CPC/back-projection error propagation

Ben Bolker

April 8, 2011

This vignette is a technical overview: see the `cpc-intro` vignette for a more basic introduction to how to use the package.

The problem here is to figure out how the error in estimating the common principal components of a set of data from multiple treatments propagates, and should be considered, when testing the differences between groups.

Specifically: suppose we have two groups of individuals (e.g. prey exposed to predators and prey not exposed to predators) and a (multivariate) set of morphological measurements on each individual. We assume that there is some underlying allometry by which individuals that change in size will also change in shape as a result (assume some appropriate transformations, i.e. log transformation of all traits). We aim to separate out changes in *shape* caused by phenotypic plasticity from changes that are simply due to changes in size.

We'll do this by calculating common principal components (CPC) for within-group variation, back-projecting to eliminate the effects of the first CPC, and doing univariate or multivariate analyses of the resulting size-standardized traits separated by group. A number of assumptions we'll make here are (1) within-group allometric variation in size-related traits is a good proxy for between-group variation; (2) the first CPC characterizes effects of size (e.g. it would help our case here if the first CPC had positive loadings for all traits); (3) it makes sense to remove the first CPC even if only the two variance-covariance matrices only have one PC in common. Generally, we'll use Phillips' program to calculate CPC even though in the special case of equal variance-covariance matrices we should get very similar (but not identical [??]) results by subtracting the within-group means of all traits from each trait, pooling the data, and calculating ordinary principal components (which is what I'll do here since I'm simply doing examples with known equal underlying VC matrices).

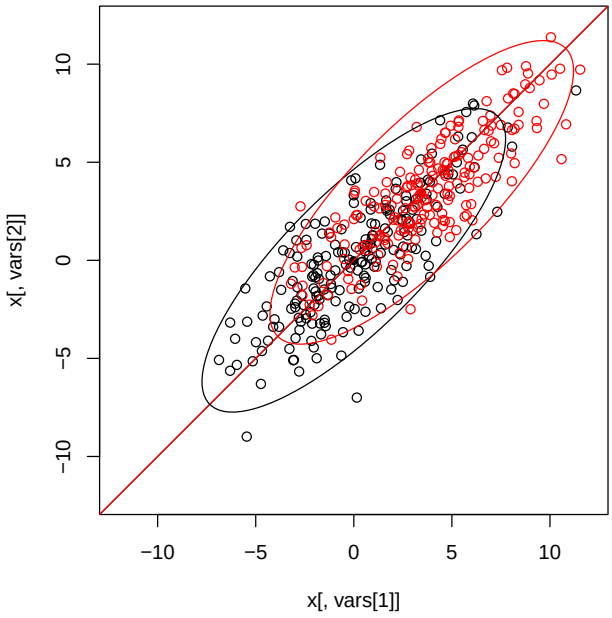
Some of these concepts make more sense with > 2 traits (e.g. some but not all PC in common), but I'm going to illustrate with two-trait examples for simplicity.

```
> library(MASS)
> library(ellipse)
> library(cpcbp)
```

Now plot some pictures. I'm going to draw this twice, once with automatically scaled axes and once with equal-scaled axes, because automatically scaled axes give a quite misleading picture of the actual geometry ...

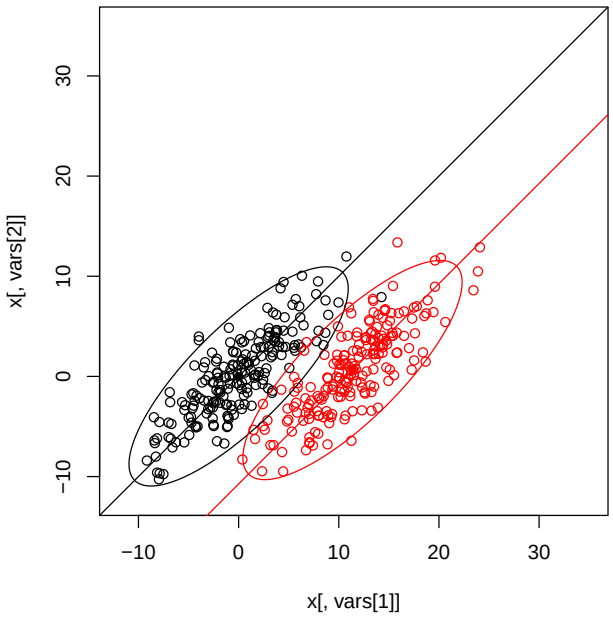
Our first example is a null case: an offset along the first common principal component (size axis) only (only variables 1 and 2 are shown; we can also use `plot_multigrp` to plot all pairs).

```
> set.seed(1001)
> X1 = simdata(offset = 6)
> T1 = sim.theor(offset = 6)
> op = par(pty = "s")
> plot_dat.theor(X1, vars = 1:2, xlim = c(-12, 12), ylim = c(-12,
+     12), theor = T1)
> par(op)
```



