The Truth about Principal Components and Factor Analysis

36-350, Data Mining

3 October 2008

1 The Truth about Principal Components Analysis

Principal components tries to re-express the data as a sum of uncorrelated components. There are lots of other techniques which try to do similar things, like Fourier analysis, or wavelet decomposition. Things like Fourier analysis decompose the data into a sum of a *fixed* set of basis functions or basis vectors. This has the advantage of making results comparable across data sets, and of making the meaning of the components clear. So why ever do PCA rather than a Fourier transform?

First, in some situations the idea of doing a Fourier transform is just embarrassingly weird. For the states or cars data ets, we could number the features and take cosines of the feature numbers, etc., but it just seems crazy. No such embarrassment attends PCA. Second, when using a fixed set of components, there is no guarantee that a small number of components will give a good reconstruction of the original data. PCA guarantees that the first k components will do a better (mean-square) job of reconstructing the original data than any other linear method using only k components. Third, it is good at preserving distances between the points — the component scores give the optimal linear multidimensional scaling (see section 3.7 of *Principles of Data Mining*).

PCA gives us uncorrelated components, which are generally not independent components; for that you need independent component analysis (Stone, 2004). PCA looks for *linear* combinations of the original features; one could well do better by finding nonlinear combinations. Rather than directions in feature space, these would be curves or surfaces.

PCA is purely a descriptive technique; in itself it makes no prediction about what future data will look like.

1.1 Convergence

If the data come from IID samples of a distribution with covariance matrix $\mathbf{U}$, then the sample covariance matrix $\mathbf{V} \equiv \frac{1}{n}\mathbf{X}^T\mathbf{X}$ will converge on $\mathbf{U}$ as $n \to \infty$.

Since the principal components are functions of $\mathbf{V}$ (namely its eigenvectors), they will tend to converge as n grows[1]. So, along with that additional assumption about the data-generating process, PCA does make a prediction: that future PCA results will look like the present ones.

1.2 Simulating with PCA

One can also try to turn PCA into a model which makes predictions about future data vectors more directly. The observed features are re-written in terms not just of the PCs but also of the projections along those PCs. One could try replacing those projection scores with random numbers and then transforming back into features to get new, simulated feature vectors. That is, PCA writes $\mathbf{H} = \mathbf{X}\mathbf{w}$, so that $\mathbf{H}\mathbf{w}^{-1} = \mathbf{X}$. (This holds exactly if we use the full set of all p principal components.) Replace the component scores in $\mathbf{H}$ with similar but random numbers, say $\mathbf{J}$, and one will get a new set of random feature vectors, $\mathbf{Y} = \mathbf{J}\mathbf{w}^{-1}$. We could get $\mathbf{J}$ either by fitting some distribution to $\mathbf{H}$, or, less parametrically, by re-sampling the latter's columns.[2] (Exercise: What will the covariance matrix of $\mathbf{Y}$ be?) This kind of approach is sometimes used to create synthetic data for testing other algorithms, or to check whether the combination of components resembles the original in more qualtitative ways than just mean squared error.[3]

[1] There is a wrinkle if U has "degenerate" eigenvalues, i.e., two or more eigenvectors with the same eigenvalue. Then any linear combination of those vectors is also an eigenvector, with the same eigenvalue. (Exercise: show this.) For instance, if $\mathbf{U}$ is the identity matrix, then every vector is an eigenvector, and PCA routines will return an essentially arbitrary collection of mutually perpendicular vectors. Generically, however, any arbitrarily small tweak to $\mathbf{U}$ will break the degeneracy.

[2] That is, to generate a new value for the jth principal component, one just draws uniformly from the jth column in $\mathbf{H}$. We will see more of this kind of thing later, when we consider bootstrapping.

[3] Brian Whitman's Eigenradio (`eigenradio.media.mit.edu`) would do something like this in real time to a couple of radio stations. Occasionally it would even sound like human music. The site is offline now, but see `http://www.bagatellen.com/archives/interviews/000974.html` for an interview where Whitman tries to explain it to a music blog.

2 The Truth about Factor Analysis

Recall the factor-analysis model:

$$\mathbf{X} = \mathbf{F}\mathbf{w} + \epsilon$$

The factor-score matrix $\mathbf{F}$ is smaller than the data matrix $\mathbf{X}$ ($n \times k$ versus $n \times p$), but $\mathbf{F}\mathbf{w}$ has nearly the same correlations as the original features. If we want to eliminate some dimensions while preserving correlations, then the factor scores are a good summary of the data.

