

Dimension Reduction

Machine Learning (BSMC-GA 4439)

Wenke Liu

09-13-2018

Biomedical data are usually high-dimensional

- Number of samples (n) is relatively small whereas number of features (p) can be large
- Sometimes $p \gg n$



Problems

- Difficulty in interpretation of data
- The curse of dimensionality
 - Data points are 'sparse' in high-dimensional space
 - Statistical methods that rely on 'local' properties may not work well
 - Extracted patterns may be unstable
- Dimension reduction is usually the first step

Dimension Reduction

- Mapping the data to a low-dimensional space
 - For each p -dimensional data point $\mathbf{x} = (x_1, x_2 \mathbf{K} x_p)^T$ find a k -dimensional representation $\mathbf{x}^* = (x_1^*, x_2^* \mathbf{K} x_k^*)^T$ that captures the content of original data, where
- When there is additional information (supervised)
 - feature selection, ridge regression, LASSO, and other regularization methods
- When we only have \mathbf{X} alone (unsupervised)
 - feature extraction with linear transformation (projection) of the original data
 - nonlinear extensions
- In this lecture, we will focus on the unsupervised case, especially the projection methods

A Note on Notation

- The data matrix \mathbf{X} is a $p \times n$ matrix, sometimes $p \gg n$ (for example, p gene expression levels for n different samples)

$$\begin{bmatrix} x_{11} & \cdots & x_{1j} & \cdots & x_{1n} \\ \vdots & \ddots & \vdots & & \vdots \\ x_{i1} & \cdots & x_{ij} & \cdots & x_{in} \\ \vdots & & \vdots & \ddots & \vdots \\ x_{p1} & \cdots & x_{pj} & \cdots & x_{pn} \end{bmatrix}_{p \times n}$$

Matrix multiplication

"Dot Product"

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \times \begin{bmatrix} 7 & 8 \\ 9 & 10 \\ 11 & 12 \end{bmatrix} = \begin{bmatrix} 58 \\ \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \times \begin{bmatrix} 7 & 8 \\ 9 & 10 \\ 11 & 12 \end{bmatrix} = \begin{bmatrix} 58 & 64 \\ & \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \times \begin{bmatrix} 7 & 8 \\ 9 & 10 \\ 11 & 12 \end{bmatrix} = \begin{bmatrix} 58 & 64 \\ 139 & 154 \end{bmatrix} \quad \checkmark$$

$$m \times n \times n \times p \rightarrow m \times p$$

Mapping data to lower-dimensional space by linear combination

- Seek a factorization of the data matrix \mathbf{X}

$$\mathbf{X}_{p \times n} \approx \mathbf{B}_{p \times k} \mathbf{R}_{k \times n}$$

where column of the matrix \mathbf{B} are k **basis vectors**, and n columns of the matrix \mathbf{R} are **representations** of the n columns of \mathbf{X} defined on \mathbf{B} . $k \leq \min(p, n)$. Both \mathbf{B} and \mathbf{R} contain useful information!

- Or, in other words, find a projection matrix of \mathbf{X} that maps it to a lower dimensional space:

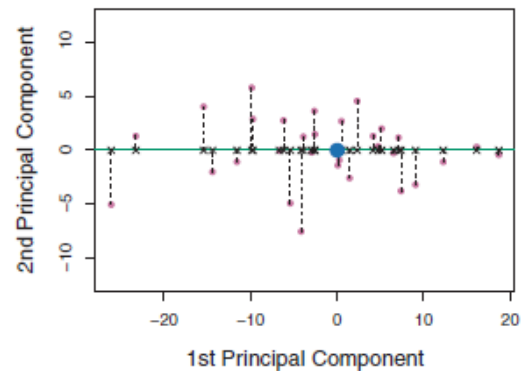
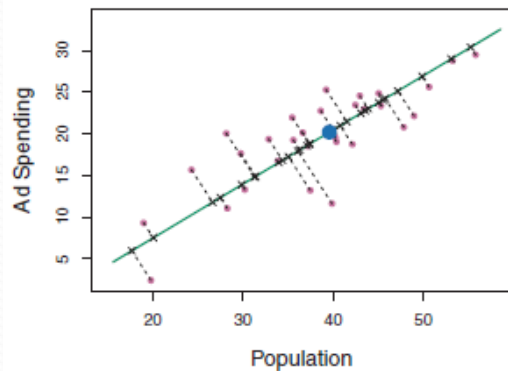
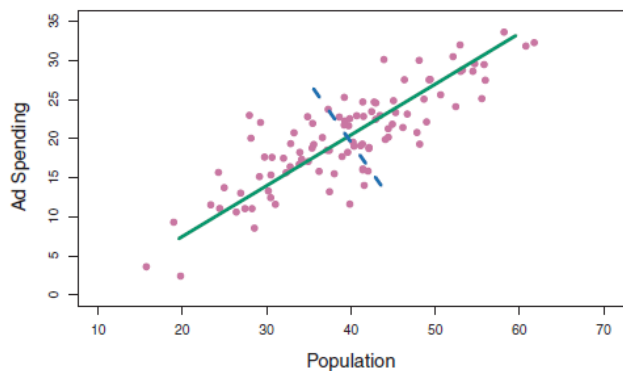
$$\mathbf{R}_{k \times n} = \mathbf{P}_{k \times p} \mathbf{X}_{p \times n}$$

- Different methods have different constraints on \mathbf{B} (or \mathbf{P}) and different criteria of approximating \mathbf{X}

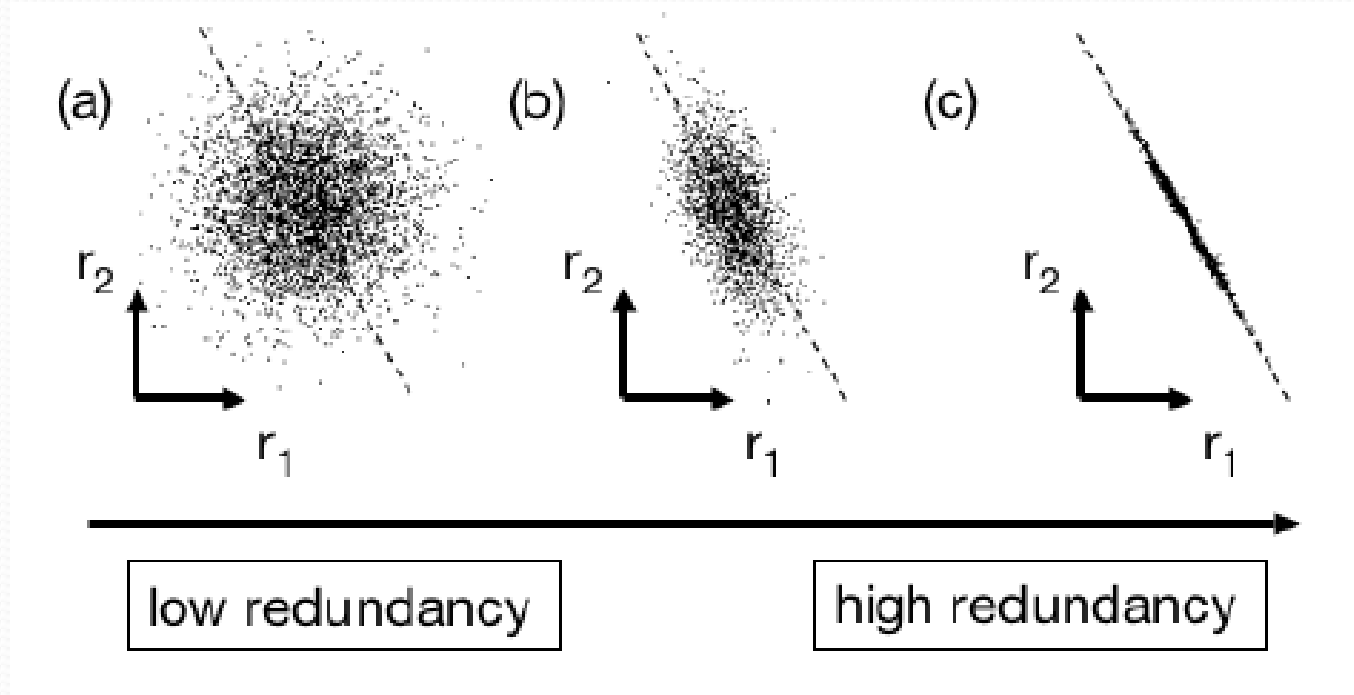
Principal Component Analysis (PCA)

- An intuitive approach to the dimension reduction problem
- Project data matrix \mathbf{X} to a new space, such that:
 - Axes of the new coordinate system are orthogonal and of the same scale as in the original space
 - Projected values preserve variance of the original data
 - Projected vectors are uncorrelated
- In other words, we want to **rotate** the original axes and align them with directions with largest variance

A two-dimensional example of PCA



PCA reveals redundancy in the data



Mathematically...

