Dimensionality Reduction

SC4/SM4 Data Mining and Machine Learning, Hilary Term 2017

Dino Sejdinovic

1.1 Data Matrices and Notation

- We will typically assume that we have collected p variables (features/attributes/dimensions) on n examples (items/observations) which can be represented as an $n \times p$ *data matrix* $\mathbf{X} = (x_{ij})$, where x_{ij} is the observed value of the j-th variable for the i-th example:

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1j} & \dots & x_{1p} \\ x_{21} & x_{22} & \dots & x_{2j} & \dots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{i1} & x_{i2} & \dots & x_{ij} & \dots & x_{ip} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{nj} & \dots & x_{np} \end{bmatrix}. \tag{1.1}$$

- We will denote the rows of $\mathbf{X}$ as $x_i \in \mathbb{R}^p$ and treat them as *column vectors*: i.e., x_i is the transpose of the i-th row of the data matrix $\mathbf{X}$.

$$x_i = \begin{bmatrix} x_{i1} \\ x_{i2} \\ \vdots \\ x_{ip} \end{bmatrix} = [x_{i1}, x_{i2}, \dots, x_{ip}]^\top, \quad i = 1, \dots, n. \tag{1.2}$$

- We often assume that $x_1, \dots, x_n$ are *independent and identically distributed (i.i.d.)* samples of a *random vector* X over $\mathbb{R}^p$. When referring to the j-th dimension of X, we will write $X^{(j)}$.

Broadly speaking, dimensionality reduction aims to, for each data item $x_i \in \mathbb{R}^p$, find a lower dimensional representation $z_i \in \mathbb{R}^k$ with $k \ll p$ such that the map $x \mapsto z$ preserves certain *interesting statistical properties* in data.

1.2 Principal Components Analysis

Principal Components Analysis (PCA) is a dimensionality reduction technique which aims to preserve *variance* in the data. PCA is a *linear* dimensionality reduction technique: it essentially looks for a *new basis* to represent a noisy dataset.

For simplicity, we will assume for PCA that our dataset is *centred*, i.e., that its average is $\bar{x} =$

$\frac{1}{n}\sum_{i=1}^{n} x_i = 0$. If not, we can always subtract it from each x_i (this is called *data centering*). Thus, we can write the *sample covariance matrix* S as

$$S = \widehat{\mathrm{Cov}}(X) = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})(x_i - \bar{x})^\top = \frac{1}{n-1}\sum_{i=1}^{n} x_i x_i^\top = \frac{1}{n-1}\mathbf{X}^\top\mathbf{X}. \tag{1.3}$$

Matrix S is symmetric and positive semi-definite.

PCA recovers an orthonormal basis $v_1, v_2, \ldots, v_p$ in $\mathbb{R}^p$ – vectors v_i are *called principal components (PC) or loading vectors* – such that:

- The first principal component (PC) v_1 is the *direction of greatest variance* of data.
- The j-th PC v_j is the *direction orthogonal to* $v_1, v_2, \ldots, v_{j-1}$ *of greatest variance*, for $j = 2, \ldots, p$.

Given this basis, the k-dimensional representation of data item x_i is the vector of projections of x_i onto the first k PCs:

$$z_i = V_{1:k}^\top x_i = \left[v_1^\top x_i, \ldots, v_k^\top x_i\right]^\top \in \mathbb{R}^k,$$

where $V_{1:k} = [v_1, \ldots, v_k]$ is a $p \times k$ matrix. This gives us the *transformed data matrix*, also called the *scores matrix*

$$\mathbf{Z} = \mathbf{X}V_{1:k} \in \mathbb{R}^{n\times k}. \tag{1.4}$$

1.2.1 Deriving the first principal component

Recall that we model our dataset is an i.i.d. sample $\{x_i\}_{i=1}^n$ of a random vector $X = \left[X^{(1)} \ldots X^{(p)}\right]^\top$. Projections to PCs define a linear transformation of X given by $Z = V_{1:k}^\top X$ which is a k-dimensional random vector. Dimensions of Z are called *derived variables*. Consider the first dimension of Z:

$$Z^{(1)} = v_1^\top X = v_{11}X^{(1)} + v_{12}X^{(2)} + \cdots + v_{1p}X^{(p)}. \tag{1.5}$$

The first PC $v_1 = [v_{11}, \ldots, v_{1p}]^\top \in \mathbb{R}^p$ is chosen to maximise the sample variance $\widehat{\mathrm{Var}}(Z^{(1)}) = v_1^\top \widehat{\mathrm{Cov}}(X) v_1$, i.e. it is defined as the solution to

$$\begin{aligned} &\max_{v_1} \; v_1^\top S v_1 \\ &\text{subject to: } v_1^\top v_1 = 1. \end{aligned}$$