Many people also think of the factor model as a generative model, an account of how the data were produced in the first place. Seen this way, it is also a predictive model. Its prediction is that

$$X \sim \mathcal{N}(0, \mathbf{w}^T\mathbf{w} + \Psi) \tag{1}$$

Of course it might seem like it makes a more refined prediction,

$$X|F \sim \mathcal{N}(F\mathbf{w}, \Psi) \tag{2}$$

but the problem is that there is no way to guess at or estimate F until after we've seen X, at which point anyone can predict X perfectly. So the actual *forecast* is given by Eq. 1.[4]

Now, without going through the trouble of factor analysis, one could always just postulate that

$$X \sim \mathcal{N}(0, U) \tag{3}$$

and estimate U; the maximum likelihood estimate of it is the observed covariance matrix. The closer our estimate $\widehat{U}$ is to U, the better our predictions. One way to think of factor analysis is that it looks for the maximum likelihood estimate, but constrained to matrices of the form $\mathbf{w}^T\mathbf{w} + \Psi$.

On the plus side, the constrained estimate has a faster rate of convergence. That is, both the constrained and unconstrained estimates are consistent and will converge on their optimal, population values as we feed in more and more data, but for the same amount of data the constrained estimate is probably closer to its limiting value. In other words, the constrained estimate $\widehat{\mathbf{w}}^T\widehat{\mathbf{w}} + \widehat{\Psi}$ has less variance than the unconstrained estimate $\widehat{U}$.

On the minus side, maybe the true, population U just can't be written in the form $\mathbf{w}^T\mathbf{w} + \Psi$. Then we're getting biased estimates of the covariance and the bias will *not* go away, even with infinitely many samples. Using factor analysis rather than just fitting a multivariate Gaussian means betting that either this bias is really zero, or that, with the amount of data on hand, the reduction in variance outweighs the bias.

[4] A subtlety is that we might get to see some but not all of X, and use that to predict the rest. Say $X = (X_1, X_2)$, and we see X_1. Then we could, in principle, compute the conditional distribution of the factors, $p(F|X_1)$, and use that to predict X_2. Of course one could do the same thing using the correlation matrix, factor model or no factor model.

(I haven't talked about estimated errors in the parameters of a factor model. The easiest way to obtain these is through either the jack-knife method — leave each observation out, re-estimate the model, and look at the distribution of re-estimates around the full-data estimate — or the bootstrap method — randomly re-sample the data, re-estimate, and again look at the distribution around the full-data estimate. We'll see much more about both of these methods after the midterm, so just bear in mind that if you need to use factor analysis you can do this.)

2.1 The Graphical Form of Factor Models

One can represent the factor model as a graph like Figure 1. The square nodes stand for the features, which are observable, and the circles for the factors, which are not directly observable. The numbers beside the arrows are the factor loadings, taken from the matrix $\mathbf{w}$. When the loading of a feature on a factor is zero, draw no arrow. Thus $X_b = -0.75F_1 + 0.34F_2 + \epsilon_b$. The correlations between variables can be worked out from the arrows: X_a and X_b have only the factor F_1 in common, so their correlation is $(0.87)(-0.75) = -0.65$. On the other hand, X_c and X_d have two factors in common, and so their correlation is $(0.13)(0.20) + (0.73)(0.10) = 0.099$.

X_b and X_c are conditionally independent, given F_2, because that is their only common factor. On the other hand, F_1 and F_2 are conditionally *dependent* given X_b, because knowing X_b tells us something about the value of $-0.75F_1 + 0.34F_2$, and so about F_1 and F_2. We will see later that there is a whole set of rules for deducing conditional independence relations from diagrams like this. This is because factor models are a special case of the broader class of **graphical models**, specifically a variety of **linear Gaussian graphical model**.

A natural impulse, when looking at something like Figure 1, is to reify the factors and to treat the arrows **causally**: that is, to say that there really is *some* variable corresponding to each factor, and that changing the value of that variable will change the features. For instance, one might want to say that there is a real, physical variable corresponding to the factor F_1, and that increasing this by one standard deviation will, on average, increase X_a by 0.87 standard deviations, decrease X_b by 0.75 standard deviations, and do nothing to the other features. Moreover, changing any of the other factors has no effect on X_a.

Sometimes all this is even right. How can we tell when it's right?