Find a projection of \mathbf{X} as its representation:

$$\mathbf{Z} = \mathbf{P}\mathbf{X} \quad \mathbf{X} = \mathbf{P}^{-1}\mathbf{Z}$$

Columns of \mathbf{P}^{-1} are orthogonal and of unit length. \mathbf{Z} captures the variance of \mathbf{X} , such that

$$\sum_{i=1}^p (\mathbf{x}_i - \mu_i)(\mathbf{x}_i - \mu_i)^T = \sum_{i=1}^p (\mathbf{z}_i - \mu_i^*)(\mathbf{z}_i - \mu_i^*)^T$$

\mathbf{x}_i and \mathbf{z}_i are rows of \mathbf{X} and \mathbf{Z} . Or in matrix form:

$$tr(\Sigma(\mathbf{X})) = tr(\Sigma(\mathbf{Z}))$$

$$\Sigma(\mathbf{Z}) = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_p)$$

$\Sigma(\mathbf{X})$ is the covariance matrix of \mathbf{X} . Values of λ are as large as possible. Rows of \mathbf{Z} are uncorrelated **principal components** of \mathbf{X} .

Solution of Principal Components

It turns out that the columns of \mathbf{P}^{-1} are unitary **eigenvectors** of $\Sigma(\mathbf{X})$, and $\lambda_1, \lambda_2, \dots, \lambda_p$ are the corresponding **eigenvalues**.

$$\mathbf{P}^{-1} = \mathbf{P}^T \quad \mathbf{P}^T \mathbf{P} = \mathbf{I}_p$$

The i th principal component $\mathbf{z}_i = \mathbf{p}_i \mathbf{X}$, \mathbf{p}_i is the i th row of \mathbf{P} .

$$\text{Var}(\mathbf{z}_i) = \lambda_i$$

\mathbf{z}_i accounts for $\lambda_i / \sum \lambda$ of total variance in \mathbf{X} . Let $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_p$, the first k PCs with significantly large λ values will capture the majority of variability in \mathbf{X} .

Singular Value Decomposition (SVD)

PCA is closely related to **SVD**.

Let \mathbf{A} be a $p \times n$ matrix of real numbers, then there exist an $p \times p$ orthogonal matrix \mathbf{U} and a $n \times n$ orthogonal matrix \mathbf{V} such that

$$\mathbf{A}_{p \times n} = \mathbf{U}_{p \times p} \mathbf{D}_{p \times n} \mathbf{V}_{n \times n}^T$$

$$\begin{array}{c} \mathbf{A} \\ \left[\begin{array}{ccc} a_{11} & \cdots & a_{1n} \\ a_{21} & \cdots & a_{2n} \\ \vdots & \ddots & \vdots \\ a_{p1} & \cdots & a_{pn} \end{array} \right] \\ (p \times n) \end{array} = \begin{array}{c} \mathbf{U} \\ \left[\begin{array}{cccc} u_{11} & u_{12} & \cdots & u_{1p} \\ u_{21} & u_{22} & \cdots & u_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ u_{p1} & u_{p2} & \cdots & u_{pp} \end{array} \right] \\ (p \times p) \end{array} \begin{array}{c} \mathbf{D} \\ \left[\begin{array}{ccc} d_{11} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & d_m \\ 0 & \cdots & 0 \end{array} \right] \\ (p \times n) \end{array} \begin{array}{c} \mathbf{V}^T \\ \left[\begin{array}{ccc} v_{11} & \cdots & v_{13} \\ \vdots & \ddots & \vdots \\ v_{n1} & \cdots & v_{nm} \end{array} \right] \\ (n \times n) \end{array}$$

where $d_{ii} \geq 0$ are singular values of \mathbf{A} , rows of \mathbf{U} are p orthogonal eigenvectors of $\mathbf{A}\mathbf{A}^T$, rows of \mathbf{V} are n orthogonal eigenvectors of $\mathbf{A}^T\mathbf{A}$. Non-zero d_{ii}^2 are the same non-zero eigenvalues of $\mathbf{A}\mathbf{A}^T$ and $\mathbf{A}^T\mathbf{A}$.

Obtain PCs by SVD

If the data matrix \mathbf{X} is **row-centered**, then the sample covariance matrix $\mathbf{S} = \Sigma(\mathbf{X}) = \mathbf{X}\mathbf{X}^T$. After factorizing \mathbf{X} by SVD we have

$$\mathbf{S} = \mathbf{U}\mathbf{D}^2\mathbf{U}^T$$

$$\mathbf{D}^2 = \mathbf{\Lambda}$$

$$tr(\mathbf{S}) = tr(\mathbf{D}^2)$$

Rows of \mathbf{U} are unitary and mutually orthogonal, \mathbf{D}^2 is diagonal and its elements are eigenvalues of $\mathbf{X}\mathbf{X}^T$, \mathbf{U} and \mathbf{D}^2 are exactly what we want for \mathbf{P}^T and $\Sigma(\mathbf{Z})$.

Note: we can always row-center \mathbf{X} by right-multiplying

$$\mathbf{I}_n - (1/n) \mathbf{1}\mathbf{1}^T = \mathbf{I}_n - \mathbf{1}_N$$

Interpretation of PCA

Apply PCA on \mathbf{X} with SVD:

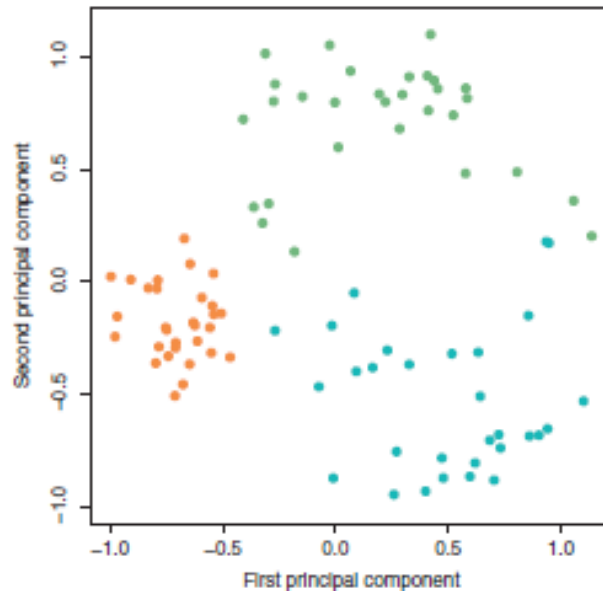
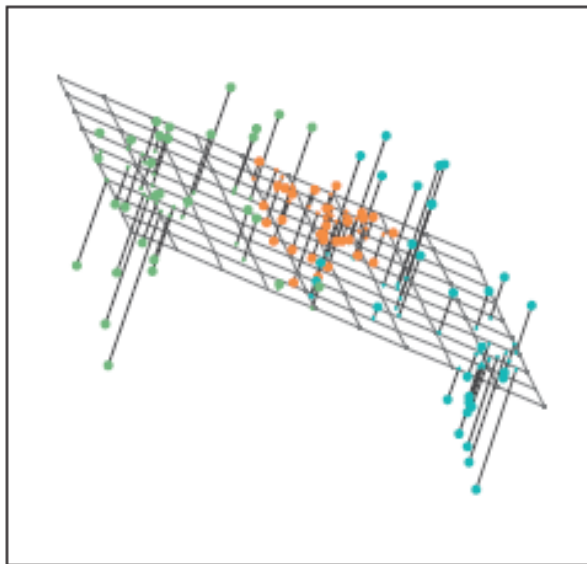
$$\mathbf{X} = \mathbf{U}\mathbf{D}\mathbf{V}^T = \mathbf{U}\mathbf{Z}$$

$$\mathbf{Z} = \mathbf{D}\mathbf{V}^T$$

- Elements of \mathbf{U} are called **loadings**. Columns of \mathbf{U} (eigenvectors of $\mathbf{X}\mathbf{X}^T$) are **loading vectors** that define the rotation directions of the original axes.
- Elements of \mathbf{Z} are called **scores**. Each column of \mathbf{Z} is a representation of the corresponding column of \mathbf{X} in the new space of \mathbf{U} . Each row of \mathbf{Z} is a **principal component**.
- If the first k ($k < p$) rows of \mathbf{U} capture a sufficient large portion of the variance of \mathbf{X} , \mathbf{X} can be reduced to a lower-dimensional space. $\mathbf{X} \approx \mathbf{U}_k \mathbf{Z}_k$

