Statistics and Machine Learning II

Dimensionality Reduction

Coursework: Carrying out Dimensionality Reduction

Luis Da Silva

February 14, 2019

1 Introduction

As data recollection techniques become more sophisticated, bigger data sets are being built not only in terms of how many individuals we are able to sample but also in the number of features we are able to measure, hence high dimensional data problems are becoming more frequent. Happily, many data sets may be represented in much lower dimensionality than the original data space [1]; therefore, dimensionality reduction techniques are required. These models rely on the assumption that continuous latent variables (i.e. hidden variables that we do not observe) are the ones that generate our data.

There are plenty of these techniques already developed, and according to the characteristics of our data, one could be more appropriate than the other. If one assumes Gaussian Distributions for both latent and observed variables, and linear dependence among them, techniques such as Principal Component Analysis (PCA) and Factor Analysis (FA) should produce good results, while some others like Independent Component Analysis (ICA) will perform better if the latent distribution is non-Gaussian [1].

2 Data

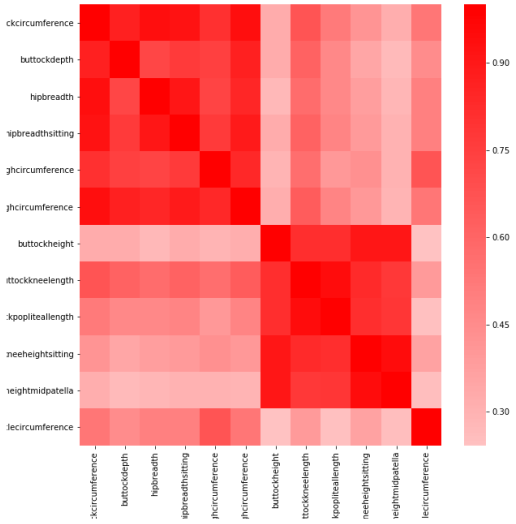
The data set in which I will test these techniques in the 2012 U.S. Army Anthropometric Survey (ANSUR 2)[1]. As the title says, it contains 107 different anthropometric measurements taken from 1986 women and 4082 men. To make analysis simpler, I selected a list of 12 measurements related to the lower body (legs), these are:

- Buttock Circumference
- Buttock Depth
- Buttock Height
- Buttock Knee Length
- Buttock Popliteal Length
- Hip Breadth
- Hip Breadth Sitting
- Knee Height Sitting
- Knee Height Mid Patella
- Lower Thigh Circumference
- Thigh Circumference
- Ankle Circumference

Given that all these measures refer to the same zone of the body, one should expect them to be highly correlated between each other. Figure 1 shows a correlation heatmap for selected variables in the female dataset and, in fact, our expectations hold true for most of the measurements. We are quickly able to see that two clusters arise: one related to body mass (e.g. buttock circumference and thigh circumference) and the other related to stature (e.g. buttock height and knee height while sitting). As a consequence, all

[1]Available here: http://mreed.umtri.umich.edu/mreed/downloads.html#ansur

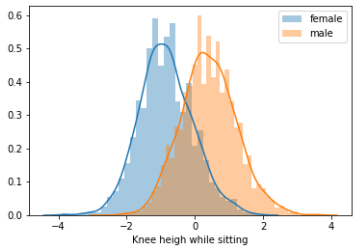
Figure 1: Lower Body Correlation Heatmap on female data set



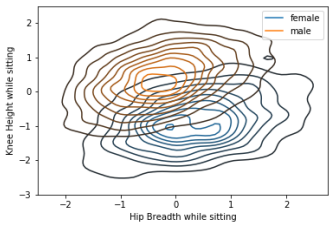
variables in the same cluster contain about the same information and one should expect dimensionality techniques to take advantage of these clusterings to accurately represent the information in only two (or three, because of ankle circumference) components.