By considering the Lagrangian:

$$\mathcal{L}(v_1, \lambda_1) = v_1^\top S v_1 - \lambda_1\left(v_1^\top v_1 - 1\right) \tag{1.6}$$

and the corresponding vector of partial derivatives

$$\frac{\partial \mathcal{L}(v_1, \lambda_1)}{\partial v_1} = 2Sv_1 - 2\lambda_1 v_1 \tag{1.7}$$

we obtain the eigenvector equation $Sv_1 = \lambda_1 v_1$, i.e. v_1 must be an eigenvector of S and the dual variable λ_1 is the corresponding eigenvalue. Since $v_1^\top S v_1 = \lambda_1 v_1^\top v_1 = \lambda_1$, the first PC must be the eigenvector associated with the *largest eigenvalue* of S.

1.2.2 Subsequent principal components

Similarly, the second PC maximizes the sample variance $\widehat{\mathrm{Var}}(Z^{(2)}) = v_2^\top \widehat{\mathrm{Cov}}(X) v_2$ of the second derived variable among the directions orthogonal to v_2, i.e.

$$\begin{aligned} &\max_{v_2} \; v_2^\top S v_2 \\ &\text{subject to: } v_2^\top v_2 = 1, \; v_1^\top v_2 = 0. \end{aligned}$$

Lagrangian is

$$\mathcal{L}\left(v_2, \lambda_2, \gamma_2\right) = v_2^\top S v_2 - \lambda_2 \left(v_2^\top v_2 - 1\right) - \gamma_2 v_1^\top v_2 \tag{1.8}$$

and setting the corresponding vector of partial derivatives to zero

$$\frac{\partial \mathcal{L}(v_2, \lambda_2, \gamma_2)}{\partial v_2} = 2Sv_2 - 2\lambda_2 v_2 - \gamma_2 v_1 = 0. \tag{1.9}$$

Left-multiplying (1.9) by $v_1^\top$ gives $2v_1^\top S v_2 = \gamma_2$. However, since S is symmetric and v_1 is its eigenvector, we have

$$\gamma_2 = 2v_1^\top S v_2 = 2v_2^\top S v_1 = 2\lambda_1 v_2^\top v_1 = 0. \tag{1.10}$$

Hence $Sv_2 = \lambda_2 v_2$ and similarly as before v_2 must be the eigenvector corresponding to the second largest eigenvalue λ_2 of S.

Continuing the process further, we obtain the *eigenvalue decomposition* of S given by

$$S = V \Lambda V^\top \tag{1.11}$$

where Λ is a diagonal matrix with eigenvalues

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p \geq 0 \tag{1.12}$$

on the diagonal and V is a $p \times p$ orthogonal matrix (i.e. $VV^\top = V^\top V = I$) whose *columns* are the p eigenvectors of S, i.e. the principal components $v_1, \ldots, v_p$.

In summary,

- Derived scalar variable (projection to the j-th principal component) $Z^{(j)} = v_j^\top X$ has sample variance λ_j, for $j = 1, \ldots, p$.
- Derived variables are *uncorrelated*: $\mathrm{Cov}(Z^{(i)}, Z^{(j)}) \approx v_i^\top S v_j = \lambda_j v_i^\top v_j = 0$, for $i \neq j$.
- The *total sample variance* is given by $\mathrm{Tr}(S) = \sum_{i=1}^p S_{ii} = \lambda_1 + \ldots + \lambda_p$, so the *proportion of total variance explained* by the j^{th} PC is $\frac{\lambda_j}{\lambda_1 + \lambda_2 + \ldots + \lambda_p}$

1.2.3 Reconstruction view of PCA

We can map back to the original p-dimensional space using

$$\hat{x}_i = V_{1:k} V_{1:k}^\top x_i. \tag{1.13}$$

This is a *reconstruction* of data item x_i. It can be shown (problem sheet) that PCA gives the *optimal linear reconstruction* based on a k-dimensional compression.

1.2.4 PCA via the Singular Value Decomposition

PCA can also be understood using the Singular Value Decomposition (SVD) of data matrix $\mathbf{X}$. Recall that any real-valued $n \times p$ matrix $\mathbf{X}$ can be written as $\mathbf{X} = UDV^\top$ where

- U is an $n \times n$ orthogonal matrix: $UU^\top = U^\top U = I_n$.
- D is a $n \times p$ matrix with decreasing *non-negative* elements on the diagonal (the singular values of $\mathbf{X}$) and zero off-diagonal elements.
- V is a $p \times p$ orthogonal matrix: $VV^\top = V^\top V = I_p$.