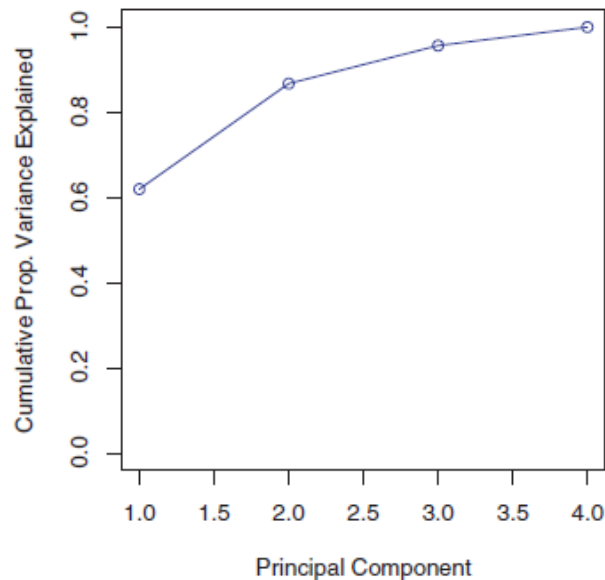
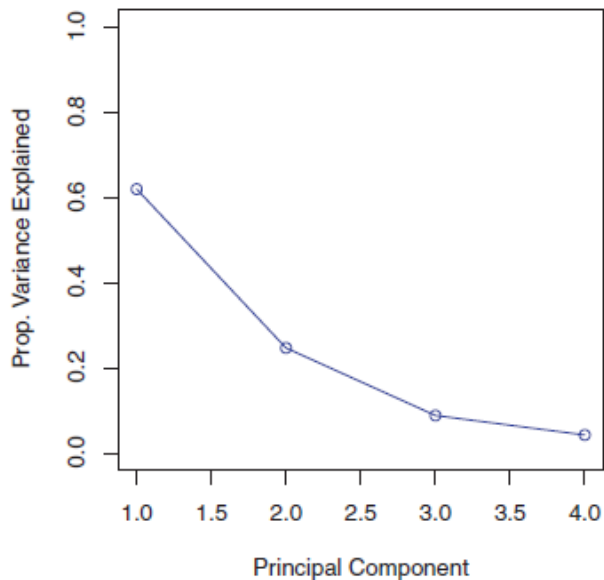
PCA as an exploratory analysis

- PCA is often a pre-processing step before clustering or regression analysis



How Many PCs?

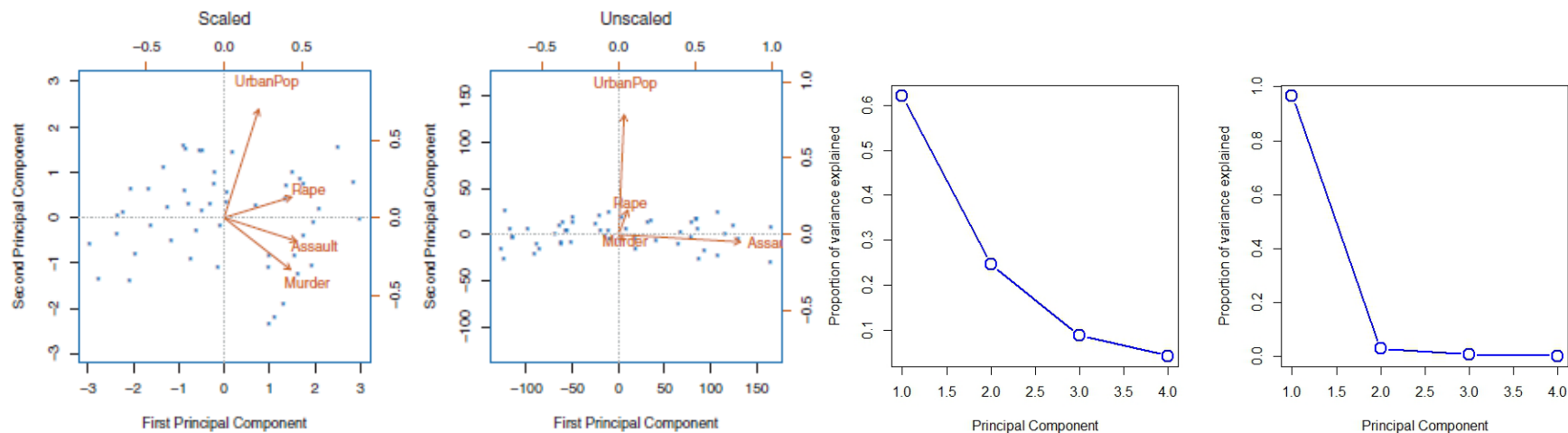
- \mathbf{X} can be reduced to a k -dimensional space by only looking at the first k components that explained large proportions of variance
- Find the 'elbow' on the 'scree plot'



To standardize or not to standardize

- PCA can be performed on the sample covariance matrix **S** or the sample correlation matrix **R**.
- The latter is equivalent to performing PCA on the standardized data (each of the p feature measurements is normalized to have mean 0 and sd 1).
- If the observed features are of different units, it is recommended to standardize the data.

Effects of standardization

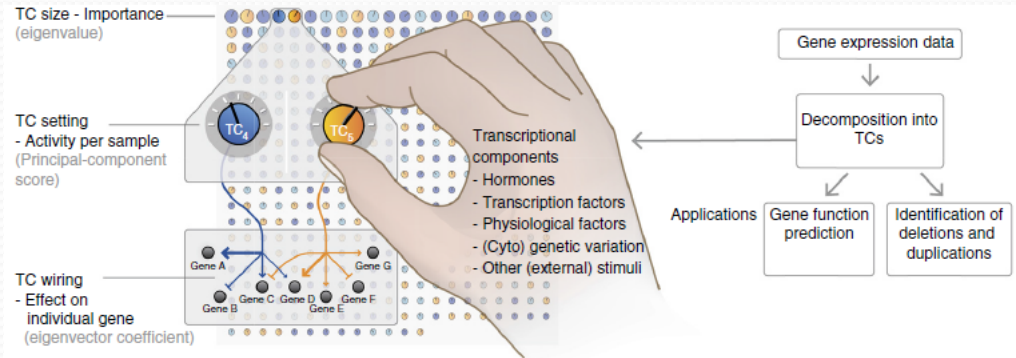


A use case in genomics study

- Apply PCA to the **correlation matrix** of different probe sets (gene expression features)
- Extract transcriptional components (eigenvector loadings) and activity measurements (principal component scores).

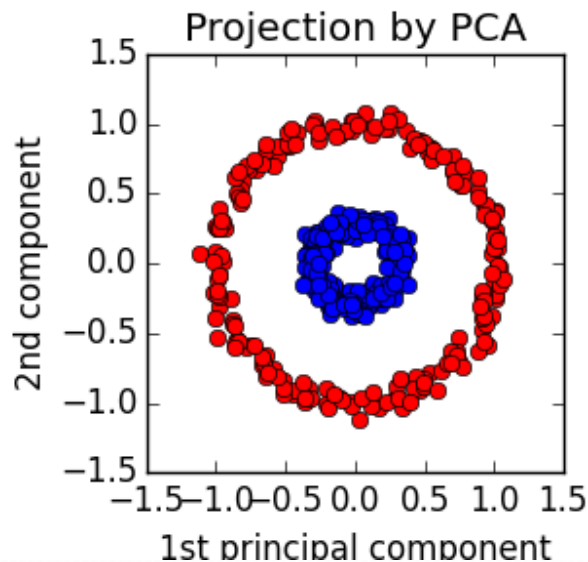
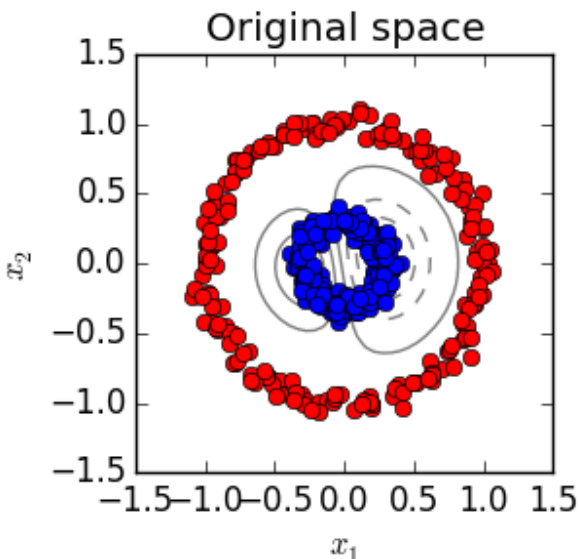
Gene expression analysis identifies global gene dosage sensitivity in cancer

