Principal Component Analysis

Signal Data Science

In this assignment, you'll be learning about principal component analysis (PCA), a method of linear dimensionality reduction. PCA is a method which *rotates* a dataset such that the new ordered set of coordinate axes maximally captures the variation of the data.

In R, PCA can be used via `prcomp()`. A few points to keep in mind: Let `p = prcomp(df, scale.=TRUE)` be the result of PCA run on a data frame `df`. Then:

- `p$rotation` gives a matrix with rows corresponding to columns of `df`, columns corresponding to principal components, and entries corresponding to the *loadings* of a particular column of `df` on a particular principal component. Put another way, each principal component is formed out of a linear combination of the variables of `df`, and the column corresponding to each principal column corresponds to the *coefficients* of that linear combination.
- `p$x` gives the *principal component scores* for each row of `df`, that is, the *actual value* of each principal component for every row in `df`.
- `p$sdev` gives the *eigenvalues of the covariance matrix* of the data. One can interpret the `n`th value in `p$sdev` as corresponding to the relative proportion of the variance in the data explained by the `n`th principal component. Put another way, `p$sdev[n] / sum(p$sdev)` is the proportion of the variance in the data explained by the `n`th principal component.

Write up your findings in an "R Markdown" file. (It's one of the options in RStudio when you go to make new files.) At the end, click on the "Knit HTML" button in RStudio, select "Knit HTML", and generate a HTML file which you can upload to Github! (Now you can share your findings with friends and family in a more accessible form!) For additional guidance, read Yihui Xie's blog (written by the author of the `knitr` package, which provides the underlying functionality behind R script → HTML compilation).

PCA on the `msq` dataset

Prepare the data in this fashion:

- Load the `psych` pachage. Extract the columns `active` through `scornful` from the `msq` dataset.
- Look at the number of `NA`s in each column (hint: use `colSums()` in conjunction with `is.na()`). For simplicity, throw out the columns with a very lage number of missing values and subsequently remove all the rows with any `NA`s.

Afterward, run PCA on the remaining variables.

- Write a function `top(n)` that prints out the top 10 *loadings* of the `n`th principal component, ordered by absolute value. This function is useful for interactively exploring the loadings of principal components and subsequently interpreting their meaning.
- Look at the PCA loadings for the first 5-10 principal components. Use `corrplot()` on the loadings (with the `is.corr=FALSE` option) to visually explore these relationships. After doing so, interpret and assign concise names to the principal components which seem to represent something coherent.
- Plot the eigenvalues obtained via `prcomp(...)$sdev`. How do their relative magnitudes relate to the interpretability of each principal component?

One alternative to using regularized linear models is principal component regression (PCR), where we run a regression of the target variable against the first k principal component scores. The parameter k can be selected via cross-validation.

In the following, you can either calculate the RMSE by implementing cross-validation yourself, like you did in the assignment on resampling, or use `CVlm()` from the `DAAG` package. To use `CVlm()`, the `form.lm` parameter necessitates either (1) calling `lm()` and then use `formula()` to extract the corresponding formula or (2) creating the correct formula with `paste()`.

- Suppose that we use the first kth principal components to predict Extraversion and Neuroticism using a simple, unregularized linear model. Calculate a cross-validated RMSE for $k = 1, 2, \ldots, n$ and plot the RMSE values against k. Which value of k performs the best? Compare the RMSE values to the cross-validated RMSE which you calculated in the self-assessment when using regularized linear regression with all of the original variables. Interpret the results.
- Read about the history of trait theories. How do the principal components relate to Extraversion and Neuroticism?

PCA on the speed dating dataset

Return to the aggregated speed dating dataset (`speeddating-aggregated.csv` in the `speed-dating` folder) which you used for logistic regression.

- Clean the data by dropping any rows with `NA`s and run PCA on the 17 self-rated activities.
- Perform the same analysis which you did with the `msq` dataset: looking at and interpreting the loadings of each principal component, visualizing them with `corrplot()`, and looking at the associated eigenvalues. Since there aren't very many variables, you can `cbind()` the activities to the principal component scores and then use `cor()` → `corrplot()` for easy visualization. As before, assign appropriate names to the principal components which seem to have a coherent meaning or interpretation.
- Predict gender, race (restricting to whites and Asians), and career code (restricting to academia and business / finance) using the principal components with logistic regression. Do so with the 1st principal component, the 1st and the 2nd, the 1st, 2nd, and 3rd, ... all the way up to every principal component. (You can just use `glm()` so you get the p-values too.) Interpret the coefficients.
- For the above regressions, use the `pROC` package to calculate the associated areas under the ROC curve. Compare to the results of using stepwise or regularized regression on all of the activities for the same predictions.

In the following, use unregularized multinomial logistic regression as before, by calling `glmnet(..., family="multinomial")` and then passing `s=0` into `predict()` and `coef()`.

- Restrict to the four most commonly listed careers and use unregularized multinomial logistic regression to predict career in terms of average ratings by other participants and the 17 activities. Make predictions on the whole dataset and calculate the principal components of the resulting log-odds ratios. Use `corrplot()` to visualize the loadings of the principal components and interpret their meaning.

We *could* have expanded out the `career_c` variable into binary indicator variables and looked at the principal components of those indicator variables. However, it's better to look at principal components of the log-odds ratios. Intuitively, the latter will capture the "extent" to which someone is a given career, because the logistic regression model incorporates information from the whole dataset. As such, we can better understand the dimensions of "career variation".[1]

[1] Note also that it is more appropriate to use polychoric correlation for calculating the principal components of binary variables than the standard Pearson correlation coefficient.