Note that

$$(n-1)S = \mathbf{X}^\top \mathbf{X} = (UDV^\top)^\top (UDV^\top) = VD^\top U^\top UDV^\top = VD^\top DV^\top,$$

using orthogonality of U. The eigenvalues of S are thus the diagonal entries of $\Lambda = \frac{1}{n-1} D^\top D$.

We also have

$$\mathbf{X}\mathbf{X}^\top = (UDV^\top)(UDV^\top)^\top = UDV^\top VD^\top U^\top = UDD^\top U^\top,$$

using orthogonality of V.

The $n \times n$ matrix $\mathbf{B} = \mathbf{X}\mathbf{X}^\top$ with entries $\mathbf{B}_{ij} = x_i^\top x_j$ is called the *Gram matrix* of dataset $\mathbf{X}$. Note that $\mathbf{B}$ and $(n-1)S = \mathbf{X}^\top \mathbf{X}$ have the same nonzero eigenvalues, equal to the non-zero squared singular values of $\mathbf{X}$ (non-zero entries on the diagonals of $D^\top D$ and $DD^\top$).

If we consider projections to *all principal components*, the transformed data matrix is

$$\mathbf{Z} = \mathbf{X}V = UDV^\top V = UD, \tag{1.14}$$

If $p \leq n$ this means

$$z_i = [U_{i1} D_{11}, \ldots, U_{ip} D_{pp}]^\top, \tag{1.15}$$

and if $p > n$ only the first n projections are defined (sample covariance will be at most rank n):

$$z_i = [U_{i1}D_{11}, \ldots, U_{in}D_{nn}, 0, \ldots, 0]^\top . \tag{1.16}$$

Thus, $\mathbf{Z}$ can be obtained from the eigendecomposition of Gram matrix $\mathbf{B}$. When $p \gg n$, eigendecomposition of $\mathbf{B}$ requires much less computation, $O(n^3)$, than the eigendecomposition of the covariance matrix, $O(p^3)$, so is the preferred method for PCA in that case.

1.3 Biplots

Denote $\mathbf{e}_j = [0, \ldots, 0, 1, 0 \ldots, 0] \in \mathbb{R}^p$ with 1 at the j-th dimension. This the unit vector pointing in the direction of the original variable $X^{(j)}$. Let us write

$$\nu_j \in \mathbb{R}^p = V^\top \mathbf{e}_j = [V_{j1}, \ldots, V_{jp}]^\top$$

for the j-th row of V (should not be confused with v_j which is the j-th column of V, and the j-th principal component). Thus ν_j is the projection of $\mathbf{e}_j$ to principal components and as such indicates the weighting each PC gives to the original variable $X^{(j)}$.

Unscaled biplots plot first two dimensions of each ν_j, $j = 1, \ldots, p$ - this visualises *the original variables* in the first two principal components directions.

By SVD, we have that the individual entries in the data matrix are given by

$$x_{ij} = \sum_{\ell=1}^{\min\{n,p\}} U_{i\ell}D_{\ell\ell}V_{j\ell} = z_i^\top \nu_j. \tag{1.17}$$

Scaled biplots consider a set of projections different than (1.15), which is given by (assuming $p \leq n$ for simplicity):

$$\tilde{z}_i = \left[U_{i1}D_{11}^{1-\alpha}, \ldots, U_{ip}D_{pp}^{1-\alpha}\right]^\top , \tag{1.18}$$

for some $\alpha \in [0, 1]$, i.e. the case $\alpha = 0$ recovers the regular projections, i.e., the unscaled biplot.

The case $\alpha = 1$, i.e., $\tilde{\mathbf{Z}} = U_{1:n,1:p}$, is particularly interesting as the sample covariance of the transformed data is

$$\widehat{\mathrm{Cov}}\left(\tilde{Z}\right) = \frac{1}{n-1}U_{1:n,1:p}^\top U_{1:n,1:p} = \frac{1}{n-1}I_p,$$

which means that the derived variables are uncorrelated and have equal variance.

To visualise the original variables in this space, we plot the first two dimensions of each

$$\tilde{\nu}_j = [D_{11}^\alpha V_{j1}, \ldots, D_{pp}^\alpha V_{jp}]^\top.$$

Note that by SVD, we have $x_{ij} = \tilde{z}_i^\top \tilde{\nu}_j$ as in (1.17).