Rudolf S N Fehrmann^{1,2,12}, Juha M Karjalainen^{2,12}, Malgorzata Krajewska¹, Harm-Jan Westra², David Maloney³, Anton Simeonov³, Tune H Pers⁴⁻⁷, Joel N Hirschhorn^{4-6,8}, Ritsert C Jansen⁹, Erik A Schultes^{10,11}, Herman H B M van Haagen¹⁰, Elisabeth G E de Vries¹, Gerard J te Meerman², Cisca Wijmenga², Marcel A T M van Vugt¹ & Lude Franke²



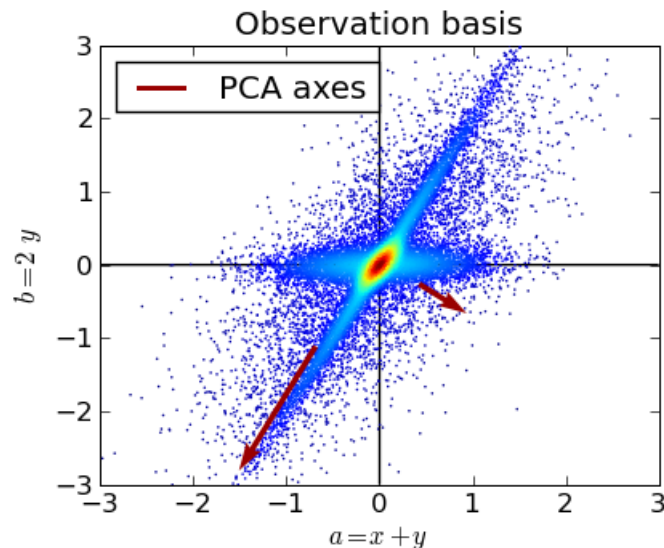
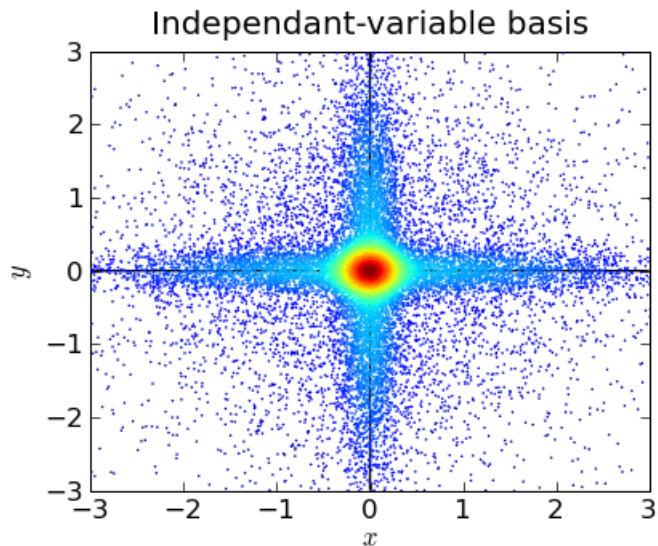
Limitations of PCA

- PCA is a linear method, it may not be productive if the features are non-linear.
- Extensions such as kernel-PCA may improve the analysis.



Limitations of PCA (cont'd)

- PCA captures variance of the data, but variance may not be informative.
- Projective methods with different constraints may be more desirable.



Independent Component Analysis (ICA)

- Looking for linear transformation that result in statistically independent non-Gaussian signals

$$\mathbf{X}=\mathbf{A}\mathbf{S}$$

$$\mathbf{S}=\mathbf{A}^{-1} \mathbf{X}$$

Where \mathbf{A} is the mixing matrix, its columns are latent vectors. Rows of \mathbf{S} are statistically independent signal **sources** with unit variance.

- Based on higher order statistics rather than variance
- Stronger constraints than PCA (independence in addition to lack of correlation)

Formulating ICA

In information theory, dependence among random variables is measured by **mutual information**:

$$I(y_1, y_2 \dots y_m) = \sum_{i=1}^m H(y_i) - H(\mathbf{y})$$

Where $H(y_i)$ is the differential entropy of individual y_i , and $H(\mathbf{y})$ is the differential entropy of the joint distribution.

If $y_1, y_2 \dots y_m$ are **uncorrelated**, mutual information can be expressed in the form of negentropy $J(\mathbf{y})$:

$$I(y_1, y_2 \dots y_m) = J(\mathbf{y}) - \sum_{i=1}^m J(y_i)$$

Where $J(\mathbf{y})$ measures the distance of \mathbf{y} from normality:

$$J(\mathbf{y}) = H(\mathbf{y}_{gauss}) - H(\mathbf{y})$$

Formulating ICA

Negentropy $J(\cdot)$ is **invariant** under invertible linear transformations. Therefore, the search for a linear transformation of \mathbf{X} such that $\mathbf{Z}=\mathbf{W}\mathbf{X}$ has minimized mutual information is equivalent to finding a \mathbf{W} such that negentropy of each row in \mathbf{Z} is maximized, under the constraint that rows of \mathbf{Z} are uncorrelated.

$$I(z_1, z_2, \dots, z_m) = J(\mathbf{w}\mathbf{x}) - \sum_{i=1}^m J(z_i) = J(\mathbf{x}) - \sum_{i=1}^m J(z_i)$$

Formulating ICA

Negentropy $J(z_i)$ is not easy to compute. But it can be approximated using a **contrast function** $G(\cdot)$:

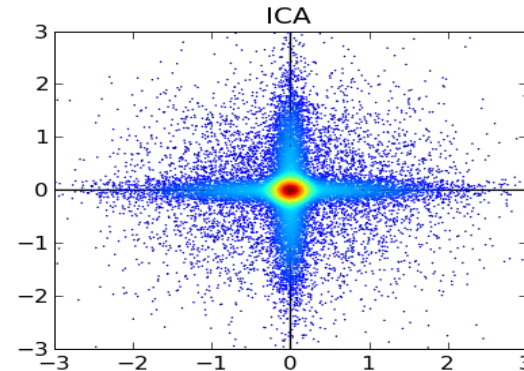
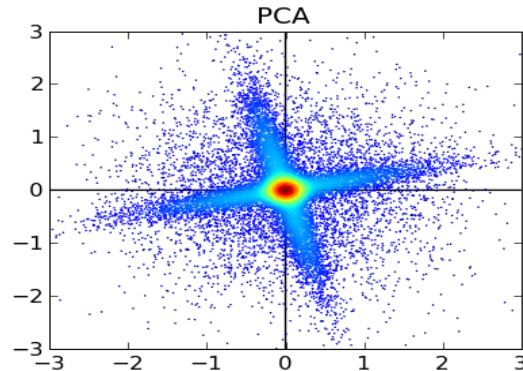
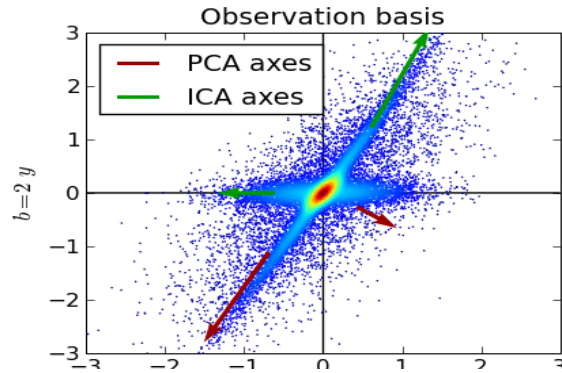
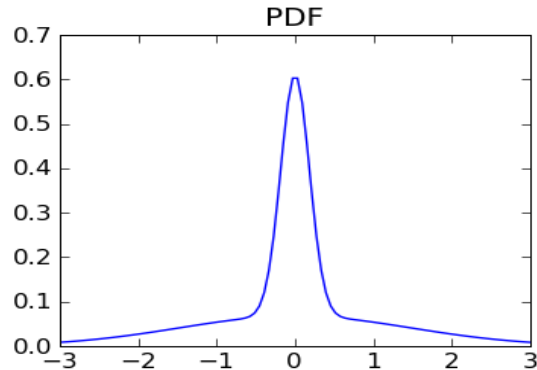
$$J(z) \approx [E\{G(z)\} - E\{G(v)\}]^2$$

Where $G(\cdot)$ is a non-quadratic function, and v is a standardized Gaussian random variable. \mathbf{Z} and \mathbf{W} can be found by maximizing the approximation of each $J(z_i)$, and they are estimates of \mathbf{S} and \mathbf{A}^{-1} .