The feature on this list that achieves the best gender separability by itself if "Knee height while sitting", its distribution is shown in figure 2a. If one wants to add another dimension, "Hip Breadth while sitting" will perform

Figure 2: Best features for separability of sex



(a) Distribution of Knee height while sitting per sex



(b) KDE on scaled variables based on best 2D separability

best. Figure 2b shows the level of separability achieved by these features.

Before proceeding to apply the techniques, it is important to notice that measurements' distribution vary hugely among each other (Buttock Circumference has a mean of 1021 mm with 76 mm standard deviation for females, while Buttock Depth has a mean of 233 m and standard deviation of 24 mm), so they were standardized by making:

$$\boldsymbol{v}_s = \frac{\boldsymbol{v} - \bar{v}}{\sigma}$$

Where $\boldsymbol{v}_s$ is the vector of standardized values, $\boldsymbol{v}$ is the original vector, $\bar{v}$ its mean and σ its standard deviation.

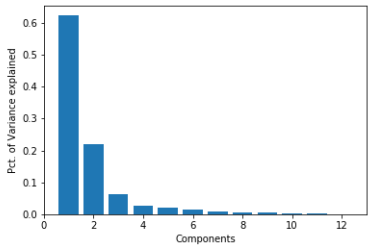
3 Principal Component Analysis

Principal Component Analysis (PCA) is the orthogonal projection of the data to a lower dimensional space [2]. It may be equivalently calculated by minimizing the projection cost function (Residual Sum of Squares) or by maximizing projected variance. This will all sum up to calculating the underlying eigensystem. We generally assume that the target dimension (D) is smaller than the original dimension (M), but PCA still works if one wants to keep $D = M$, in which case there would be no dimensionality reduction, but only an axis rotation that allows decorrelation of components.

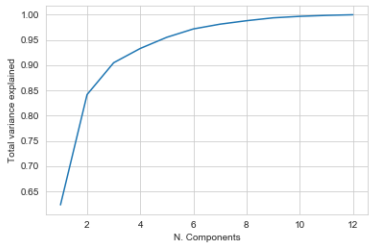
How does PCA perform on our data? First, I calculated eigenvalues on the female dataset. Figure 3a shows the percentage of variance in data that each component explains, and it seems to be recollecting most of the information from the two clusters in the first two components, but component N. 3 still seems important. Figure 3b shows the same information in another way, showing how much variance we're explaining while adding components. How many of these components are important? It depends on the criteria we want to take into account:

- If we have theoretical reasons to choose a number of components, then we would do so. In this case, we could choose 2 because of the two clusters we saw earlier.
- If we want to reach 95% of the variance then we would choose 5 components.

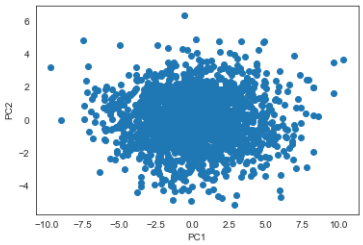
Figure 3: PCA results on female data set



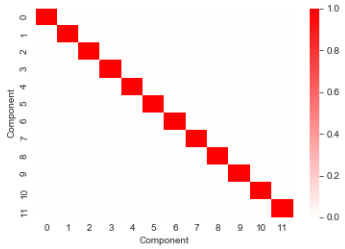
(a) Percentage of variance explained per component



(b) Total variance explained while adding components



(c) First two principal components



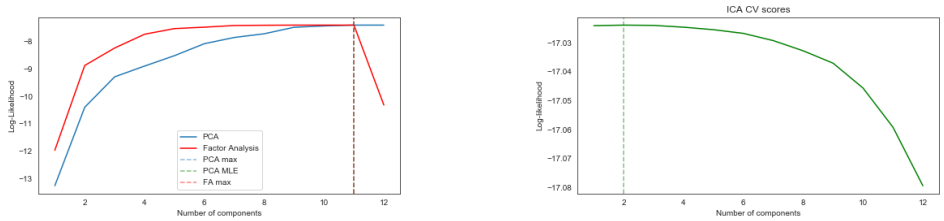
(d) PCA Correlation map

- If we set a threshold on the mean expected variance per component, we would choose only 2 components, as the explained variance ratio of the third component is $0.063 < \frac{1}{12}$
- PCA presents a likelihood function, which allows us to do a direct comparison with other probabilistic density models and to maximize it [1]. If we are to maximize log-likelihood of our original data given the new components (latent variables), then we would choose 11 components (see figure 4a)

