Melbourne Bioinformatics
BIOINFORMATICS + DATA SERVICES + INFRASTRUCTURE, FOR LIFE SCIENCES TODAY

THE UNIVERSITY OF MELBOURNE

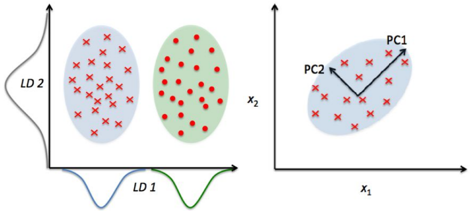
COMP90014

Algorithms for Bioinformatics

Week 9A: Dimensionality Reduction I

Dimensionality Reduction

Why do we need dimensionality reduction?



Feature selection

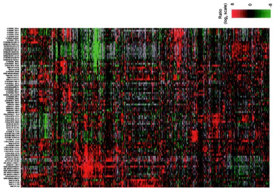
- greedy feature selection

Feature extraction

- unsupervised methods
- supervised methods
- principal component analysis (PCA)
- singular value decomposition (SVD)

Datasets have huge numbers of features (dimensions)

Documents:	thousands of words
Images:	thousands to millions of pixels
Genomics:	thousands of genes, millions of DNA variants



e.g. a gene expression microarray

- 100 samples from *Arabidopsis thaliana*
- data from 27 000 genes
- what is the dimensionality of this dataset?
- 27 000

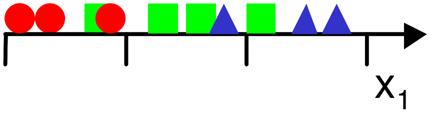
这个数据集包括了来自拟南芥（Arabidopsis thaliana）的100个样本，涵盖了27,000个不同的基因。这个数据集的维度是27,000，因为每个样本都有27,000个基因表达特征。

Wikimedia Commons; Zembutsu *et al.*, 2002. Cancer Res 62 (2): 518–527.

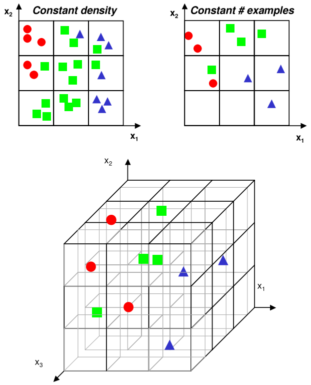
High dimensionality has many costs

- data interpretation and visualisation
- redundant and irrelevant features degrade performance of ML algorithms
- computational cost may become intractable

The curse of dimensionality



当数据的维度（即特征的数量）增加时，数据的分布会变得非常稀疏，这会导致很多机器学习算法的性能下降，因为算法难以从稀疏数据中学习到规律



Ricardo Gutierrez-Osuna, Wright State University

- as the number of dimensions increases the data become ***sparse***
- an huge amount of data is needed to “cover” all the dimensions
  - number of data points needed **grows exponentially with the number of dimensions**

随着维度的增加，数据变得稀疏。
需要大量的数据来“覆盖”所有维度。
需要的数据点数量随着维度的增加而指数级增长。

在一维空间中，数据点可能很紧密；但在二维空间中，同样数量的数据点分布得更开；到了三维空间，数据点则分布得更加稀疏。

Goal of dimensionality reduction

1. transform data from high-dimensional space into low-dimensional space
2. retain meaningful properties of the original data

Австралиец via Wikimedia Commons

Dimensionality Reduction

Two approaches to perform dimensionality reduction

$$\mathbb{R}^N \to \mathbb{R}^M \quad (M < N)$$

- 🍇 the goal is to find a low-dimensional representation of the data
- 🍇 preserving (most of) the information or structure in the data

Feature selection: choosing a subset of all existing features

$$[x_1, x_2, \dots, x_N] \xrightarrow{\text{feature selection}} [x_{i_1}, x_{i_2}, \dots, x_{i_M}]$$

Feature extraction: creating new features by combining existing ones

$$[x_1, x_2, \dots, x_N] \xrightarrow{\text{feature extraction}} [y_1, y_2, \dots, y_M] = \mathrm{f}\left([x_{i_1}, x_{i_2}, \dots, x_{i_m}]\right)$$