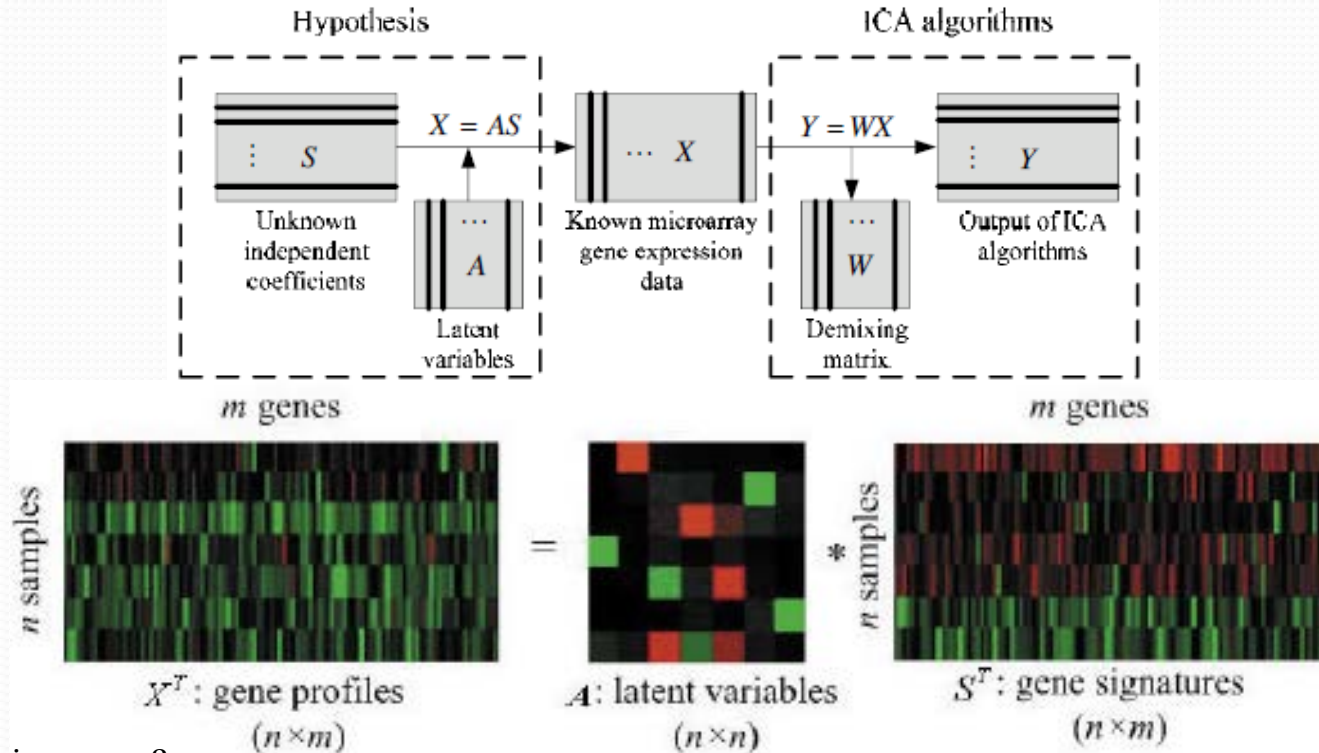
Different algorithms utilized different contrast functions. Some common choices are

$$G_1(y) = \frac{1}{a_1} \log(\cosh(a_1 y)) \quad G_2(y) = -\frac{1}{a_2} \exp(-a_2 y^2 / 2) \quad G_3(y) = -\frac{1}{a_3} y^4$$

ICA finds non-Gaussian directions



Application of ICA on expression data



Limitations of ICA

- No clear selection criteria for components.
- Initial value sensitive.
- Does not work well if distribution of the signal sources is close to Gaussian.

Non-negative Matrix Factorization (NMF)

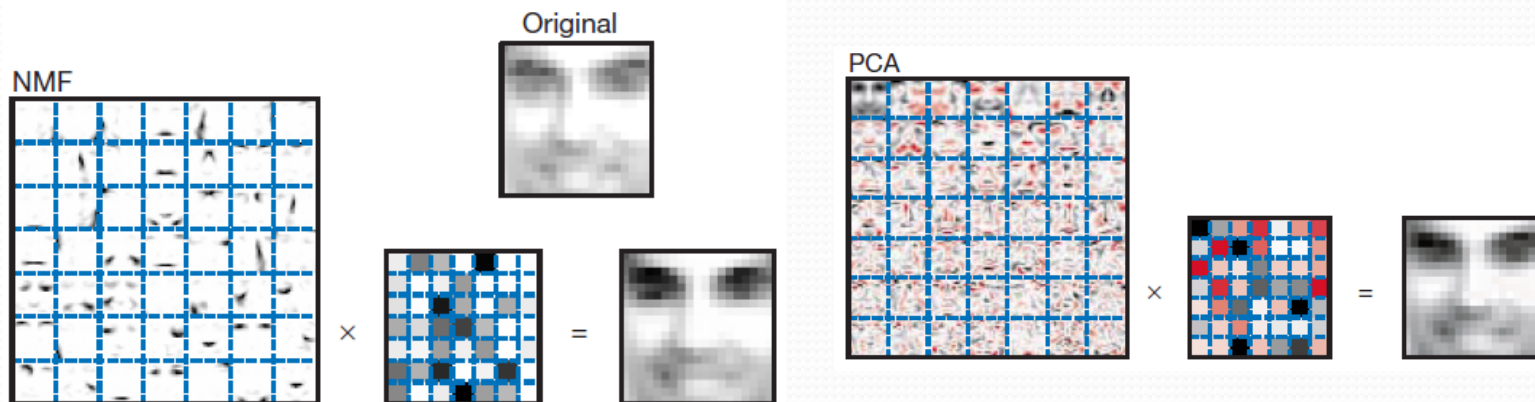
Given a nonnegative matrix \mathbf{X} , find nonnegative matrices \mathbf{W} and \mathbf{H} , such that

$$\mathbf{X}_{p \times n} \approx \mathbf{W}_{p \times k} \mathbf{H}_{k \times n}$$

Columns of \mathbf{W} are basis vectors, columns of \mathbf{H} are coefficients for each sample. The rank k has to be determined based on heuristics.

Comparison between PCA and NMF

- PCA extracts distributed representations on orthogonal directions
- NMF learns additive combination of parts

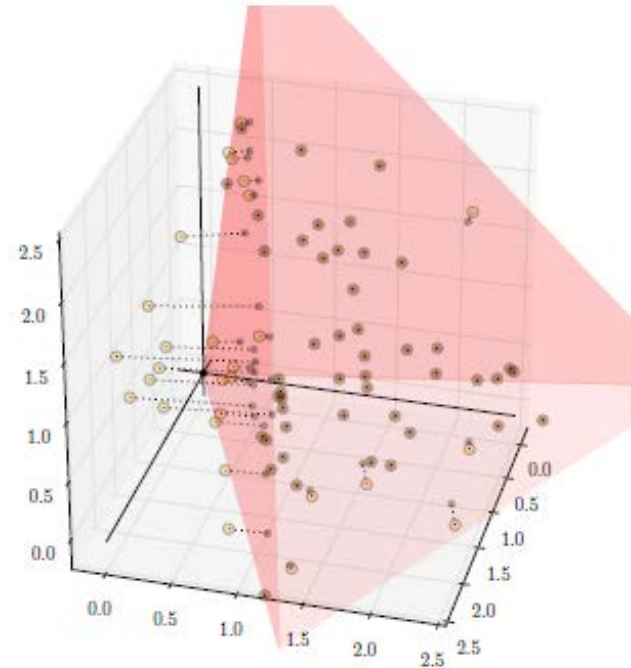
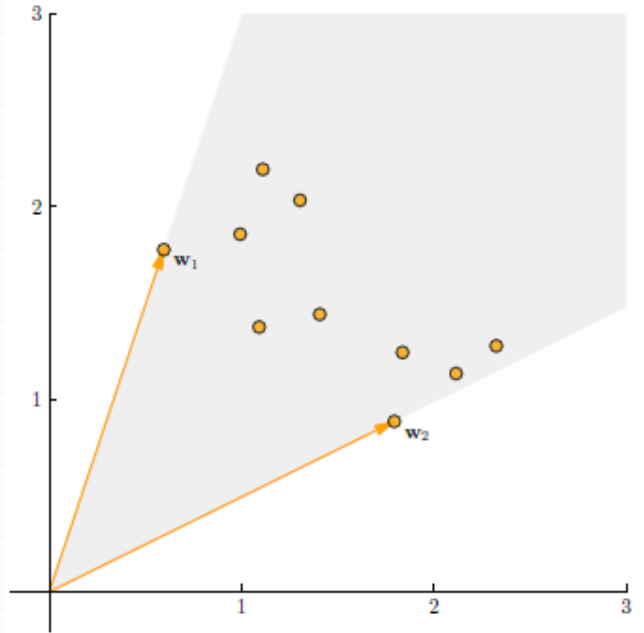