In either case, we would have achieved dimensionality reduction and decorrelation of features (see figure 3d).

PCA is not a class separation maximization algorithm as Fisher's linear discriminant for linear dimensionality reduction [1], it doesn't take into account labels (unsupervised learning) and only focuses on maximizing vari-

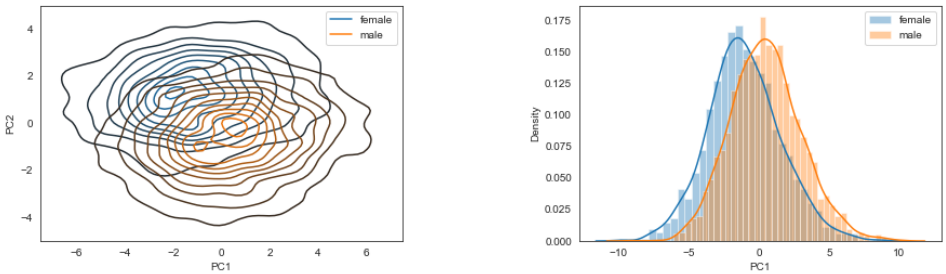
Figure 4: Log-likelihood CV scores comparison



(a) PCA and FA scores

(b) ICA scores

Figure 5: Performance of PC on class separation



(a) KDE on first two PCs

(b) First PC per sex

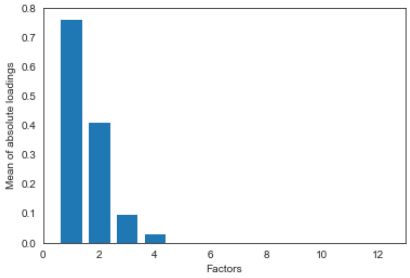
ance in the data. Even though, taking a look at how PCA does on class separation might provide insight into how it's performing.

Fitting PCA on full data shows the same structure described on the female data set, thus I am comfortable using 2 components for visualization of results. Figure 5a shows the distribution for the first two components by gender; the fact that PCA managed to keep a similar structure on a 2D space means that it is performing well.

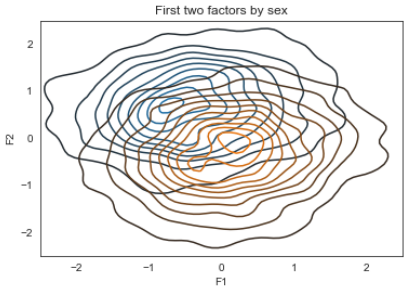
4 Factor Analysis

Just like PCA, Factor Analysis (FA) is a linear-Gaussian latent variable model. It assumes that the observed variables are independent given the

Figure 6: Factor Analysis results



(a) Mean of absolute value of factor components



(b) Firsts factors per sex

latent feature, and its covariance matrix is assumed to have a diagonal. The columns of the component subspace are called factor loadings (or commonalities), and the diagonal of the matrix $\mathbf{\Psi}$ (diagonal matrix for the variance of the observed variables given the latent features) is the uniqueness[1]. What's interesting about FA is not the coordinates by itself, but the form of the latent space. [1].

