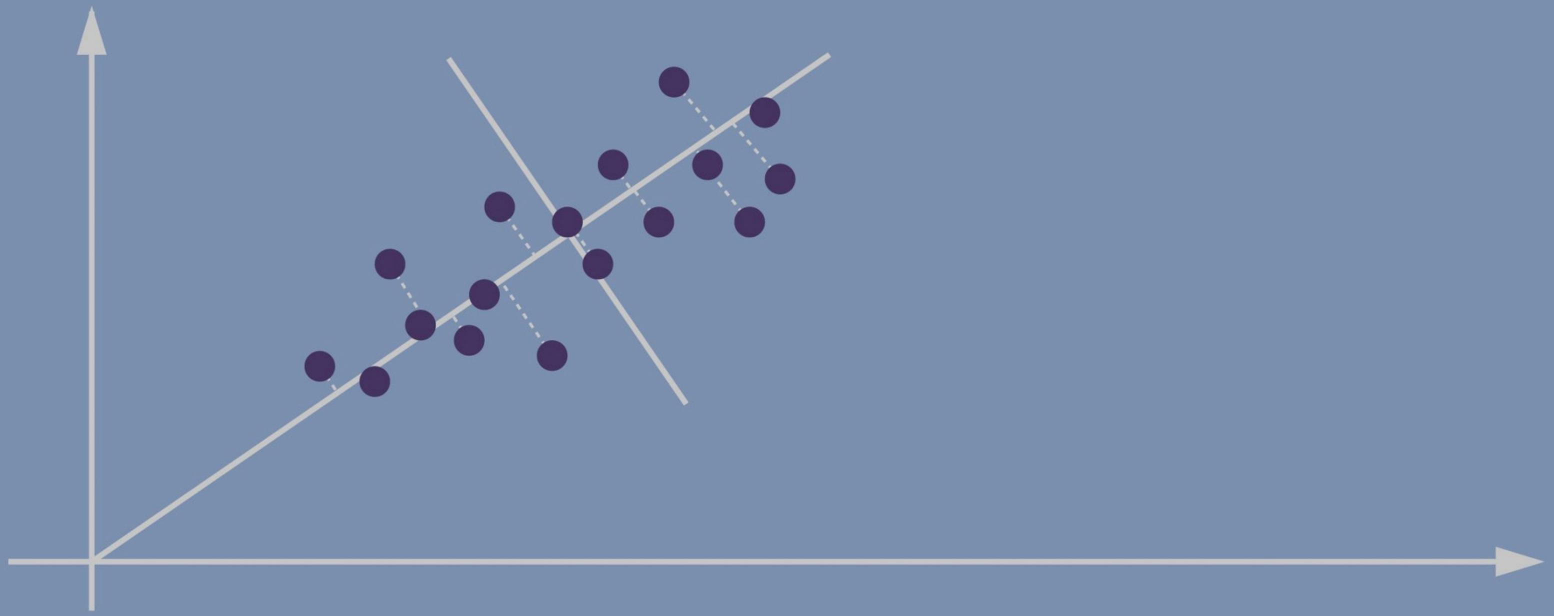
Welcome

Announcements:

- The Mock Test has been released on Blackboard
- A new Stanage reservation for completing the assignment `com6012-11` will be available from 1pm on 26 April to 1pm 3rd May
  - This DOES NOT mean you should wait until 26 April to start on your assignment
  - Use your university account to access assignment
- Please fill out the TellUS survey for this module



PCA-06-scaled.jpg (2560×1051) (perfectial.com)