2.2 The Rotation Problem Again

Consider the following matrix, call it R:

$$\begin{bmatrix} \cos 30 & -\sin 30 & 0 \\ \sin 30 & \cos 30 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Applied to a three-dimensional vector, this rotates it thirty degrees counter-clockwise around the vertical axis. If we apply R to the factor loading matrix

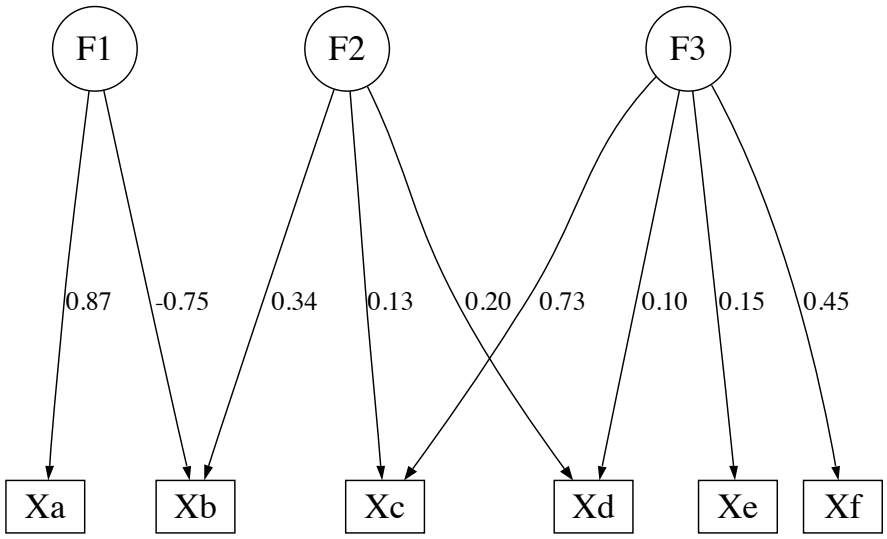


Figure 1: Graphical model form of a factor model. Circles stand for the unobserved factors, boxes for the observed features. Edges indicate non-zero entries in the factor loading matrix.

of the model in the figure, we get the model in Figure 2. Now instead of X_a being correlated with the other variables only through one factor, it's correlated through two factors, and X_d has incoming arrows from three factors.

Because the transformation is orthogonal, the new factors are still uncorrelated with each other, and the distribution of the observations is unchanged. In particular, the fit of the new factor model to the data will be *exactly* as good as the fit of the old model. If we try to take this causally, however, we come up with a very different interpretation. The quaity of the fit to the data does not, therefore, let us distinguish between these two models, and so these two stories about the causal structure of the data.

The rotation problem does not rule out the idea that checking the fit of a factor model would let us discover *how many* hidden causal variables there are.

2.3 Factors or Mixtures?

Suppose we have two distributions with probability densities $f_0(x)$ and $f_1(x)$. Then we can define a new distribution which is a **mixture** of them, with density $f_\alpha(x) = (1-\alpha)f_0(x) + \alpha f_1(x)$, $0 \leq \alpha \leq 1$. The same idea works if we combine

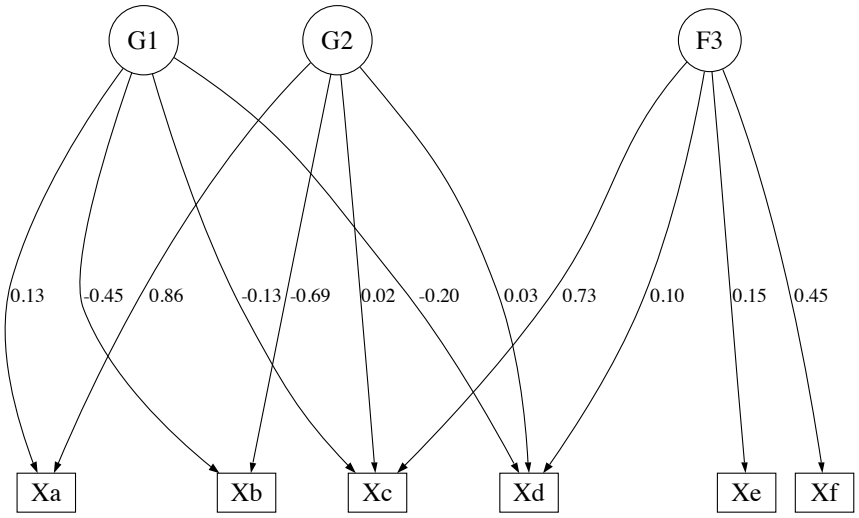


Figure 2: The model from Figure 1, after rotating the first two factors by 30 degrees around the third factor's axis. The new factor loadings are rounded to two decimal places.

