Exercises 7: Latent-feature models

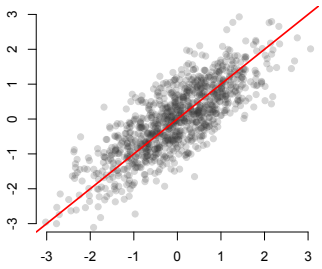
Projecting downward

A question one often encounters in statistics is: given a p-dimensional vector, how do we project it down into a smaller k-dimensional space in a way that preserves as much of the information in the original data as possible? The point is to represent a large amount of information in a tractable, more parsimonious way—in other words, to cut through the clutter.

You already know one way of doing this, namely linear regression. Given an n-dimensional outcome vector y and a matrix of covariates X, the fitted values $\hat{y} = X(X^T X)^{-1} X^T y$ (or their equivalents arising from a Bayesian model) involve a projection from $\mathcal{R}^n$ to $\mathcal{R}^p$. But what if you don't have regressors X, only the outcomes y?

A simple example: projection to $\mathcal{R}$

Say we have n observations of a p-dimensional outcome vector y_i. By Y, I mean the matrix whose ith row is the ith observation $y_i^T = (y_{i1}, \ldots, y_{ip})^T$. (Remember by convention that vectors are column vectors.) Suppose for the moment that every column of Y is standardized to have mean zero and unit variance.



Imagine projecting every observation y_i into a one-dimensional subspace—that is, defining a new scalar outcome $z_i = y_i^T w_1$ for some vector w_1. Presumably w_1 should be maximally information-preserving. The question is how to operationalize this rather loose idea.

Here's one way: choose w_1 so as to *maximize the variance* of the projected values z_i. The intuition for this is straightforward. In the picture at right, the points can be described fairly well by projecting each one onto the diagonal line and reporting the single number z_i. (Or equivalently, its actual position in p-dimensional space, which is $z_i w_1$—though this requires p numbers.) It's also easy to see that the projected points will have greater variance in this subspace than they would in any other choice of subspace. Try drawing some other line through the point cloud; you'll see that the projections of the points onto this line would be more scrunched up than along the line I've drawn.

Mathematically, this means choosing w_1 such that the projected variance

$$V_w = \frac{1}{n} \sum_{i=1}^{n} (z_i - \bar{z})^2 = \frac{1}{n} \sum_{i=1}^{n} (y_i^T w_1 - \bar{z})^2$$

is as large as possible. Of course, we can blow up the variance to be as

large as we want by choosing w_1 itself to be huge, so we must constrain it somehow. A natural constraint is that w_1 is a unit vector: $w_1^T w_1 = 1$.

1. Characterize the relationship between the singular value decomposition of Y and the eigenvalue decomposition of $\frac{1}{n} Y^T Y$.
2. Prove[1] that the unit-length w_1 which maximizes the projection variance is a left-singular vector of the data matrix Y corresponding to the largest singular value d_1. What is the relationship between V_w and d_1?
3. Load the data in "congress109.csv." The rows are members of the 109th U.S. Congress; the columns are phrases uttered during floor speeches. Entry (i, j) in the matrix is the number of times member i uttered phrase j. Find the variance-maximizing one-dimensional projection, and compute the location of each member in this one-dimensional space. (Meet R's built-in routines `svd` and `eigen`.) You've now moved from 1000 pieces of information about each member, to 1. Consult the information in "congress109members.csv." (You might find R's `merge` command helpful.) Does location in the subspace you've defined seem to correlate with relevant political facts about each member?
4. Since each projected value is $z_i = y_i^T w_1$, we can write the whole column vector of z_i's as $Z = Y w_1$, and the residuals from this projection as $R = Y - Z w_1^T$. Each row of R is the residual vector for the ith case, after the projection.

   Now imagine applying the same procedure as above to the residuals: that is, finding the maximum-variance projection of each residual vector, defined by some new vector w_2. Prove that w_2 is the singular vector of the original data matrix Y corresponding to the second largest singular value d_2.

[1] Remember that the method of Lagrange multipliers is useful for optimizing under constraints.