Principal Component Analysis

Advanced Methods for Data Analysis (36-402/36-608)

Spring 2014

1 Unsupervised learning

1.1 Supervised versus unsupervised

- Up until this point, we've been working in a setting in which we've been given pairs (x_i, y_i), $i = 1, \ldots n$, where x_i is a vector of predictor measurements, and y_i is an associated outcome. Assuming that (X, Y) is a new pair from the same joint distribution, we've learned how to predict Y from X in various settings and in various ways; this is called *supervised learning*

- A related but notably different framework is that of *unsupervised learning*, where we only observe measurements x_i, $i = 1, \ldots n$ (and no outcomes y_i, $i = 1, \ldots n$), and the goal is to discover interesting and lasting structure present in these measurements. We tend to call x_i, $i = 1, \ldots n$ *feature* measurements rather than predictor measurements, since there is no prediction involved

- Suppose that each $x_i \in \mathbb{R}^p$. Understanding the relationship between the variables or features x_{ij}, $j = 1, \ldots p$ is a core problem in unsupervised learning, and as in supervised learning (e.g., consider linear regression, nonparametric regression, classification, generalized linear models), there are many ways to approach it. For the next few lectures we'll study dimension reduction techniques

1.2 Dimension reduction

- *Dimension reduction* the task of transforming our data set $x_i \in \mathbb{R}^p$, $i = 1, \ldots n$ to a new set $z_i \in \mathbb{R}^k$, $i = 1, \ldots n$ with less features, i.e., $k < p$ (and often, substantially so). A new feature can be one of the old features, or it can be a some linear or nonlinear combination of old features

- Collect the data points $x_i \in \mathbb{R}^p$, $i = 1, \ldots n$ onto the rows of a matrix $X \in \mathbb{R}^{n \times p}$; note that dimension reduction aims to map this potentially "wide" matrix to a "tall" one $Z \in \mathbb{R}^{n \times k}$ (with rows $z_i \in \mathbb{R}^k$, $i = 1, \ldots n$)

- We want this transformation to preserve the main structure that is present in feature space. It will often be the first step in an analysis, to be followed by, e.g., visualization, clustering (which we'll learn shortly), regression, classification

- We're going to start with linear dimension reduction. This means: looking for straight lines in the feature space along which the input points x_i, $i = 1, \ldots n$ exhibit an interesting trend

- In particular, we're going to interpret "interesting" directions to mean directions of *high variance*

2 Quick review: Euclidean projection

- The dimension reduction technique that we'll learn is centered around the concept of projection, so we'll review this first
- A vector $v \in \mathbb{R}^p$ with $\|v\|_2^2 = v^T v = 1$ is said to be a *unit vector* (or have *unit norm*)
- The *projection* of $x \in \mathbb{R}^p$ onto a unit vector v is $(x^T v)v$. Think of this as $c \cdot v$, with $c = x^T v$ being the coefficient or "score"
- Now consider a data matrix $X \in \mathbb{R}^{n \times p}$, and consider projecting each row $x_i \in \mathbb{R}^p$ onto v. The entries of

$$Xv = \begin{pmatrix} x_1^T v \\ x_2^T v \\ \dots \\ x_n^T v \end{pmatrix} \in \mathbb{R}^n$$

  are the scores, and the rows of

$$Xvv^T = \begin{bmatrix} (x_1^T v)v^T \\ (x_2^T v)v^T \\ \dots \\ (x_n^T v)v^T \end{bmatrix} \in \mathbb{R}^{n \times p}$$

  are the projected points
- What happens if we want to project onto more than one vector? This is straightforward under special cirumstances. Vectors $v_1, v_2 \in \mathbb{R}^p$ are called *orthogonal* if $v_1^T v_2 = 0$. More generally, vectors $v_1, \dots v_k \in \mathbb{R}^p$ are orthogonal if $v_i^T v_j = 0$ for any $i \neq j$
- Vectors $v_1, \dots v_k \in \mathbb{R}^p$ are called *orthonormal* if they are orthogonal and they are unit vectors
- The *projection* of $x \in \mathbb{R}^p$ onto (the space spanned by) orthonormal vectors $v_1, \dots v_k \in \mathbb{R}^p$ is $\sum_{j=1}^k (x^T v_j)v_j$. Again, think of this as $\sum_{j=1}^k c_j \cdot v_j$, with the *score* along the jth vector given by $c_j = x^T v_j$
- Now write the collection $v_1, \dots v_k \in \mathbb{R}^p$ as a matrix $V \in \mathbb{R}^{p \times k}$, where each v_j is a column. Consider a data matrix $X \in \mathbb{R}^{n \times p}$, whose rows we want to project onto the columns of V. Analogous to previous case of a single vector v, the scores are given by $XV \in \mathbb{R}^{n \times k}$, with jth column