Feature selection

Forward Greedy Feature Selection

```
1  $FS^{(0)} = \varnothing; F^{(0)} = \{f_1, f_2, \ldots, f_n\};$
   $i = 0; opt = 0; iter = 0;$
2  while $i < n$ do
3      $k = size\left(F^{(i)}\right)$
4      $max = 0; feature = 0$
5      for $j = 1$ to $k$ do
6          $score = eval\left(F_j^{(i)}\right)$
7          if $score > max$ then
8              $max = score; feature = F_j^{(i)}$
9      if $max > opt$ then
10         $opt = max; iter = i$
11     $FS^{i+1} \leftarrow FS^{(i)} + feature$
12     $F^{i+1} = F^{(i)} - feature$
13     $i++$
```

Alogirithm: Liu & Hong, 2014. 10.1371/journal.pone.0061567

Supervised feature selection

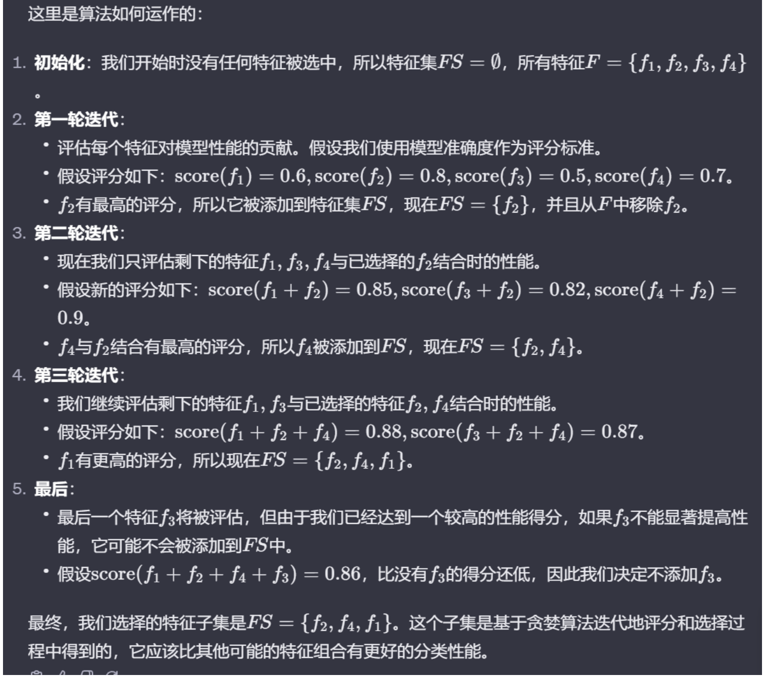
- optimise an objective function
  e.g. F1 metric

Algorithms for supervised feature selection

- Backward Greedy Feature Elimination
- Forward Greedy Feature Selection

Forward Greedy Feature Selection

1. choose the feature with the highest score
2. add feature to the pool
3. re-score other features together with feature in the pool
4. repeat the process