Again, for the case $\alpha = 1$, the scaled biplot has a nice property: since the sample covariance between $X^{(i)}$ and $X^{(j)}$ is

$$\widehat{\mathrm{Cov}}(X^{(i)}X^{(j)}) = S_{ij} = \frac{1}{n-1}\left(VD^\top DV^\top\right)_{i,j} = \frac{1}{n-1}\tilde{\nu}_i^\top \tilde{\nu}_j,$$

we can inspect the angle between the projected variables in the biplot and interpret it as the correlation between the original variables.

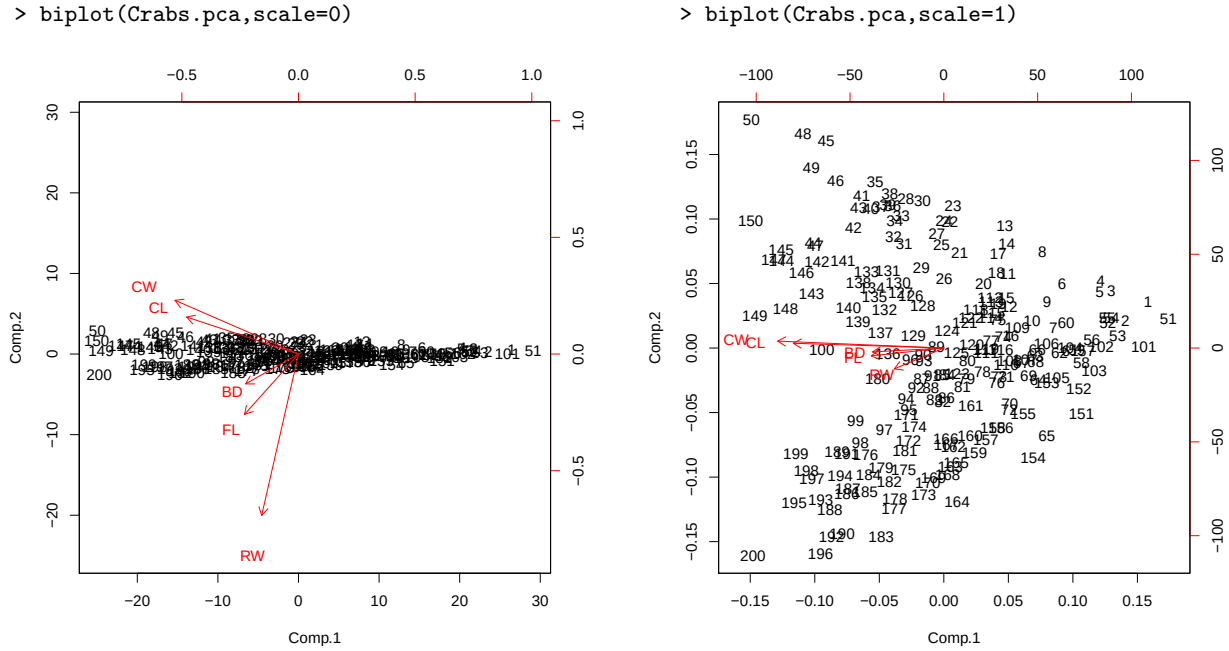

Figure 1: **Left.** Unscaled biplot of Crabs data: the first principal component explains most of the variance. **Right.** Scaled biplot of Crabs data: projections have equal variance and all original variables are strongly correlated.

1.4 Multidimensional Scaling

Suppose there are n points $\mathbf{X}$ in $\mathbb{R}^p$, but we are only given the $n \times n$ matrix $\mathbf{D}$ of squared Euclidean inter-point distances. Can we reconstruct $\mathbf{X}$? Rigid transformations (translations, rotations and reflections) do not change inter-point distances so we certainly cannot recover $\mathbf{X}$ exactly. However, as we will see, it is possible to recover $\mathbf{X}$ up to these transformations.

Let $\mathbf{D}_{ij} = \|x_i - x_j\|_2^2$ be the squared distance between points x_i and x_j. Then:

$$\begin{aligned} \mathbf{D}_{ij} &= (x_i - x_j)^\top (x_i - x_j) \\ &= x_i^\top x_i + x_j^\top x_j - 2 x_i^\top x_j. \end{aligned}$$

Let $\mathbf{B} = \mathbf{X}\mathbf{X}^\top$ be the $n \times n$ Gram matrix of dot-products, $\mathbf{B}_{ij} = x_i^\top x_j$. The above shows that $\mathbf{D}$ can be computed from $\mathbf{B}$. In matrix form,

$$\mathbf{D} = \operatorname{diag}(\mathbf{B})\,\mathbf{1}^\top + \mathbf{1}\operatorname{diag}(\mathbf{B})^\top - 2\mathbf{B}.$$

Exercise 1.1. Show that $\mathbf{B}$ can be recovered from $\mathbf{D}$ if we assume centred data, i.e. $\sum_{i=1}^n x_i = 0$.