Geometric interpretation of NMF

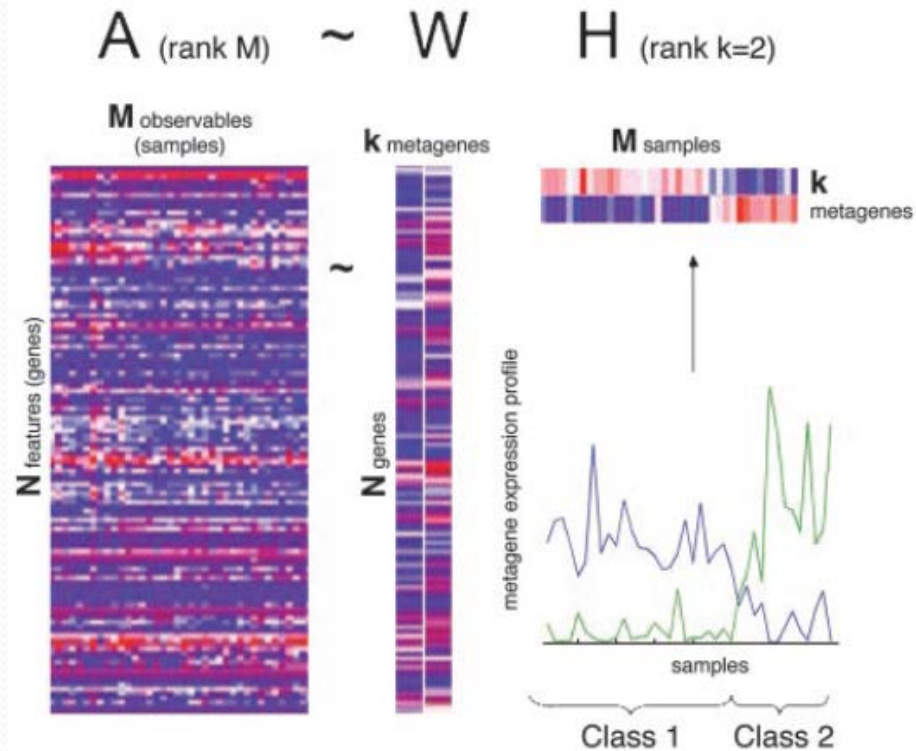
The factorization $\mathbf{X} = \mathbf{WH}$ has a geometric counterpart: all data points \mathbf{x} all lie in the **convex cone** generated by column vectors of \mathbf{W} , which is embedded in the first orthant of \mathbf{R}^p . In fact, if all the data points are strictly positive, there are many such cones, and the factorization is not unique.

Geometrically, the NMF method seeks to approximate the conic hull of \mathbf{X} by a low-dimensional cone in the first orthant. Data points lie outside the cone cannot be reconstructed.

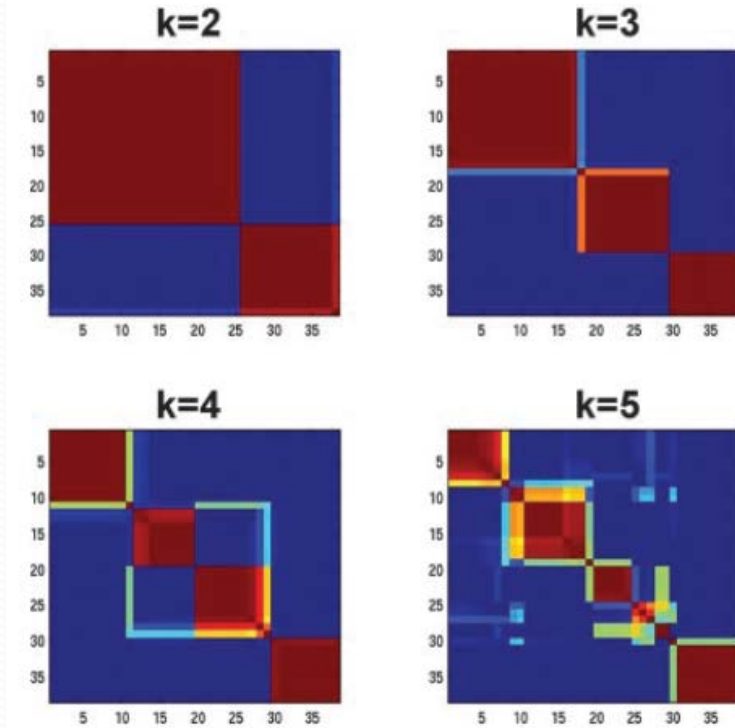
Geometric interpretation of NMF



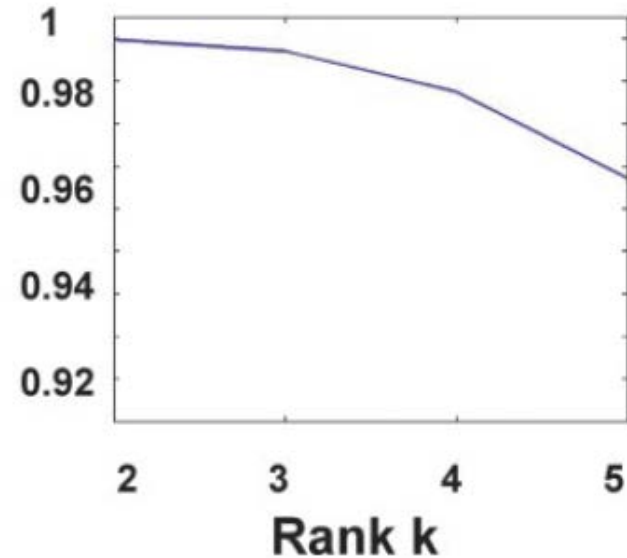
NMF of gene expression data



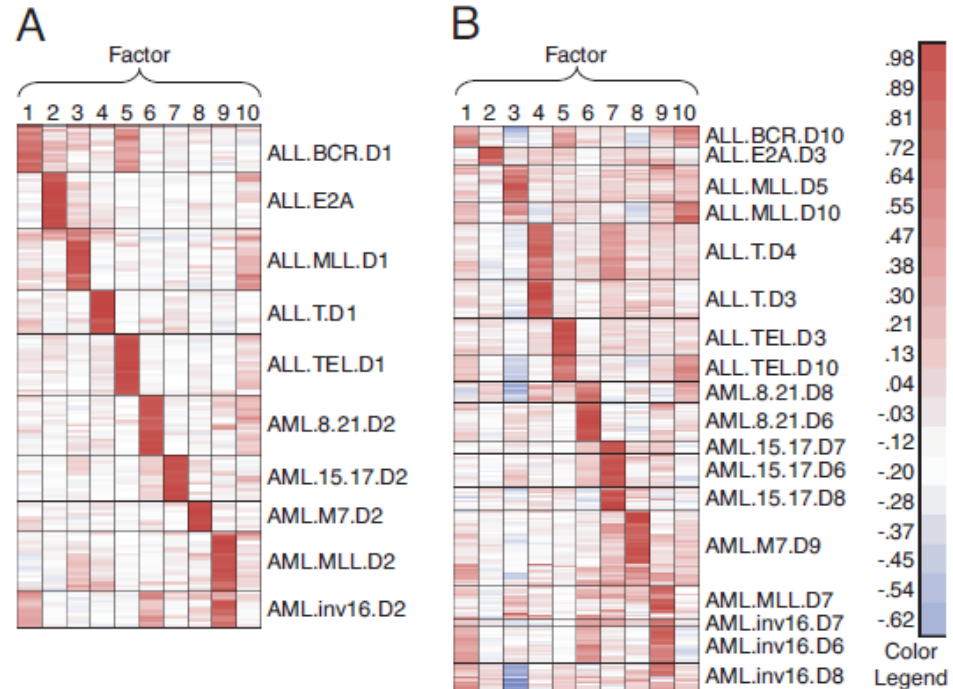
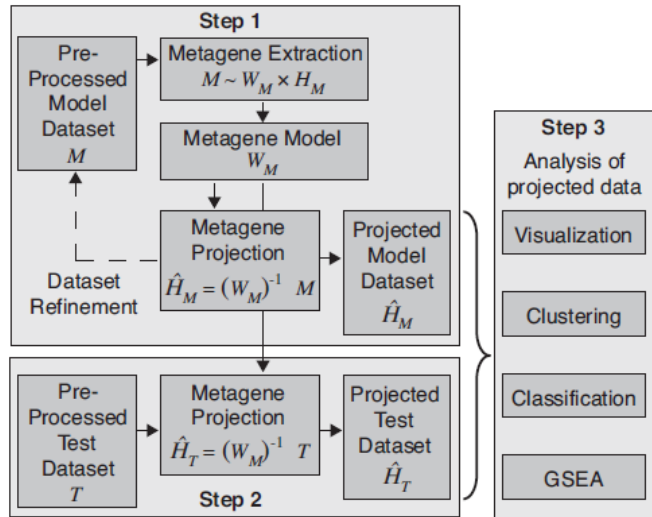
Determine the rank of factorization