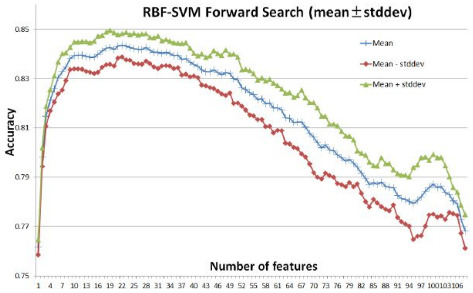


这里是算法如何运作的：

1. **初始化**：我们开始时没有任何特征被选中，所以特征集$FS = \emptyset$，所有特征$F = \{f_1, f_2, f_3, f_4\}$。
2. **第一轮迭代**：
   - 评估每个特征对模型性能的贡献。假设我们使用模型准确度作为评分标准。
   - 假设评分如下：$\text{score}(f_1) = 0.6, \text{score}(f_2) = 0.8, \text{score}(f_3) = 0.5, \text{score}(f_4) = 0.7$。
   - f_2有最高的评分，所以它被添加到特征集FS，现在$FS = \{f_2\}$，并且从F中移除f_2。
3. **第二轮迭代**：
   - 现在我们只评估剩下的特征f_1, f_3, f_4与已选择的f_2结合时的性能。
   - 假设新的评分如下：$\text{score}(f_1 + f_2) = 0.85, \text{score}(f_3 + f_2) = 0.82, \text{score}(f_4 + f_2) = 0.9$。
   - f_4与f_2结合有最高的评分，所以f_4被添加到FS，现在$FS = \{f_2, f_4\}$。
4. **第三轮迭代**：
   - 我们继续评估剩下的特征f_1, f_3与已选择的特征f_2, f_4结合时的性能。
   - 假设评分如下：$\text{score}(f_1 + f_2 + f_4) = 0.88, \text{score}(f_3 + f_2 + f_4) = 0.87$。
   - f_1有更高的评分，所以现在$FS = \{f_2, f_4, f_1\}$。
5. **最后**：
   - 最后一个特征f_3将被评估，但由于我们已经达到一个较高的性能得分，如果f_3不能显著提高性能，它可能不会被添加到FS中。
   - 假设$\text{score}(f_1 + f_2 + f_4 + f_3) = 0.86$，比没有$f_3$的得分还低，因此我们决定不添加$f_3$。

最终，我们选择的特征子集是$FS = \{f_2, f_4, f_1\}$。这个子集是基于贪婪算法迭代地评分和选择过程中得到的，它应该比其他可能的特征组合有更好的分类性能。

Feature selection



$$[x_1, x_2, \ldots, x_N] \xrightarrow{\text{feature selection}} [x_{i_1}, x_{i_2}, \ldots, x_{i_M}]$$

Nguyen & Allebach, 2017. 10.2352/ISSN.2470-1173.2017.12.IQSP-238

Time complexity

- forward / backward: $O(n^2)$
- consider evaluation time: $O(n^2 \times \text{eval})$
- might be unfeasible, if
  - you have a lot of features
  - it takes a long time to evaluate features
- stop after selecting / removing K features
- stop when objective function stops improving

Feature extraction

Define a mapping function $y = \mathrm{f}(x)$ over all features:

$$[x_1, x_2, \ldots, x_N] \xrightarrow{\text{feature extraction}} [y_1, y_2, \ldots, y_M] = \mathrm{f}\left([x_{i_1}, x_{i_2}, \ldots, x_{i_m}]\right)$$

- $y = \mathrm{f}(x)$ is, in general, non-linear and problem-dependent
- *e.g.* $\mathrm{BMI} = \dfrac{\text{weight}}{\text{height}^2}$

身体质量指数（BMI）是通过体重除以身高的平方来计算的，这是一个简单的非线性特征提取示例。

We usually use linear projections $y = Wx$

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} \xrightarrow{\text{linear feature extraction}} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_M \end{bmatrix} = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1N} \\ w_{21} & w_{22} & \cdots & w_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ w_{M1} & w_{M2} & & w_{MN} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix}$$

分类：目标是最大化不同类别之间的分离度。例如，线性判别分析（LDA）就是一种分类算法，它试图找到能够最大化类别分离的特征子空间。
回归：目标是最大化预测数据和目标变量之间的相关性。偏最小二乘回归（PLS）是一种常用的回归技术，它通过最大化因变量和预测变量之间的协方差来提取特征。

Feature extraction

Two criteria to find the best mapping function $y = f(x)$

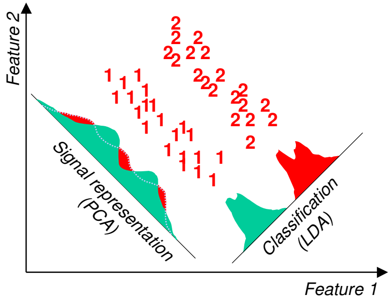
Supervised

Classification: maximize separation among classes
e.g. Linear Discriminant Analysis (LDA)

Regression: maximize correlation between projected data and target variable
e.g. Partial Least Squares (PLS)

Signal representation

Unsupervised: retain as much data variance as possible
Principal Component Analysis (PCA)
Singular Value Decomposition (SVD)



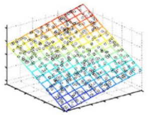
Ricardo Gutierrez-Osuna, Wright State University

Dimensionality Reduction

Principal Component Analysis

- widely used method for unsupervised, linear dimensionality reduction
- find a k-dimensional set of vectors (***components***) such that, if we project our data into this subspace, as much as possible of the variance is retained