Now recall that if we knew $\mathbf{X}$, we can compute the SVD[1] $\mathbf{X} = UDV^\top$.

Also recall the eigendecomposition of $\mathbf{B}$:

$$\mathbf{B} = \mathbf{X}\mathbf{X}^\top = UDD^\top U^\top = U\Lambda U^\top.$$

As $\mathbf{X}$ has rank at most $r = \min(n, p)$, we have at most r non-zero singular values in D. Let $\tilde{x}_i^\top = U_i \Lambda^{\frac{1}{2}} \in \mathbb{R}^r$. If $r < p$, pad $\tilde{x}_i$ with 0s so that it has length p. Then,

$$\tilde{x}_i^\top \tilde{x}_j = U_i \Lambda U_j^\top = \mathbf{B}_{ij} = x_i^\top x_j$$

and we have found a set of vectors in $\mathbb{R}^p$ with dot-products given by $\mathbf{B}$, and hence their distances are given by $\mathbf{D}$, as desired. But note that this eigendecomposition can be obtained from $\mathbf{B}$ without the knowledge of $\mathbf{X}$. The vectors $\tilde{x}_i$ differ from x_i only via the orthogonal matrix $V^\top$ (recall that $x_i^\top = U_i D V^\top = \tilde{x}_i^\top V^\top$) so are equivalent up to rotation and reflections.

Now, we can use only the largest $k < \min(n, p)$ eigenvalues and eigenvectors in the reconstruction, giving the 'best' k-dimensional view of the data. This is called *classical Multidimensional Scaling (MDS)* and it is equivalent to PCA, but as we have seen the original data matrix $\mathbf{X}$ need not even be observed directly – instead we observe the distance matrix $\mathbf{D}$, i.e. data items are observed only through their dissimilarities from other data items.

More generally, MDS is a class of dimensionality reduction techniques which constructs a $z_1, \ldots, z_n \in \mathbb{R}^k$ which (approximately) preserves the inter-item dissimilarities $\mathbf{D}_{ij} = \rho(x_i, x_j)$ (we can use Euclidean distances but other dissimilarities are possible) according to a suitable criterion, with

$$\|z_i - z_j\|_2 \approx \rho(x_i, x_j) = \mathbf{D}_{ij},$$

and differences in dissimilarities measured by the appropriate loss $\Delta(\mathbf{D}_{ij}, \|z_i - z_j\|_2)$. The objective is to find $\mathbf{Z}$ which minimizes the *stress function*

$$S(\mathbf{Z}) = \sum_{i \neq j} \Delta(\mathbf{D}_{ij}, \|z_i - z_j\|_2).$$

Choices of (dis)similarities and (*stress*) functions lead to different algorithms:

- *Classical/Torgerson*: preserves inner products instead - *strain function* (`cmdscale` in R)

$$S(\mathbf{Z}) = \sum_{i \neq j} (\mathbf{B}_{ij} - \langle z_i - \bar{z}, z_j - \bar{z} \rangle)^2$$

- *Metric Shephard-Kruskal*: preserves distances w.r.t. squared stress

$$S(\mathbf{Z}) = \sum_{i \neq j} (\mathbf{D}_{ij} - \|z_i - z_j\|_2)^2$$

- *Sammon*: preserves shorter distances more (`sammon`)

$$S(\mathbf{Z}) = \sum_{i \neq j} \frac{(\mathbf{D}_{ij} - \|z_i - z_j\|_2)^2}{\mathbf{D}_{ij}}$$

[1] do not confuse D (the matrix with singular values on the diagonal and zeros off-diagonal) with $\mathbf{D}$ (the distance matrix)

- *Non-Metric Shephard-Kruskal*: ignores actual distance values, only preserves ranks (`isoMDS`), which alternates between minimizing stress over z's using gradient descent and over an increasing function g using isotonic regression.

$$S(\mathbf{Z}) = \min_{g \text{ increasing}} \frac{\sum_{i \neq j} (g(\mathbf{D}_{ij}) - \|z_i - z_j\|_2)^2}{\sum_{i \neq j} \|z_i - z_j\|_2^2}.$$