$$Xv_j = \begin{pmatrix} x_1^T v_j \\ x_2^T v_j \\ \dots \\ x_n^T v_j \end{pmatrix} \in \mathbb{R}^n,$$

  which contains the scores from projecting X onto v_j. The projected points are the rows of $XVV^T \in \mathbb{R}^{n \times p}$, which can be written as

$$XVV^T = \begin{bmatrix} \sum_{j=1}^k (x_1^T v_j)v_j^T \\ \sum_{j=1}^k (x_2^T v_j)v_j^T \\ \dots \\ \sum_{j=1}^k (x_n^T v_j)v_j^T \end{bmatrix} \in \mathbb{R}^{n \times p}$$

- Tip: if you can internalize the matrix notation, it will often be easier to remember (and use) than the componentwise formulas

3 Principal component analysis

- *Principal component analysis* (PCA) is an old topic. Because it has been widely studied, you will hear it being called different things in different fields
- Consider a data matrix $X \in \mathbb{R}^{n \times p}$, so that we have n points (row vectors) and p features (column vectors). We assume that the columns of X have been centered (i.e., for each, we have subtracted out its sample mean). Note that this will not change the structure that we're interested in finding, since our goal is to find directions of high variance—but centering makes the math much simpler

3.1 First principal component direction and score

- The *first principal component direction* of X is the unit vector $v_1 \in \mathbb{R}^p$ that maximizes the sample variance of $Xv_1 \in \mathbb{R}^n$ when compared to all other unit vectors. Because we have centered X, the sample variance of Xv_1 turns out to be simply

$$\frac{1}{n}\sum_{j=1}^{p}(Xv_1)_j^2 = \frac{1}{n}\|Xv_1\|_2^2 = \frac{1}{n}v_1 X^T X v_1.$$

  Therefore the first principal component direction v_1 can be expressed as

$$v_1 = \underset{\|v\|_2=1}{\operatorname{argmax}}\ v^T(X^T X)v$$

- Accordingly, we define the *first principal component score* as $Xv_1 \in \mathbb{R}^n$. Note that its components are the scores from projecting the rows of X onto v_1
- If we let $d_1 = \sqrt{v_1^T(X^T X)v_1}$, then the quantity $d_1^2/n = \frac{1}{n}v_1^T(X^T X)v_1$ is called the *amount of variance explained* by v_1
- Finally, $u_1 = (Xv_1)/d_1 \in \mathbb{R}^n$ is sometimes referred to as the normalized first principal component score
- In a nutshell: consider projecting the rows of X onto any vector $v \in \mathbb{R}^p$, and looking at the scores from this projection; the first principal component direction v_1 is the direction that makes these scores the most spread out (highest sample variance)

3.2 Subsequent principal component directions and scores

- What happens next? The idea is to successively find *orthogonal* directions of the highest variance. Why orthogonal? Because we've already explained the variance in X along v_1, and now we want to look at variance in a different direction. Any direction not orthogonal to v_1 would neccessarily have some overlap with v_1, i.e., it would create some redundancy in explaining the variance in X. (Plus, it makes the math easier!)
- Given $k-1$ principal component directions $v_1, \ldots v_{k-1} \in \mathbb{R}^p$, orthonormal by construction, we define the *kth principal component direction* $v_k \in \mathbb{R}^p$ to be

$$v_k = \underset{\substack{\|v\|_2=1 \\ v^T v_j=0,\ j=1,\ldots k-1}}{\operatorname{argmax}}\ v^T(X^T X)v$$

- The vector $Xv_k \in \mathbb{R}^n$ is called the *kth principal component score* of X

- Letting $d_k = \sqrt{v_k(X^TX)v_k}$, the quantity $d_k^2/n = \frac{1}{n}v_k^T(X^TX)v_k$ is the *amount of variance explained* by v_k
- Finally, $u_k = (Xv_k)/d_k \in \mathbb{R}^n$ is the normalized kth principal component score

3.3 Properties and representations

- How many principal component directions/scores are there? There are p, because if $v_1, \ldots v_p \in \mathbb{R}^p$ are orthonormal, then they are linearly independent[1]
- For the kth principal component direction $v_k \in \mathbb{R}^p$, note that the entries of $Xv_k = d_k u_k$ are the scores from projecting X onto v_k, written as

$$Xv_k = \begin{pmatrix} x_1^T v_k \\ x_2^T v_k \\ \ldots \\ x_n^T v_k \end{pmatrix} \in \mathbb{R}^n$$