Input:	points scattered in multidimensional space
Output:	a more compact representation of the data


$3d \Rightarrow 2d$

Applications:

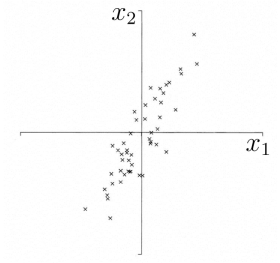
- Data visualisation
- Data compression
- Signal processing
- Assist other learning algorithms

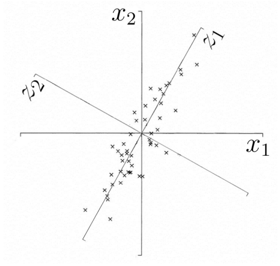
这些向量定义了一个新的子空间，在这个子空间中，数据的投影能够保留尽可能多的原始数据的方差。

基本步骤：

输入：分散在多维空间中的数据点。
输出：数据的更紧凑表示，通常在较低维度的空间中。

PCA





PCA 的目标是将可能存在相关性的多个变量转换成较少数量的不相关变量。这些不相关的变量被称为主成分（Principal Components，PCs），它们是原始数据在新坐标系统中的表示，这些坐标轴是数据方差最大的方向。

Mathematical procedure:

- transform a number of (possibly correlated) variables into a (smaller) number of uncorrelated variables
- the uncorrelated variables are called ***principal components***

Principal components (PC):

- first PC is the projection direction that maximizes the variance of the projected data
- second PC is the projection direction that is orthogonal to the first PC and maximizes variance of the projected data

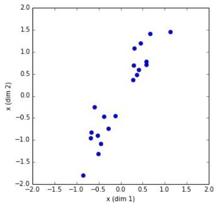
- Spatial rearrangements may reveal relationships that were hidden in higher dimension space

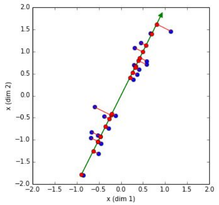
空间重排：

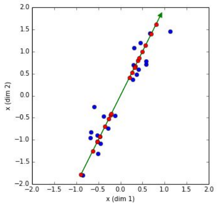
通过在主成分定义的新空间中重新排列数据，可能会揭示在高维空间中隐藏的关系。这是因为PCA有助于去除数据中的噪声和冗余，突出最重要的结构。

PCA concept

寻找一条直线：PCA的目标是在数据空间中找到一条直线，当数据点投影到这条直线上时，其方差达到最大。这条直线的方向就是第一主成分。最大化方差的同时，也意味着最小化数据点到这条直线的垂直（正交）投影误差。







1. Find a line, such that when the data is projected onto that line, it has the **maximum variance**.
   This is the same as **minimising** the orthogonal projection error.

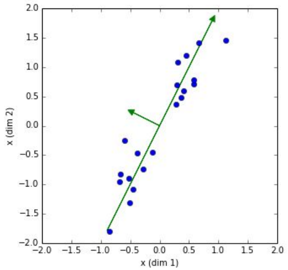
- Projecting the data onto the PC(s) gives us an approximate representation in fewer dimensions.
- Equivalent to rotating the data onto new axes, then discarding some dimensions.

数据投影：将数据点投影到这条直线（主成分）上，可以得到一个在较低维度上的近似表示。这通常涉及到将数据点从原始空间转换到由主成分定义的新空间。
降维操作：这个过程可以理解为首先将数据旋转到新的坐标轴上，然后丢弃一些维度。在这个新的坐标系统中，一些轴（对应于主成分）将捕捉数据的主要变异性，而其余的轴（通常丢弃的）则对应于较小的变异性。

PCA concept

寻找第二条线：在找到第一主成分（即方差最大的方向）之后，接下来要找的是第二条线。这条线必须与第一条线正交（在二维空间中即垂直），同时在这个新的方向上，数据的投影仍然要保持最大的剩余方差。

2. Find another line, orthogonal to the first, that has maximum projected remaining variance.
3. Repeat until we have *k* orthogonal lines (PCs)



- The projected position of a point on the PCs is its coordinates in the *k*-dimensional space