Cophenetic Correlation



NMF is widely used in classification of cancer samples

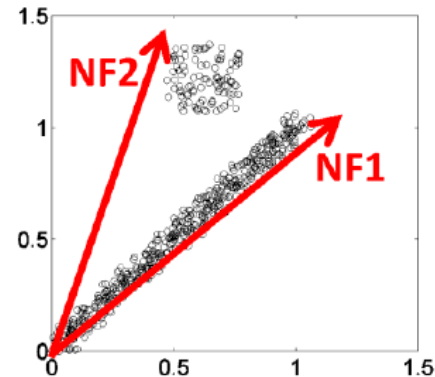
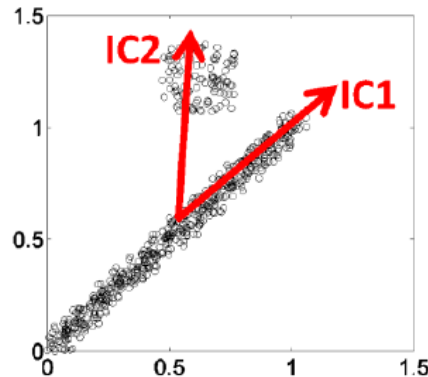
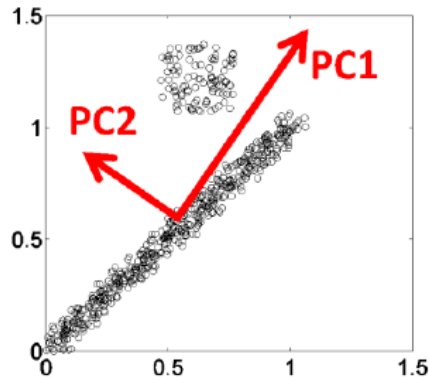


Limitations of NMF

- Usually the NMF solution is not unique
- Initial value sensitive
- High computational complexity
- Projection of new data could be confusing (\mathbf{W} is not always invertible; new \mathbf{H} may contain negative values)

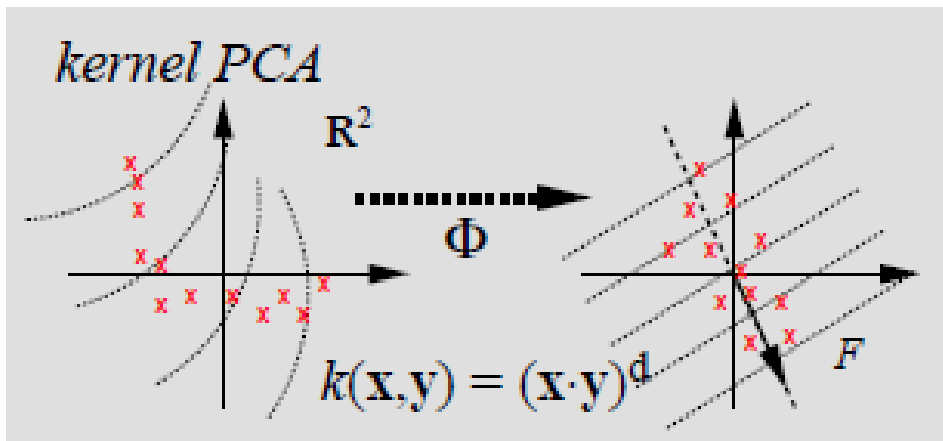
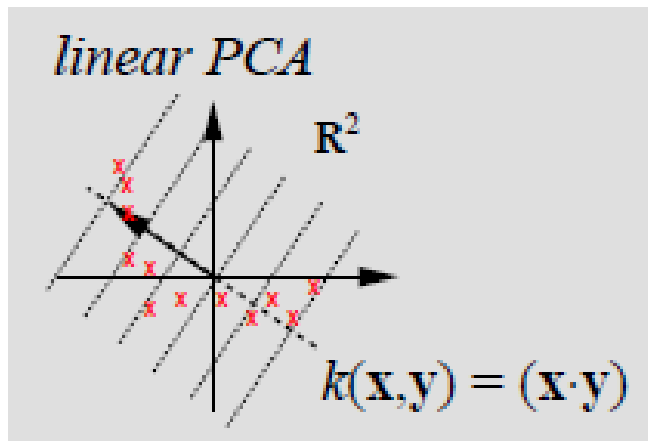
What to use?

- PCA is the most common method, but its interpretation may not be clear
- ICA extracts non-Gaussian information from the data, but it may not be clear which components are more ‘important’.
- NMF produces additive features, but it’s heavy in computation, and the solution can be unstable



Nonlinear extensions: kernel PCA

In PCA, \mathbf{X} is projected to $\mathbf{Z}=\mathbf{D}\mathbf{V}^T$, where columns of \mathbf{V}^T are eigenvectors of $\mathbf{X}^T\mathbf{X}$. Kernel PCA, instead, seeks to find representation of \mathbf{X} by finding eigenvalues and eigenvectors of a centered kernel matrix $\mathbf{K}^*=\mathbf{K}-\mathbf{1}_N\mathbf{K}-\mathbf{K}\mathbf{1}_N-\mathbf{1}_N\mathbf{K}\mathbf{1}_N$, where $\mathbf{K}_{ij}=k(\mathbf{x}_i, \mathbf{x}_j)=\Phi(\mathbf{x}_i)^T\Phi(\mathbf{x}_j)$

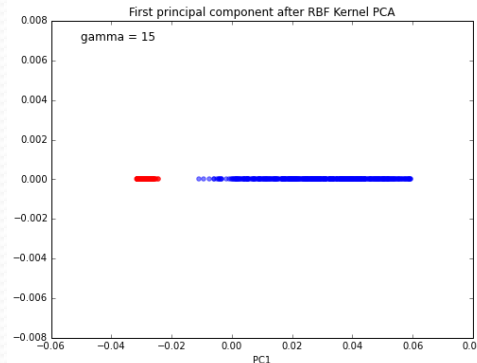
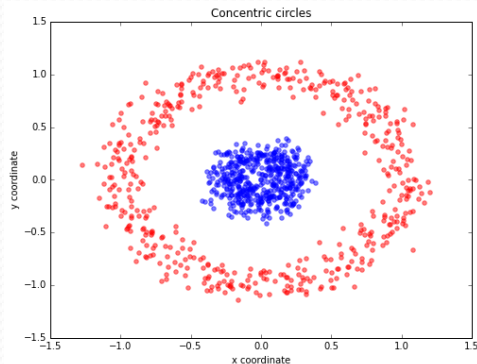


The kernel trick

The trick is, $\Phi(\cdot)$ can be a very complex function while $k(\cdot)$ remains relatively simple. kPCA projects the data to a nonlinear space without explicitly calculate each projection directions.

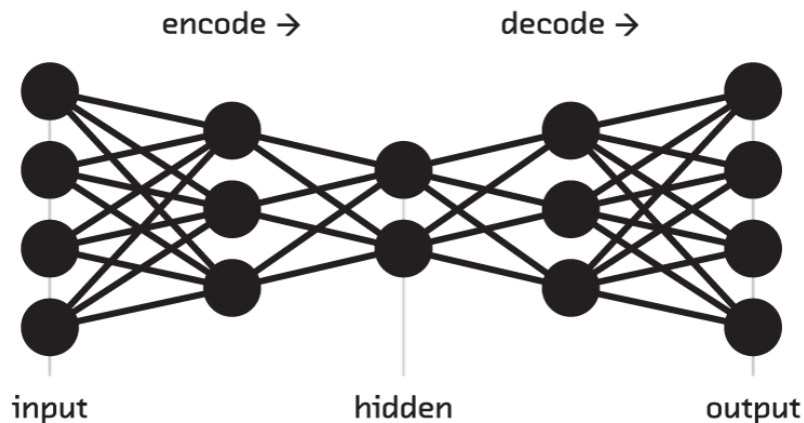
A popular kernel functions is the Gaussian Radial Basis Function (RBF):

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\sum_k (x_{ik} - x_{jk})^2}{2\sigma^2}\right)$$



Dimension reduction with neural networks

- A feed-forward neural network with non-linear activation function can approximate any function.
- An **autoencoder** network trained to reconstruct its input may represent the data in a lower dimensional latent space.