- The directions v_k and normalized scores u_k are only unique up to sign flips
- Matrix representation: let the columns of V contain the principal component directions,

$$V = \begin{bmatrix} v_1 & v_2 & \ldots & v_p \end{bmatrix} \in \mathbb{R}^{p \times p}.$$

  The principal component scores are the columns of XV,

$$XV = \begin{bmatrix} Xv_1 & Xv_2 & \ldots & Xv_p \end{bmatrix} \in \mathbb{R}^{n \times p}.$$

  If you can wrap your head around them, the matrix representations are more concise

4 Practical issues

4.1 PCA for dimension reduction

- As a dimension reduction tool: given data points $x_1, \ldots x_n \in \mathbb{R}^p$, we form the data matrix $X \in \mathbb{R}^{n \times p}$, compute the first k principal component directions $v_1, \ldots v_k \in \mathbb{R}^p$, and stack these onto the columns of $V_k \in \mathbb{R}^{n \times k}$. Then we define

$$Z = XV_k = \begin{bmatrix} Xv_1 & Xv_2 & \ldots & Xv_k \end{bmatrix} \in \mathbb{R}^{n \times k},$$

  the matrix that has the first k principal component scores along its columns. Note that these k columns represent our new features in the dimension-reduced data set
- In other words, let $z_1, \ldots z_n \in \mathbb{R}^k$ denote the rows of Z; then these become our dimension-reduced data points
- Of course, to use this in practice, we're going to have to choose k, the number of principal component scores that we take (and the dimension of our new data points)
- How can we do this? Is cross-validation going to work? It seems not, cross-validation doesn't really carry over naturally for unsupervised learning

[1] To be precise, here we are assuming that $p \leq n$ and $\text{rank}(X) = p$. In general, there are exactly $r = \text{rank}(X)$ principal component directions

- Fortunately, we can look at the *proportion of variance explained* as a function of k. Recall that the amount of variance explained by a single direction v_k was $d_k^2/n = \frac{1}{n} v_k^T (X^T X) v_k$. We define the proportion of variance explained by the first k directions $v_1, \ldots v_k$ as

$$\rho_k = \frac{\sum_{j=1}^k d_j^2}{\sum_{j=1}^p d_j^2}.$$

  This is 0 when $k = 0$ and increases monotically to 1 when $k = p$. If ρ_k is high (close to 1) for a small value of k, then this means that the main structure in X can be explained by a small number of directions

- Typically we will plot ρ_k as a function of k to see precisely what's gained, i.e., to see how much more variability is captured, by adding principal component directions. After this curve flattens out, there's not really any point in utilizing more directions

4.2 Scaling the features

- Recall that we always center the columns of X before computing principal component directions. Another common pre-processing step is to *scale* the columns of X, i.e., to divide each feature by its sample variance, so that each feature in our new X has a sample variance of 1

- Why? Otherwise the sample variance of X along a particular direction v is skewed by the sample variances of the raw features. E.g., if the first feature (first column) of X had a much, much larger sample variance than the rest, then the the first principal component direction will be something close to $v_1 = (1, 0, \ldots 0) \in \mathbb{R}^p$

- Hence, if we are in a setting in which the units of the features are arbitrary, then we scale before PCA

- But scaling is not always appropriate; e.g., when the variables are all measured in the same units in the first place (and hence differences in their sample variance are informative!)

4.3 PCA computations

- There are various ways to compute principal component directions of a data matrix X. One way is via the *singular value decomposition* (SVD) of X:

$$\underset{n \times p}{X} = \underset{n \times p}{U} \underset{p \times p}{D} \underset{p \times p}{V^T}$$

- Here $D = \text{diag}(d_1, \ldots d_p)$ is diagonal with $d_1 \geq \ldots \geq d_p \geq 0$, and U, V both have orthonormal columns. This gives us everything:
  - the columns of V, $v_1, \ldots v_p \in \mathbb{R}^p$, are the principal component directions
  - the columns of U, $u_1, \ldots u_p \in \mathbb{R}^n$, are the normalized principal component scores
  - the jth diagonal element of D squared and divided by n, d_j^2/n, is the variance explained by v_j

- Don't forget that we must first center the columns of X

- Note that

$$XV = UDV^T V = UD$$

because $V^T V = I$. This means that

$$Xv_j = d_j u_j, \quad j = 1, \ldots p$$

showing the two ways of representing the principal component scores, as expected

- Note also that

$$X^T X = V D^2 V^T$$

and so $v_1, \ldots v_p$ are *eigenvectors* of $X^T X$. (Check?)

- The last two facts suggest another way of computing the principal component directions and scores: via an *eigendecomposition* of $X^T X$