第一主成分（PC1）具有最大的方差，第二主成分（PC2）具有第二大的方差，依此类推。

PCs are ordered

- PC1 has the most variance
- PC2 has the second most, *etc.*

主成分的不相关性：由于主成分是正交的，它们是不相关的。这意味着在新的坐标系统中，数据的各个主成分没有线性关联

Principal components are uncorrelated

- Original data may contain correlated features
- Principal components are orthogonal

重复过程：这个过程将重复进行，直到我们找到了 k 条正交线（即 k 个主成分），每一步都在寻找剩余方差最大的方向，同时确保新找到的方向与之前的所有主成分正交。
数据点的新位置：每个数据点在主成分上的投影位置表示了它在 k 维空间中的坐标。这些坐标构成了数据点在低维空间中的新表示。

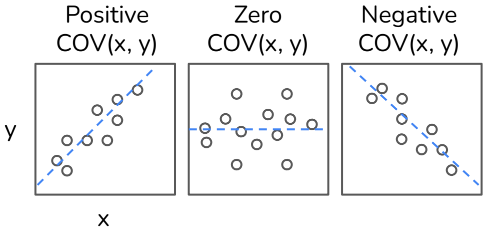
尽管原始数据中的特征可能存在相关性，但通过PCA转换后的主成分却是彼此独立的。

PCA Main Terminology

协方差是度量两个变量之间关系的统计量。
正协方差表示两个变量有相同的趋势，即一个变量增加时，另一个变量也增加。
零协方差表示两个变量之间没有线性关系。
负协方差表示两个变量有相反的趋势，即一个变量增加时，另一个变量减少。

Covariance

- Measure of relationship between variables
- Positive = trend the same
- Zero = no relationship
- Negative = trend inversely



Linear combination

- Multiplying each variable by a scalar
- Eg. 1.2x + 3.4y + ...
- Used by PCA to define new axes which best explain the data in a smaller number of dims

线性组合是通过将每个变量乘以一个标量（即权重），然后将结果相加得到的。例如，

1.2x+3.4y 就是变量
x 和
y 的一个线性组合。
在PCA中，线性组合用于定义新的轴，这些轴能够以较少的维度最好地解释数据。

PCA Main Terminology

(For a given PC)

Eigenvector

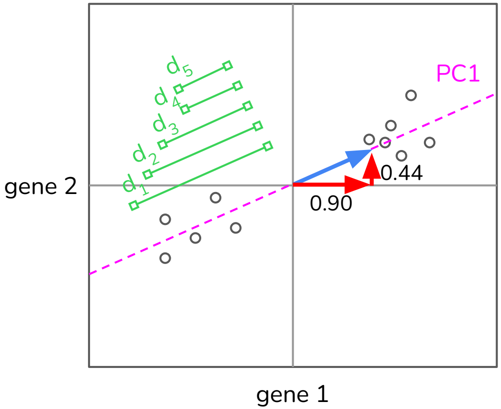
- Defines the line of best fit
- Unit vector (length = 1)
- Also known as singular vector

Loading Scores

- Combination of each variable to create the PC
- 0.9xgene1 + 0.44xgene2
- A linear combination!

Eigenvalue

- Measure of variance explained by this PC
- Average of SS(distances for PC)
- Eigenvalue = $(d_1^2 + d_2^2 + \ldots) / n-1$



PCA algorithm

Input: Data matrix X, integer K
Output: Projected matrix Z

1. Calculate the covariance between each dimension of the data.
2. Find ***eigenvectors*** of the covariance matrix. This:
   a. Finds basis vectors which are uncorrelated
   b. Finds ***eigenvalues***
      i. Give a measure of variance along each eigenvector
3. Choose K: how many principal components to keep
4. $Z \leftarrow$ Project data points onto K basis vectors

🍇 Principal Components chosen are the eigenvectors with largest eigenvalues

$$\mathrm{Cov}_{xy} = \frac{\sum (x - \bar{x})(y - \bar{y})}{n - 1}$$

$$Ax = \lambda x$$

For a matrix A, there is a vector x (eigenvector) such that the product of Ax is equal to a scalar multiplied by x

计算数据每个维度之间的协方差。
找到协方差矩阵的特征向量。这涉及到：
a. 找到互不相关的基向量。
b. 找到特征值。
i. 每个特征向量的方差量度。
选择K：保留多少个主成分。
Z ← 把数据点投影到K个基向量上。