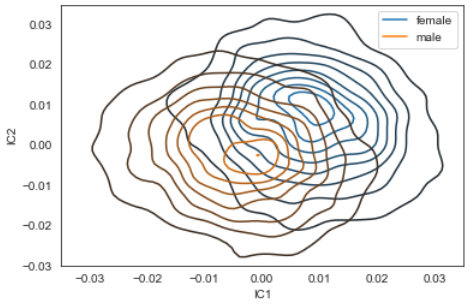
For the female data set, FA returns 4 relevant factors while the rest of them are returned as 0. Figure 6a is produced by applying to every loading component:

$$\overline{|\ L\ |} = \frac{1}{n}\sum_{i=1}^{n} |\ L_i\ |$$

Where n is the number of loading factors. The structure of these means is similar to the one found in PCA, where the first two components have much more importance than the rest of them, therefore we may say that FA has been successful on dimensionality reduction to either 2 or 5 dimensions. On figure 6b we see that FA also keeps the separation observed between gender, but it does not improve it either.

By taking advantage of the likelihood function existent in both PCA and FA, one may compare them directly. We see on figure 4a that FA actually does a better job on dimensionality reduction than PCA on this dataset.

Figure 7: Two components ICA KDE per gender



5 Independent Component Analysis

Independent Component Analysis (ICA), in contrast with PCA or FA, allows us to consider non-Gaussian latent variable distributions. This method assumes that all features are comprehended by linear combinations of the underlying latent variables and tries to separate each of them[1]. This method is especially useful on Blind Source Separation problems, and the classical example is based on the cocktail party problem, in which on separating mixed sounds from a set of synchronized microphones[2].

Figure 4b shows the 5 fold Cross-Validation log-likelihood estimates for the original data and is shown on a separate graph because is considerably lower than PCA and FA, suggesting that it is performing worse and thus the latent variables might be in fact Gaussian. On the other hand, this log-likelihood score function gets maximized when the number of components equals 2, suggesting that it may have separated the two clusters saw in the beginning. On a two dimensional level, ICA distributions perform very similar on gender separation than PCA and FA (see figure 7).

6 Conclusions

We have seen that all techniques considered seem to agree in that reducing the data dimensionality into only two components may be a good idea, and

[2]There are some good demonstrations on this website: https://bit.ly/2IblMQL

even when we know that some information may be lost in the process, most of it is retained. Factor Analysis managed to be crowned as the best performance technique for this particular data set, making it easier to say that the latent variable distribution is in fact Gaussian with a non-isotropic covariance matrix. Finally, none of the techniques achieved greater separability of the gender than the original data had.

References

[1] Christopher M. Bishop. Pattern Recognition and Machine Learning. 2006 Springer Science+Business Media.

[2] H. Hotelling. Analysis of a complex of statistical variables into principal components. 1933 Journal of Educational Psychology