Visualizing high-dimensional data

- Multidimensional scaling (MDS)
- t-Distributed Stochastic Neighbor Embedding (t-SNE)

MDS

- Multidimensional Scaling (MDS) maps high-dimensional data to low-dimensional space. It preserves the pairwise distance (or dissimilarity) between original data points.
- This can be done in linear, nonlinear or nonmetric manner.

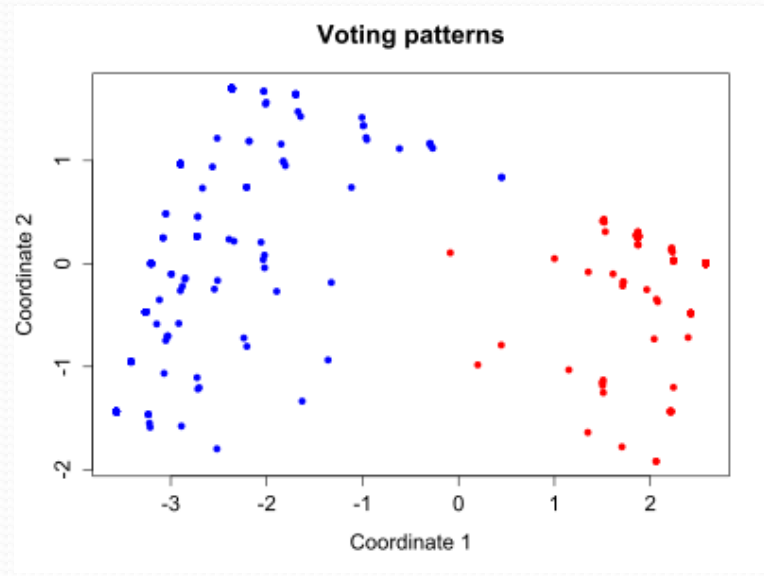
Metric MDS

- In the case of ‘classical’ MDS, it is equivalent to PCA.

$$S(z_1, z_2 \dots z_n) = \sum_{i < j} (d_{ij} - \|z_i - z_j\|)^2$$

- Nonlinear MDS such as Sammon mapping preserves the nearby points:

$$S_{NL}(z_1, z_2 \dots z_n) = \frac{1}{\sum_{i < j} d_{ij}} \sum_{i < j} \frac{(d_{ij} - \|z_i - z_j\|)^2}{d_{ij}}$$



Non-metric MDS

- Use an increasing function of the original distance (preserve ranks).

$$S_{NM}(z_1, z_2, \dots, z_n) = \frac{\sum_{i < j} (\theta(d_{ij}) - \|z_i - z_j\|)^2}{\sum_{i < j} \|z_i - z_j\|^2}$$

- Have to optimize both the new coordinates and the function θ .
- Enables nonlinear transformation.

t-SNE

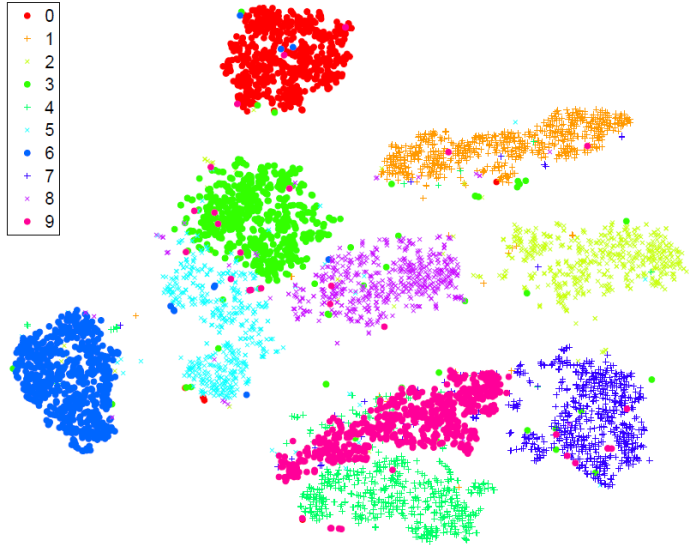
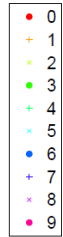
- Preserve the **joint probabilities** of pairs of original data.
- Model the original data with Gaussian distribution, represent in lower dimensional space with t -distribution, minimize the Kullback-Leibler divergence as the cost function.

$$p_{j|i} = \frac{\exp(-\|x_i - x_j\|^2 / 2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|x_i - x_k\|^2 / 2\sigma_i^2)} \quad p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n}$$

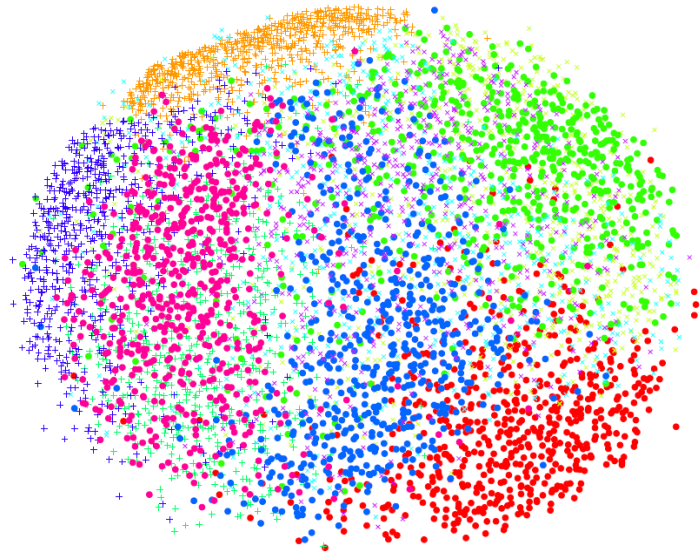
$$q_{ij} = \frac{(1 + \|y_i - y_j\|^2)^{-1}}{\sum_{k \neq l} (1 + \|y_k - y_l\|^2)^{-1}} \quad C = KL(P||Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

- Computationally expensive, have to select random subsets for large data.


t-SNE vs. nonlinear metric MDS



(a) Visualization by t-SNE.



(b) Visualization by Sammon mapping.

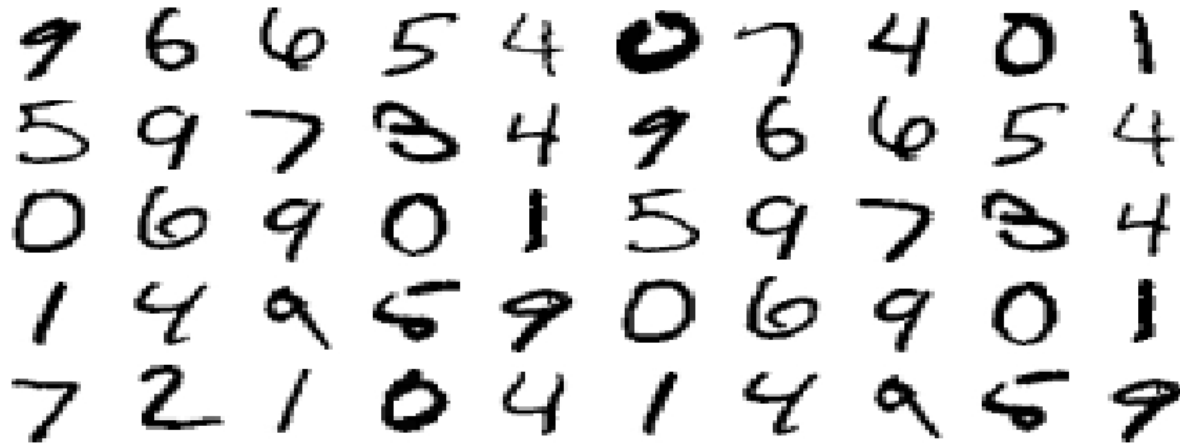


“I still believe that unsupervised learning is going to be crucial, and things will work incredibly much better than they do now when we get that working properly, but we haven't yet. ”

- Geoffrey Hinton, 2017

Homework

- Cluster analysis and dimension reduction on a subset of the MNIST data



Further reading

Text book:

The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Hastie T, Tibshirani R, Friedman J. Springer: 2011. Chapter 14.

PCA:

Shlens J (2003). A tutorial on principal component analysis.

ICA:

Hyvärinen A (1997). Independent Component Analysis by Minimization of Mutual Information.

Lee SI, Batzoglou S (2003). Application of independent component analysis to microarrays. *Genome Biol.* 4(11):R76.

NMF:

Lee DD, Seung HS (1999). Learning the parts of objects by non-negative matrix factorization. *Nature.* 401(6755):788-91.

General Topics:

Zinovyev A, Kairov U, Karpenyuk T, Ramanculov E (2013). Blind source separation methods for deconvolution of complex signals in cancer biology. *Biochem Biophys Res Commun.* 430(3):1182-7.