more than two distributions, so long as the sum of the **mixing weights** sum to one (as do α and $1-\alpha$). We will look more later at mixture models, which provide a very flexible and useful way of representing complicated probability distributions. They are also a probabilistic, predictive alternative to the kind of clustering techniques we've seen before this: each distribution in the mixture is basically a cluster, and the mixing weights are the probabilities of drawing a new sample from the different clusters.

I bring up mixture models here because there is a very remarkable result: any linear, Gaussian factor model with k factors is equivalent to some mixture model with $k+1$ clusters, in the sense that the two models have the same means and covariances (Bartholomew, 1987, pp. 36–38). Recall from Lecture 13 that the likelihood of a factor model depends on the data only through the correlation matrix. If the data really were generated by sampling from $k+1$ clusters, then a model with k factors can match the covariance matrix very well, and so get a very high likelihood. This means it will, by the usual test, seem like a very good fit. Needless to say, however, the causal interpretations of the mixture model and the factor model are very different. The two *may* be distinguishable if the clusters are well-separated (by looking to see whether the data are unimodal or

not), but that's not exactly guaranteed.

All of which suggests that factor analysis can't really tell us whether we have k continuous hidden causal variables, or one discrete hidden variable taking $k+1$ values.

2.4 The Thomson Sampling Model

We have been working with fewer factors than we have features. Suppose that's not true. Suppose that each of our features is actually a linear combination of a *lot* of variables we don't measure:

$$X_i = \eta_i + \sum_{j=1}^{q} A_j T_{ji} = \eta_i + A \cdot T_i \tag{4}$$

where $q \gg p$. Suppose further that the latent variables A_j are totally independent of one another, but they all have mean 0 and variance 1; and that the noises η_i are independent of each other and of the A_j, with variance ϕ_i. What then is the covariance between X_a and X_b? Well, because $\mathbf{E}[X_a] = \mathbf{E}[X_b] = 0$, it will just be the expectation of the product of the features:

$$\begin{aligned}
\mathbf{E}[X_a X_b] &= \mathbf{E}[(\eta_a + A \cdot T_a)(\eta_b + A \cdot T_b)] && (5)\\
&= \mathbf{E}[\eta_a \eta_b] + \mathbf{E}[\eta_a A \cdot T_b] + \mathbf{E}[\eta_b A \cdot T_a] + \mathbf{E}[(A \cdot T_a)(A \cdot T_b)] && (6)\\
&= 0 + 0 + 0 + \mathbf{E}\left[\left(\sum_{j=1}^{q} A_j T_{ja}\right)\left(\sum_{j'=1}^{q} A_{j'} T_{jb}\right)\right] && (7)\\
&= \mathbf{E}\left[\sum_{j,j'} A_j A_{j'} T_{ja} T_{j'b}\right] && (8)\\
&= \sum_{j,j'} \mathbf{E}[A_j A_{j'} T_{ja} T_{j'b}] && (9)\\
&= \sum_{j,j'} \mathbf{E}[A_j A_{j'}]\,\mathbf{E}[T_{ja} T_{j'b}] && (10)\\
&= \sum_{j=1}^{q} \mathbf{E}[T_{ja} T_{jb}] && (11)
\end{aligned}$$

where to get the last line I use the fact that $\mathbf{E}[A_j A_{j'}] = 1$ if $j = j'$ and $= 0$ otherwise. If the coefficients T are fixed, then the last expectation goes away and we merely have the same kind of sum we've seen before, in the factor model.

Instead, however, let's say that the coefficients T are themselves random (but independent of the A_j and η_j). For each feature X_a, we fix a proportion z_a between 0 and 1. We then $T_{ja} \sim \text{Bernoulli}(z_a)$, with $T_{ja} \perp T_{j'b}$ unless $j = j'$ and $a = b$. Then

$$\mathbf{E}[T_{ja} T_{jb}] = \mathbf{E}[T_{ja}]\,\mathbf{E}[T_{jb}] = z_a z_b$$

and

$$\mathbf{E}\left[X_a X_b\right] = q z_a z_b$$

which is *exactly* what it should be in the one-factor model, according to what we saw in Lecture 12.