PCA Output

Output

- Data is centered
- Axes are now the PCs

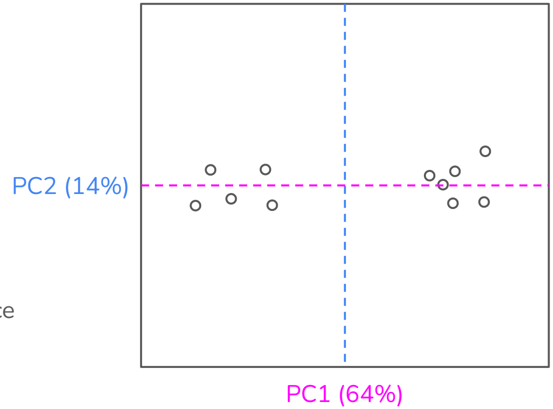
Variation explained by each PC

- Can determine using Eigenvalues
- Eigenvalues: measure of variance
- Eg `PC1=9, PC2=2, ..., Sum=14`
- Variation explained by PC1 = 9/14 = 0.64

- First 2 PCs explain 78% of the total variance
- The higher the better!

How many components will be produced?

- Min(num variables, num samples)
- Whichever is smaller



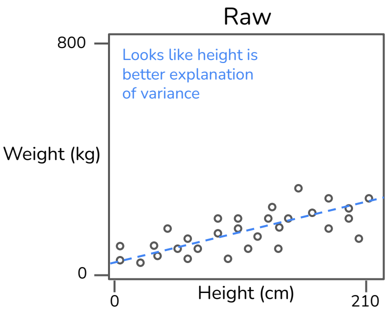
PCA Preprocessing

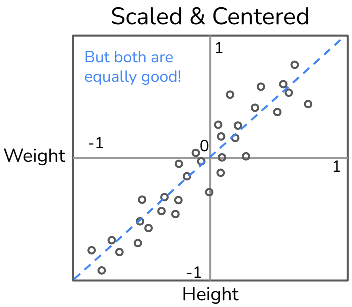
Scaling the data

- Variables should be on same scale
- RNAseq: Counts per million (CPM), Log CPM
- Log scale often useful for biological count data!

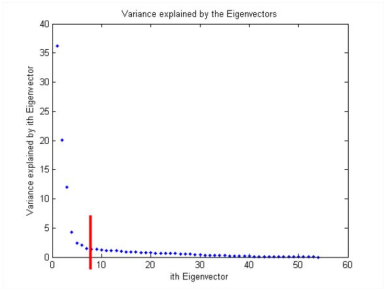
Centering the data

- PCA needs the data to be centered first
- Some libraries will do this for you, others won't
- Imagine centering about the "origin"
- Why? Conceptually, PCA finds best fit lines which go through the origin.





Using PCA



How many components to choose?

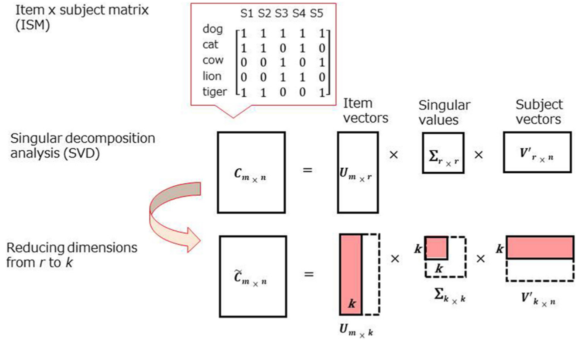
- plot the proportion of variance explained by each eigenvector
  - eigenvalue divided by the sum of eigenvalues
- *e.g.* use eigenvectors that cover at least X % of the variance

Time complexity (n dimensions and m observations)

- covariance matrix computation is $O(n^2m)$
- eigenvalue decomposition is $O(n^3)$
- total for PCA is $O(n^2m + n^3)$

Alternatives?

