamma_i)}.$$

   Also, the new $\hat{\pi}$ is $n^{-1} \sum_{i=1}^n \gamma_i$.

4. Repeat steps 2 and 3 until convergence.

The EM algorithm climbs hills, so in general it converges to a local (but possibly not global) maximum. In part, it is fast because it increases the criterion function on *both* steps, not just the maximization step.

The EM algorithm does not solve the identifiability problem that can arise in mixture models. But it does find a local mle.

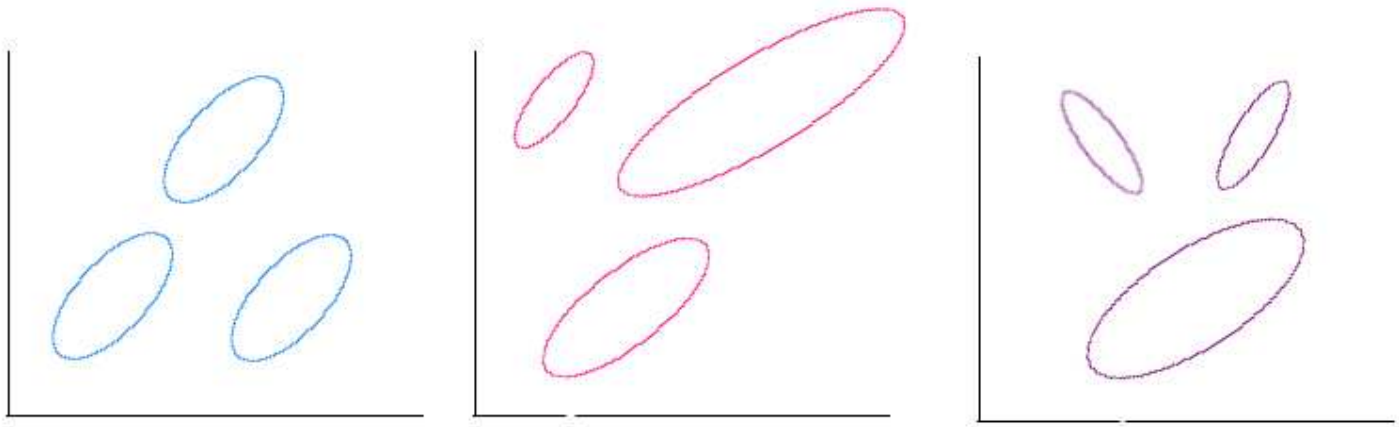
The EM algorithm is considerably more flexible than this example indicates, and it applies in a wide variety of cases (e.g., imputation in missing data problems, inference on causation through counterfactual data). The review given here follows the application in Hastie, Tibshirani, and Friedman (2001; *The Elements of Statistical Learning*, Wiley, chap. 8.5).

10.3.1 Model-Based Cluster Analysis

This is a parametric family of mixture models (usually Gaussian mixtures) that are becoming widely used in both cluster analysis and data mining.

Essentially, this uses a nested sequence of normal mixtures for cluster analysis. The nesting enables one to use likelihood ratio tests to determine which level of modeling is appropriate.

1. At the most nested level, the model assumes that the data come from a k-component mixture of normal distributions, each with a common covariance matrix but different means.
2. At the next-most nested level, the covariance matrices are allowed to differ by an unknown scalar multiple.
3. At the highest level, the covariance matrices for different components are completely unrelated.



The nesting problem does not avoid the problem of identifiability or choice of k. The EM algorithm is used for estimation.