Lecture 9: Scalable PCA for Dimensionality Reduction

COM6012: Scalable ML with Robert Loftin

Slides courtesy of Haiping Lu

Week 9 Contents / Objectives

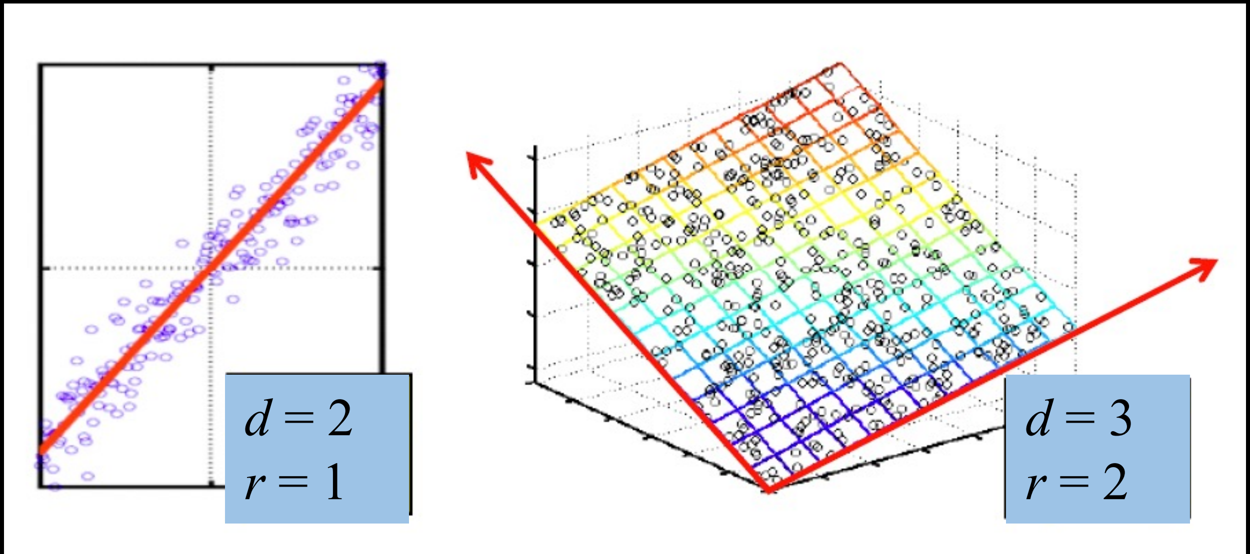
- Principal Component Analysis
- Singular Value Decomposition (SVD)
- PCA via SVD
- Scalable PCA in Spark

Week 9 Contents / Objectives

- Principal Component Analysis
- Singular Value Decomposition (SVD)
- PCA via SVD
- Scalable PCA in Spark

Dimensionality Reduction

- Raw data: complex and high-dimensional
- **Assumption**: data lie on a low-dimensional subspace
  - Axes of this subspace → representation of the data
  - Simpler, more compact, showing interesting patterns

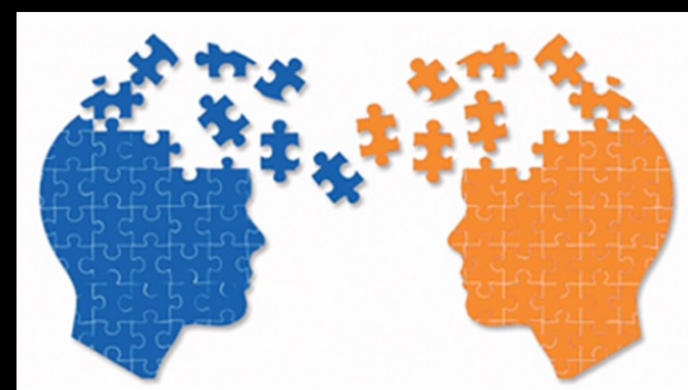


Uses of Dimensionality Reduction

- Discover hidden correlations/topics
- Remove redundant/noisy features
- Interpretation and visualisation
- Easier storage and processing of the data

owners-icebergs-blog-image-300x300.jpg (resettogrow.com)

1*KvKIx9OnlxdoTfNxWKAY_g.jpeg (480×320) (medium.com)



Interpreting and Translation Blog: Image (wordpress.com)

Principal Component Analysis

- Input: n data points in a d-dimensional feature space
  - $\mathrm{X}_0 \leftarrow n \times d$ data matrix, data point → row vector x_i
  - "Centered" data X – mean of each column is zero
- Goal: Find a feature transformation W $(d \times r)$ such that $\mathrm{T} = \mathrm{X}\,\mathrm{W}$ preserves important information
- Idea: Find W that explains most of the variance of X
  - The first principal component w_1 maximizes variance
  - kth PC w_k maximizes variance after subtracting the variance explained by the first $k-1$ principal components