Or alternatively with a shape change — an offset perpendicular to the size axis:

```
> X2 = simdata(offset = 15, offset2 = 10, vars = c(20, 20, 20))
> T2 = sim.theor(offset = 15, offset2 = 10, vars = c(20, 20, 20))
> op = par(pty = "s")
> plot_dat.theor(X2, vars = 1:2, xlim = c(-12, 35), ylim = c(-12,
+     35), theor = T2)
> par(op)
```



The back-projection equation is:

$$\mathbf{x}_b = (\mathbf{1} - \boldsymbol{\beta}_1 \boldsymbol{\beta}_1^T)\mathbf{x}, \tag{1}$$

where $\mathbf{x}$ is a data vector (measurements of all traits for a single individual); $\boldsymbol{\beta}_1$ is the first principal direction (eigenvector), scaled so that $\boldsymbol{\beta}_1 \cdot \boldsymbol{\beta}_1 = 1$. To understand this formula, think about breaking up $(\boldsymbol{\beta}_1 \boldsymbol{\beta}_1^T)\mathbf{x}$. The first multiplication $(\boldsymbol{\beta}_1^T \mathbf{x})$ projects $\mathbf{x}$ onto the first principal direction (calculating a scalar that is the score for the first principal component); the second (multiplying by $\boldsymbol{\beta}_1$) translates this score back into the original coordinate system.

As we now know, the error in calculating the back-projection matrix enters (or should enter) into the estimate of the error in the differences between groups. How do we account for this extra error? In principle, we know how to compute the errors on any element of the eigenvector matrix (see [1], p. 82–83). Specifically, (4.6) of [1] tells us

$$\hat{\theta}_{jh}^{(i)} = r_i^{-1} \frac{\hat{\lambda}_{ij}\hat{\lambda}_{ih}}{(\hat{\lambda}_{ij} - \hat{\lambda}_{ih})^2}, \tag{2}$$

where $r_i = n_i/n$ (fraction of total data points in group i) and $\hat{\lambda}_{ij}$ is the estimate of the jth eigenvalue of group i's variance-covariance matrix. Given $\hat{\theta}_{jh}^{(i)}$ we can

calculate a harmonic mean

$$\hat{\theta}_{jh} = \left(\sum_{i=1}^{k} \left(\hat{\theta}_{jh}^{(i)} \right)^{-1} \right)^{-1} \tag{3}$$

and find the large-sample estimate of the standard error of $\hat{\beta}_{mh}$ to be

$$s(\hat{\beta}_{mh}) = \left(\frac{1}{n} \sum_{j=1, j \neq h}^{p} \hat{\theta}_{jh} \hat{\beta}_{mj}^{2} \right)^{1/2}, \tag{4}$$

or

$$\Sigma^2(\hat{\beta}) = \frac{1}{n} \left(\hat{\beta}^2 \right)^T \cdot \Theta, \tag{5}$$

where Θ is θ_{ij} as above for $i \neq j$, 0 on the diagonal.

More generally (see (2.5) from [1]) we have that the variance-covariance matrix of the elements in $\boldsymbol{\beta}_1$ is:

$$\frac{1}{n} \sum_{j=1, j \neq h}^{p} \hat{\theta}_{1h} \boldsymbol{\beta}_h \boldsymbol{\beta}_h' \tag{6}$$

(eq. 2.5 provides a general variance-covariance expression for the elements of any principal component with any other principal component, but p.c. 1 is the only one we will be concerned with). Note that $\boldsymbol{\beta}_h \boldsymbol{\beta}_h'$ is the *outer product* (a matrix) of $\boldsymbol{\beta}_h$ with itself . . . there is probably some clever outer-product way to combine this whole expression into a single matrix/tensor operation in terms of Θ and $\boldsymbol{\beta}_h$, but it would just be more confusing. `calc.cpcerr` in the `cpcbp` library calculates the result of (6).

Now suppose we have calculated the error variances $\sigma^2_{\beta_{1j}}$ for each component of the first eigenvector. Then the (ij)th element of the outer-product matrix $\boldsymbol{\beta}_1 \boldsymbol{\beta}_1^T$ is $\beta_{1i}\beta_{1j}$. In general, the formula for combining the errors of two quantities is (from [2])

$$V(f(a,b)) \approx V(a) \left(\frac{\partial f}{\partial a} \right)^2 + V(b) \left(\frac{\partial f}{\partial b} \right)^2 + 2C(a,b) \left(\frac{\partial f}{\partial a} \frac{\partial f}{\partial b} \right), \tag{7}$$

which equals

$$V(a)b^2 + V(b)a^2 + 2C(a,b)ab = a^2 b^2 \left(\frac{V(a)}{a^2} + \frac{V(b)}{b^2} \right) + 2C(a,b)ab \tag{8}$$

if $f(a,b) = a \cdot b$. So the error variance of $\beta_{1i}\beta_{1j}$ is approximately

$$\sigma^2_{\beta_{1i}\beta_{1j}} = (\beta_{1i}\beta_{1j})^2 \left(\frac{\sigma^2_{\beta_{1i}}}{\beta^2_{1i}} + \frac{\sigma^2_{\beta_{1j}}}{\beta^2_{1j}} \right) + 2C(\beta_{1i}, \beta_{1j})\beta_{1i}\beta_{1j} \tag{9}$$