Now, it doesn't make a lot of sense to imagine that every time we make an observation we change the coefficients T randomly. Instead, let's suppose that they are first generated randomly, giving values T_{ji}, and then we generate feature values according to Eq. 4. The covariance between X_a and X_b will be $\sum_{j=1}^{q} T_{ja} T_{jb}$. But this is a sum of IID random values, so by the law of large numbers as q gets large this will become very close to $q z_a z_b$. Thus, for nearly all choices of the coefficients, the feature covariance matrix should come very close to satisfying the tetrad equations and looking like there's a single general factor.

In this model, each feature is a linear combination of a *random sample* of a huge pool of completely independent features, plus some extra noise specific to the feature.[5] Precisely *because* of this, the features are correlated, and the pattern of correlations is that of a factor model with one factor. The appearance of a single common cause actually arises from the fact that the number of causes is immense, and there is no particular pattern to their influence on the features.

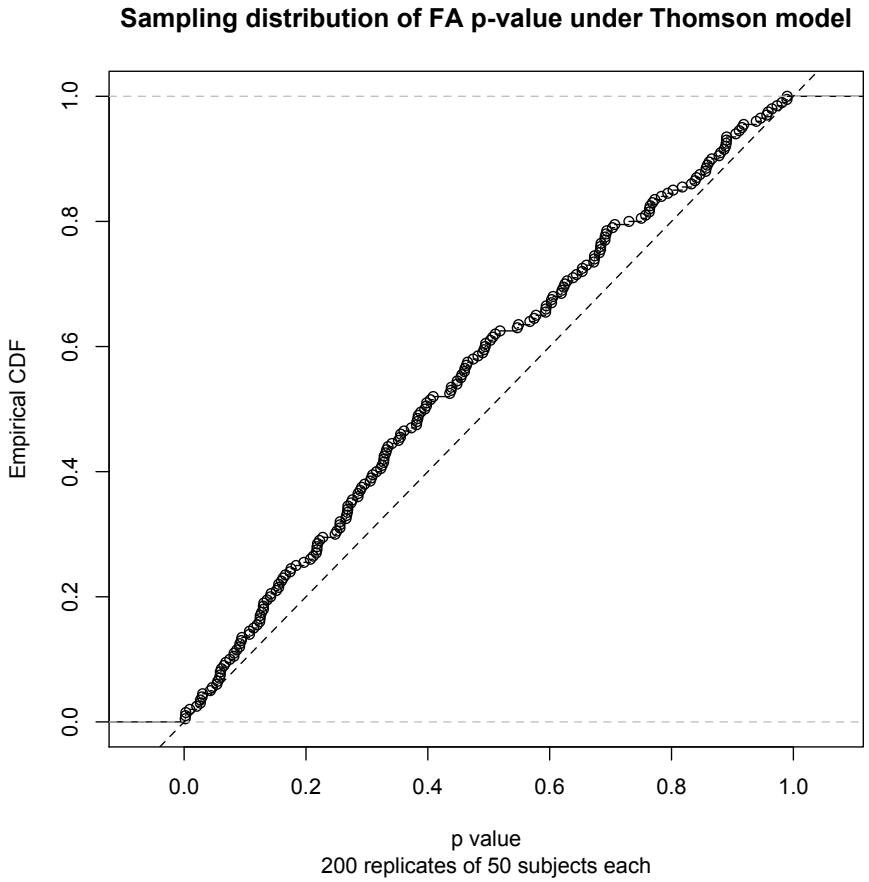
The file `thomson-model.R` (on Blackboard) simulates the Thomson model.

```
> tm = rthomson(50,11,500,50)
> factanal(tm$data,1)
```

The first command generates data from $n = 50$ items with $p = 11$ features and $q = 500$ latent variables. (The last argument controls the average size of the specific variances ϕ_j.) The result of the factor analysis is of course variable, depending on the random draws; my first attempt gave the proportion of variance associated with the factor as 0.391, and the p-value as 0.527. Repeating the simulation many times, one sees that the p-value is pretty close to uniformly distributed, which is what it should be if the null hypothesis is true (Figure 3). For fixed n, the distribution becomes closer to uniform the larger we make q. In other words, the goodness-of-fit test has little or no power against the alternative of the Thomson model.