Singular Value Decomposition (SVD)


Item x subject matrix (ISM)
S1 S2 S3 S4 S5
dog 1 1 1 1 1
cat 1 1 0 1 0
cow 0 0 1 0 1
lion 0 0 1 1 0
tiger 1 1 0 0 1
Singular decomposition analysis (SVD)
Item vectors
Singular values
Subject vectors
$C_{m\times n} = U_{m\times r} \times \Sigma_{r\times r} \times V'_{r\times n}$
Reducing dimensions from r to k
$\tilde{C}_{m\times n} = U_{m\times k} \times \Sigma_{k\times k} \times V'_{k\times n}$
k

- data matrix C
- factors U, Σ and V^T
  - Σ (diagonal matrix) with singular values (σ)
  - singular values (σ) are the square roots of the eigenvalues
- U and V^T are orthogonal eigenvectors
- can use SVD to calculate PCA

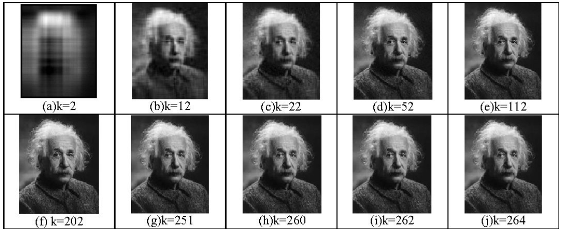
奇异值分解（Singular Value Decomposition，简称SVD）是一种矩阵分解方法，它可以用来降维、特征提取和数据压缩等应用。在SVD中，我们有一个数据矩阵C，
通过将其分解为三个矩阵U、Σ和V^T的乘积，可以获得有关数据的重要信息。

Sumiyoshi *et al.*, 2018. 10.3389/fpsyt.2018.00087.

C = U * Σ * V^T
Σ是一个m行n列的对角矩阵，对角线上的元素称为奇异值（σ）
奇异值是特征值的平方根

PCA and SVD

主成分分析（PCA）和奇异值分解（SVD）之间有着密切的关系，可以将PCA解释为在中心化协方差矩阵上进行的SVD。

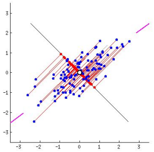


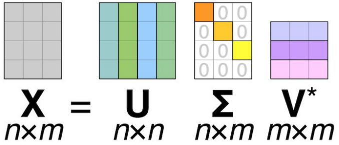
- can interpret PCA as a SVD on the centered covariance matrix
- SVD doesn't require you to calculate covariance matrix: more efficient

SVD不需要显式计算协方差矩阵，因此在某些情况下更高效。

Linear projections: PCA and SVD

线性投影是数据分析和降维技术中的重要概念，而主成分分析（PCA）和奇异值分解（SVD）都涉及线性投影，但它们的方法略有不同。





PCA旨在找到一组正交向量（主成分），以最大程度地表示原始数据中的方差

PCA: Finds orthogonal vectors (components) to represent as much variance is as possible. Decomposing the covariance matrix, $C = W \Sigma W^T$. See this animation.

SVD: More efficient
Can interpret PCA as a SVD on the centered covariance matrix

- StatQuest: Principal Component Analysis (20 minute video from Josh Starmer)

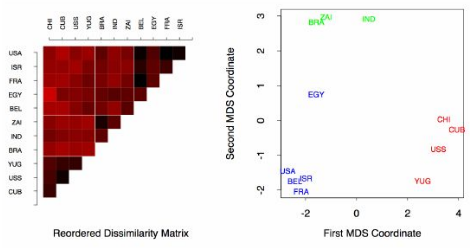
SVD不需要显式计算协方差矩阵，因此在某些情况下更高效

amoeba via stackexchange; Cmglee via Wikimedia Commons

Dimensionality Reduction

1. Why do we need dimensionality reduction?
2. Approaches for dimensionality reduction
3. Principal Component Analysis (PCA)
4. Multi-dimensional scaling (MDS)

Multi-Dimensional Scaling (MDS)

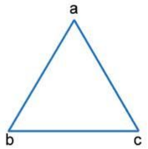


- feature extraction
- produces a lower-dimensional representation that ***preserves pairwise distances or dissimilarities***
- for visualising data
- sometimes called Principal Coordinates Analysis, but it’s **not the same as PCA**!