Appendix: Python Code

```
def score(model, X, scoring=None):
    n = X.shape[1]
    scores = []

    for i in range(1, n+1):
        model.n_components = i
        scores.append(np.mean(cross_val_score(model, np.array(X), cv=5,
        scoring=scoring)))

    n_max = np.argmax(scores) + 1

    return scores, n_max

def calculate_score(estimator, X, y=None):
    components = estimator.components_
    n_features = X.shape[1]
    noise_variance = np.ones(n_features, dtype=X.dtype)

    # Get precision using matrix inversion lemma
```

```
    precision = np.dot(components / noise_variance, components.T)
    precision.flat[::len(precision) + 1] += 1.
    precision = np.dot(components.T,
                       np.dot(linalg.inv(precision), components))
    precision /= noise_variance[:, np.newaxis]
    precision /= -noise_variance[np.newaxis, :]
    precision.flat[::len(precision) + 1] += 1. / noise_variance

    n_features = X.shape[1]
    log_like = -.5 * (X * (np.dot(X, precision))).sum(axis=1)
    log_like -= .5 * (n_features * log(2. * np.pi)
                      - fast_logdet(precision))
    return np.mean(log_like)

female = pd.read_csv("ANSUR_II_FEMALE_Public.csv")
female.shape

list(female.columns)

legs = ['buttockcircumference', 'buttockdepth', 'hipbreadth',
'hipbreadthsitting', 'lowerthighcircumference', 'thighcircumference',
       'buttockheight', 'buttockkneelength', 'buttockpopliteallength',
        'kneeheightsitting',
        'kneeheightmidpatella', 'anklecircumference']

len(legs)

plt.figure(figsize=(10,10))
sns.heatmap(female[legs].corr(), cmap='bwr', center=0)
plt.show()

def scaler(series):
    return (series - series.mean())/series.std()


fs = female[legs].copy()
for var in legs:
    fs[var] = scaler(fs[var])

male = pd.read_excel("ANSUR_II_MALE_Public.xlsx")
```

```
ms = male[legs].copy()
for var in legs:
    ms[var] = scaler(ms[var])

female['sex'] = 'female'
male['sex'] = 'male'
people = pd.concat([female, male], sort=False)
ps = people[legs].copy()
for var in legs:
    ps[var] = scaler(ps[var])

ps['sex'] = people['sex']

sns.pairplot(ps, hue='sex')
plt.show()

sns.distplot(ps['kneeheightsitting'][ps['sex']=='female'], label='female'
sns.distplot(ps['kneeheightsitting'][ps['sex']=='male'], label='male')
plt.legend()
plt.xlabel("Knee␣heigh␣while␣sitting")
plt.savefig('Graphs/knee_distribution.png')
plt.show()

sns.kdeplot(ps['hipbreadthsitting'][ps['sex']=='female'],
ps['kneeheightsitting'][ps['sex']=='female'], label='female')
sns.kdeplot(ps['hipbreadthsitting'][ps['sex']=='male'],
ps['kneeheightsitting'][ps['sex']=='male'], label='male')
plt.legend()
plt.xlabel('Hip␣Breadth␣while␣sitting')
plt.ylabel('Knee␣Height␣while␣sitting')
plt.xlim((-2.5, 2.75))
plt.ylim((-3, 2.5))
plt.savefig('Graphs/hip_knee.png')
plt.show()

# # PCA

pca = skd.PCA(svd_solver='full')
fs_pca = pca.fit_transform(fs)
fs_var_ratio = pca.explained_variance_ratio_
```

```
n_features = fs_pca.shape[1]
plt.bar(range(1,n_features+1), fs_var_ratio )
plt.xlabel('Components')
plt.ylabel('Pct. of Variance explained')
plt.savefig('Graphs/pca_component_variance.png')
plt.show()

sns.set_style("whitegrid")
sns.lineplot(range(1, n_features+1), fs_var_ratio.cumsum())
plt.xlabel('N. Components')
plt.ylabel('Total variance explained')
plt.savefig('Graphs/pca_cumulative_variance.png')
plt.show()

sns.set_style("white")
plt.scatter(fs_pca[:,0], fs_pca[:,1])
plt.xlabel('PC1')
plt.ylabel('PC2')
plt.savefig('Graphs/pca_scatter.png')