Modifying the Thomson model to look like multiple factors grows notationally cumbersome; the basic idea however is to use multiple pools of independently-

[5] When Godfrey Thomson introduced this model in 1914, he used a slightly different procedure to generate the coefficient T_{ji}. For each feature he drew a uniform integer between 1 and q, call it q_i, and then sampled the integers from 1 to q *without replacement* until he had q_i random numbers; these were the values of j where $T_{ji} = 1$. This is basically similar to what I describe, setting $z_i = q_i/q$, but a bit harder to analyze in an elementary way. — Thomson (1916), the original paper, includes what we would now call a simulation study of the model, where Thomson stepped through the procedure to produce simulated data, calculate the empirical correlation matrix of the features, and check the fit to the tetrad equations. Not having a computer, Thomson generated the values of T_{ji} with a deck of cards, and of the A_i and η_i by rolling *5220* dice.




```
> plot(ecdf(replicate(200,factanal(rthomson(50,11,500,50)$data,1)$PVAL)),
       xlab="p value",ylab="Empirical CDF",
       main="Sampling distribution of FA p-value under Thomson model",
       sub="200 replicates of 50 subjects each")
> abline(0,1,lty=2)
```

Figure 3: Mimcry of the one-factor model by the Thomson model. The Thomson model was simulated 200 times with the parameters given above; each time, the simulated data was then fit to a factor model with one factor, and the p-value of the goodness-of-fit test extracted. The plot shows the empirical cumulative distribution function of the p-values. If the null hypothesis were exactly true, then $p \sim \text{Unif}(0, 1)$, and the theoretical CDF would be the diagonal line (dashed).

sampled latent variables, and sum them:

$$X_i = \eta_i + \sum_{j=1}^{q_1} A_j T_{ji} + \sum_{j=1}^{q_2} B_j R_{ji} + \ldots$$

where the T_{ji} coefficients are independent of the R_{ji}, and so forth.

It's not feasible to estimate the T_{ji} of the Thomson model in the same way that we estimate factor loadings, because $q > p$. This is not the point of considering the model, which is rather to make it clear that we actually learn very little about where the data come from when we learn that a factor model fits well. It could mean that the features arise from combining a small number of factors, or on the contrary from combining a huge number of factors in a random fashion. A lot of the time the latter is a more plausible-sounding story.[6]

For example, a common application of factor analysis is in marketing: you survey consumers and ask them to rate a bunch of products on a range of features, and then do factor analysis to find attributes which summarize the features. That's fine, but it may well be that each of the features is influenced by lots of aspects of the product you don't include in your survey, and the correlations are really explained by different features being affected by many of the same small aspects of the product. Similarly for psychological testing: answering any question is really a pretty complicated process involving lots of small processes and skills (of perception, several kinds of memory, problem-solving, attention, etc.), which overlap partially from question to question.

[6] Thomson (1939) remains one of the most insightful books on factor analysis, though obviously there have been a lot of technical refinements since he wrote. It's strongly recommended for anyone who plans to make much use of factor analysis. While out of print, used copies are reasonably plentiful and cheap.

3 Advice

Principal components is a pretty good thing to try if you need or want to do dimension reduction but aren't sure what exactly to use. It's got some reasonable mathematical properties, can often be interpreted, and runs fast (comparatively speaking).

Factor analysis does not offer any general advantages over PCA when it comes to data reduction, except for preserving correlations. One or the other of them may work better, depending on your data and what you want to do with it. Factor analysis can also be used as a predictive model. This is possible because it fits a distribution to the data, and not because it actually gets at the underlying causal structure with any reliability or power.

In both cases, the dimensions found by PCA and FA may be real features of the data, or they may just be more-or-less convenient fictions and summaries. That they are real is a hypothesis which these methods can suggest but for which they can provide only very weak evidence. This matters because ultimately we do data mining to discover knowledge on which we can *act*. It's one thing if our action is just a prediction to help us adjust to the world, but it's another if we go out and try to change the world based on how we think the different parts of it depend on each other. To do that well, we need to know what those parts really are.

References

Bartholomew, David J. (1987). *Latent Variable Models and Factor Analysis*. New York: Oxford University Press.

Stone, James V. (2004). *Independent Component Analysis: A Tutorial Introduction*. Cambridge, Massachusetts: MIT Press.

Thomson, Godfrey H. (1916). "A Hierarchy without a General Factor." *British Journal of Psychology*, **8**: 271–281.

— (1939). *The Factorial Analysis of Human Ability*. Boston: Houghton Mifflin Company.