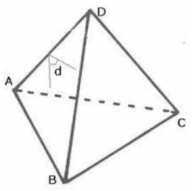
特征提取：MDS通常用于特征提取，即从高维数据中提取出最重要的信息。它通过创建一个低维表示来实现这一目标，该表示尽量保持原始数据点之间的关系。这对于降低数据的复杂性和噪音有很大帮助。

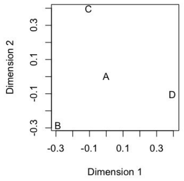
MDS

$$D(x_i, x_j) = \begin{array}{c|ccc} & a & b & c \\ \hline a & 0 & 1 & 1 \\ b & 1 & 0 & 1 \\ c & 1 & 1 & 0 \end{array}$$



$$D(x_i, x_j) = \begin{array}{c|cccc} & a & b & c & d \\ \hline a & 0 & 1 & 1 & 1 \\ b & 1 & 0 & 1 & 1 \\ c & 1 & 1 & 0 & 1 \\ d & 1 & 1 & 1 & 0 \end{array}$$





- start with a pairwise distance matrix or dissimilarity matrix
- we can represent three points that are equally-spaced in 3D **exactly in 2D**
- we can represent four points that are equally-spaced in 3D **exactly in 3D** ...
- ... **but not in 2D**
- in general, we need $N - 1$ dimensions to exactly represent pairwise distances between N samples

幻灯片最后指出，一般来说，要准确表示 N 个样本之间的成对距离，我们需要 N-1 维。这意味着，要完美地在低维空间中表示这些点，所需的最小维数比点的数量少一。对于三个点，我们可以在二维空间中完美地表示它们的距离；对于四个点，则需要三维空间来做到这一点。然而，如果我们试图在二维空间中表示四个点，就无法完美地做到这一点。右下角的图展示了在二维空间中对四个点进行 MDS 分析的结果，其中点 D 不能被放置在与其他三个点等距离的位置上

Types of MDS

度量型 MDS 的目标是最小化“应力”（stress），这里的“应力”是指在低维空间中重构的距离与原始高维空间中的距离之间的误差，也就是不拟合度量（lack-of-fit measure）
保持每个向量xi xj
之间的距离尽可能接近原始高维空间中的距离是很重要的。
应力的计算公式是通过所有点对（i，j）的实际距离
与在低维空间中重构的距离 之间的差异的平方和来定义的。

Metric (distance-based) MDS

- Minimise ***stress***
- Stress is the error in the distances (lack-of-fit measure)
- Try to preserve distance between each vector x_i, x_j

$$\text{Cost} = \sum_{i<j} \left(d_{ij} - \hat{d}_{ij}\right)^2$$

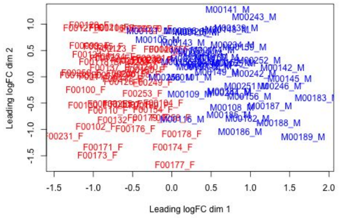
- actual cost/stress used may be more complex

Non-metric MDS

- Try to maintain original ranking of pairwise distances

非度量型 MDS 则是尝试保持原始成对距离的等级顺序，而不是距离本身。它不直接在乎重构的距离与原始距离的具体数值，而是在乎这些距离的相对大小关系是否得到保持。

Interpreting low-dimensionality visualisation



- clusters may represent real clusters
- outliers may represent real outliers
- the position of points is ***not useful*** in MDS
- resulting dimensions are not ordered by importance/variance
- can’t reconstruct original data

簇可能代表真实的簇：在图中形成的紧密点群可能代表数据中实际存在的分组或类别。
离群点可能代表真实的离群点：远离主要簇的单个点可能是实际数据中的异常值。
点的位置在 MDS 中不是有用的：在 MDS 可视化中，点的绝对位置不像在其他类型的维度缩减技术（如主成分分析）中那样重要，因为 MDS 更注重点之间的相对距离。
结果维度不是按重要性/方差排序的：与主成分分析不同，MDS 的维度不是按照解释数据变异性的能力排序的。
无法重构原始数据：MDS 图展示的是原始高维数据在低维空间的一种表示，但从这个低维表示中无法完全恢复出原始数据的所有维度和细节。

Melbourne Bioinformatics

BIOINFORMATICS + DATA SERVICES + INFRASTRUCTURE, FOR LIFE SCIENCES TODAY

THE UNIVERSITY OF MELBOURNE

Thank you!

Today: Dimensionality Reduction I

Next time: Dimensionality Reduction II