plt.show()


# PCA does average log-likelihood of all samples.
pca_scores, pca_components = score(pca, fs)

fs_pca

sns.heatmap(pd.DataFrame(fs_pca).corr(), cmap='bwr', center=0)
plt.xlabel('Component')
plt.ylabel('Component')
plt.savefig('Graphs/pca_correlation_map.png')
plt.show()

ps_pca = pca.fit_transform(ps.iloc[:,:-1])
ps_var_ratio = pca.explained_variance_ratio_
plt.bar(range(1,n_features+1), ps_var_ratio )
plt.xlabel('Component')
plt.ylabel('Pct. of variance explained')
plt.title('Pct. of variace explained per component')
```

```
plt.savefig('Graphs/all_data_pca_variace.png')
plt.show()

sns.set_style("white")
males = ps['sex'] == 'male'
sns.kdeplot(ps_pca[:,0][~males], ps_pca[:,1][~males], label='female')
sns.kdeplot(ps_pca[:,0][males], ps_pca[:,1][males], label='male')
plt.xlim((-7.5, 7.5))
plt.ylim((-5,5))
plt.xlabel('PC1')
plt.ylabel('PC2')
plt.legend()
plt.savefig('Graphs/all_pca_kde.png')
plt.show()

sns.distplot(ps_pca[:,0][ps['sex']=='female'], label='female')
sns.distplot(ps_pca[:,0][ps['sex']=='male'], label='male')
plt.legend()
plt.xlabel('PC1')
plt.ylabel('Density')
plt.savefig('Graphs/pc1_dist.png')
plt.show()


# # Independent Component Analysis (ICA)
ica = skd.FastICA(max_iter = 1000, tol=0.01)
fs_ica = ica.fit_transform(fs)

ica_scores, ica_components = score(ica, fs, calculate_score)

ica = skd.FastICA(n_components = 2, max_iter = 1000, tol=0.001)
ps_ica = ica.fit_transform(ps.iloc[:,:-1])
sns.kdeplot(ps_ica[:,0][~males], ps_ica[:,1][~males], label='female')
sns.kdeplot(ps_ica[:,0][males], ps_ica[:,1][males], label='male')
plt.xlim((-.035, .035))
plt.ylim((-.03,.035))
plt.xlabel('IC1')
plt.ylabel('IC2')
plt.legend()
plt.savefig('Graphs/all_ica_kde.png')
```

```
plt.show()

# # Factor Analysis

factor = skd.FactorAnalysis()
fs_factor = pd.DataFrame(factor.fit_transform(fs))

fs_factor.shape

factor_scores, factor_components = score(factor, fs)

factor_comp = pd.DataFrame(factor.fit(fs).components_.T)

plt.bar(range(1, n_features+1), np.absolute(factor_comp).mean())
plt.xlabel('Factors')
plt.ylabel('Mean of absolute loadings')
plt.savefig('Graphs/factors_bar.png')
plt.show()

ps_factor = factor.fit_transform(ps.iloc[:,:-1])
sns.kdeplot(ps_factor[:,0][~males], ps_factor[:,1][~males])
sns.kdeplot(ps_factor[:,0][males], ps_factor[:,1][males])
plt.xlim((-2.5,2.5))
plt.ylim((-2.5,2.5))
plt.xlabel('F1')
plt.ylabel('F2')
plt.title('First two factors by sex')
plt.savefig('Graphs/all_factor_scatter.png')
plt.show()

# # Plot

pca = skd.PCA(svd_solver='full', n_components='mle')
pca.fit(fs)
n_components_pca_mle = pca.n_components_

plt.figure(figsize=(8, 4))

plt.plot(range(1, n_features+1), pca_scores, label='PCA')
plt.plot(range(1, n_features+1), factor_scores, label='Factor Analysis',
```

```
plt.axvline(pca_components, label='PCA max', linestyle='--', alpha=.5)
plt.axvline(n_components_pca_mle, label='PCA MLE', linestyle='--',
color='g', alpha=.5)
plt.axvline(factor_components, label='FA max', linestyle='--', color='r',
alpha=.5)
plt.xlabel("Number of components")
plt.ylabel("Log-Likelihood")
plt.legend()

plt.savefig('Graphs/cv_scores_comparison.png')
plt.show()

plt.figure(figsize=(8, 4))
plt.plot(range(1, n_features+1), ica_scores, label='ICA', color='g')
plt.axvline(ica_components, label='ICA max', linestyle='--', color='g',
 alpha=.5)
plt.title("ICA CV scores")
plt.xlabel("Number of components")
plt.ylabel('Log-likelihood')
plt.savefig('Graphs/ica_score_comparison.png')
plt.legend()
plt.show()
```