We can write this in matrix formulation as well: Gentle, eq. 1.40 gives

$$V(R) \approx J_g(\theta)V(T)((J_g(\theta))^T, \tag{10}$$

where J is the Jacobian $(\partial g_i/\partial\theta_j)$ and V represents the variance-covariance matrix. The Jacobian of $(\beta_{1i}\beta_{1j})$ is ...

We also have to compute the *covariances* of the elements of the back-projection matrix $b_{ij} = \beta_{1i}\beta_{1j}$ with each other — but we actually will only need to multiply b_{ij} by b_{ik}, so we only need $(\sigma_{b_{ij},b_{ik}})$ [1]

$$\begin{aligned} C_{b_{ij},b_{jk}} =& 2\left(\beta_{1i}\beta_{1j}\sigma_{\beta_{1i},\beta_{1k}} + \beta_{1i}\beta_{1k}\sigma_{\beta_{1i},\beta_{1j}}\right) \\ &+ {\beta_{1i}}^2\sigma_{\beta_{1j},\beta_{1k}} + \beta_{1j}\beta_{1k}\sigma^2_{\beta_{1i}} \end{aligned} \tag{12}$$

Call the back-projection matrix for variable i $\mathbf{b}_i$.

Denote by $\mathbf{M}_i$ the matrix with $\sigma^2_{b_{1i}}$ on the diagonal and $C_{b_{ij},b_{jk}}$ as the off-diagonal elements — this is the variance-covariance matrix of the back-projection vector for variable i. Then if X_g is the vector of the back-projected means of the variables for a group, the back-projection variance for this group is $\sigma^2_{ig,bp} = X_g^T M_i X_g$. If there are n_g samples in group g, then the back-projection sum of squares for is $n_g^2\sigma^2_{ig,bp}$; the total back-projection sum of squares is $\sum_{g=1}^{N_g} n_g^2\sigma^2_{ig,bp}$; and the total error sum of squares is the sum of the within-group sums of squares (the within-group variance for each group, assumed to be equal, times the number of individuals in the group — also assumed to be equal) and the BP sum of squares: $\sum_{g=1}^{N_g} n_g\sigma^2_{x_{ig}} + n_g^2\sigma^2 ig, bp$.

References

[1] Bernard Flury. *Common principal components and related multivariate models.* Wiley, New York, 1988.

[2] Louis Lyons. *A practical guide to data analysis for physical science students.* Cambridge University Press, Cambridge, England, 1991.

[1] **derivation:** We want $\sigma_{b_{ij},b_{ik}} = E[b_{ij}b_{ik}] - E[b_{ij}]E[b_{ik}]$.

$$\begin{aligned} =& E[\beta_{1i}\beta_{1j}\cdot\beta_{1i}\beta_{1k}] - E[\beta_{1i}\beta_{1j}]\cdot E[\beta_{1i}\beta_{1k}] \\ =& (\bar\beta^2_{1i}\bar\beta_{1j}\bar\beta_{1k} + 2(\bar\beta_{1i}\bar\beta_{1j}\sigma_{\beta_{1i},\beta_{1k}} + \bar\beta_{1i}\bar\beta_{1k}\sigma_{\beta_{1i},\beta_{1j}}) \\ &+ \bar\beta^2_{1i}\sigma_{\beta_{1j},\beta_{1k}} + \bar\beta_{1j}\bar\beta_{1k}\sigma^2_{\beta_{1i}}) \\ &- (\bar\beta_{1i}\bar\beta_{1j} + \sigma_{\beta_{1i},\beta_{1j}}) \times (\bar\beta_{1i}\bar\beta_{1k} + \sigma_{\beta_{1i},\beta_{1k}}) \\ =& 2(\bar\beta_{1i}\bar\beta_{1j}\sigma_{\beta_{1i},\beta_{1k}} + \bar\beta_{1i}\bar\beta_{1k}\sigma_{\beta_{1i},\beta_{1j}}) + \bar\beta^2_{1i}\sigma_{\beta_{1j},\beta_{1k}} + \bar\beta_{1j}\bar\beta_{1k}\sigma^2_{\beta_{1i}} \\ &- \bar\beta_{1i}\bar\beta_{1j}\sigma_{\beta_{1i},\beta_{1k}} - \bar\beta_{1i}\bar\beta_{1k}\sigma_{\beta_{1i},\beta_{1j}} - \sigma_{\beta_{1i},\beta_{1j}}\sigma_{\beta_{1i},\beta_{1k}} \end{aligned} \tag{11}$$

(the last three cross terms were left out of the previous derivation).