PCA → Variance Maximisation

- The first principal component w_1 maximises the variance of the transformed data $\mathrm{X}\mathrm{w}_1$
- Mean of X is zero, so we can find w_1 by computing

$$\mathrm{w}_1 \in \underset{\mathrm{w}}{\operatorname{argmax}} \frac{\mathrm{w}^{\mathrm{T}}\mathrm{X}^{\mathrm{T}}\mathrm{X}\mathrm{w}}{\|\mathrm{w}\|_2^2}$$

- It turns out, w_1 is an eigenvector corresponding to the largest eigenvalue of $\mathrm{X}^{\mathrm{T}}\mathrm{X}$

Principal Component Analysis

- Input: n data points in a d-dimensional feature space
  - $\mathrm{X}_0 \leftarrow n \times d$ data matrix, data point → row vector x_i
- Basic PCA algorithm
  - X: subtract mean x from each row vector x_i in X_0
  - $\mathrm{X}^\mathrm{T}\mathrm{X}$: Gramian/scatter matrix for X
  - Find eigenvectors and eigenvalues of $\mathrm{X}^\mathrm{T}\mathrm{X}$
  - W ($d \times r$) ← the top r eigenvectors (PCs)
- PCA features $\mathrm{y}_\mathrm{i} = \mathrm{x}_\mathrm{i}^\mathrm{T}\mathrm{W}$ (dimension: $d \rightarrow r$)
  - Zero correlation, ordered by variance

Scalability Problems with PCA

- Input dimensionality → scatter matrix
  - Images: $100 \times 100 \rightarrow 10^4$; $1000 \times 1000 \rightarrow 10^6$
  - Scatter matrix $X^T X$ is of size d^2
    - $d = 10^4 \rightarrow X^T X$ is of size 10^8
    - $d = 10^6 \rightarrow X^T X$ is of size $= 10^{12}$

- Computing all k eigenvectors of $X^T X$ takes $O(d^3)$
- Alternative: Singular Value Decomposition (SVD)
  - Efficient algorithms available
  - Often need just top r eigenvectors

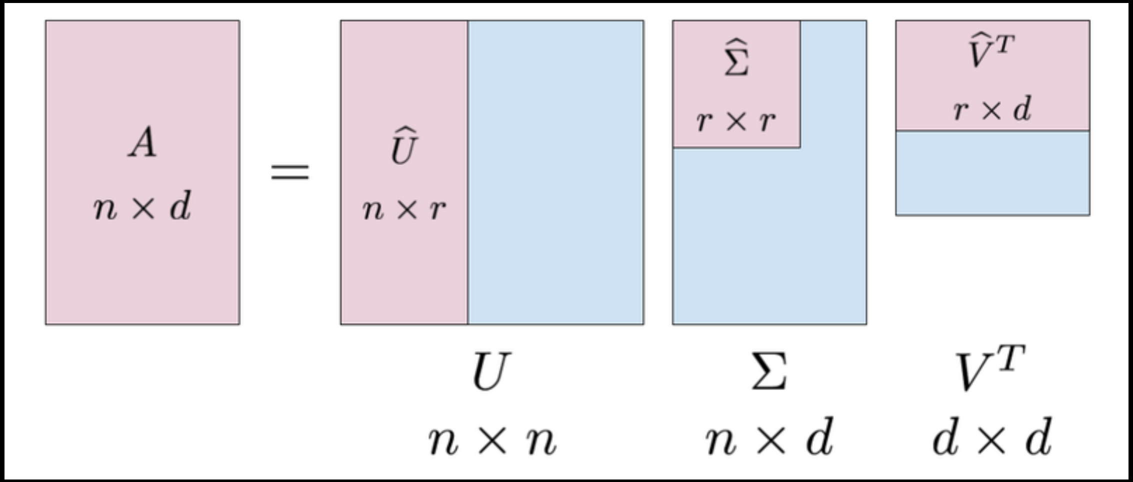
Week 9 Contents / Objectives

- Principal Component Analysis
- Singular Value Decomposition (SVD)
- PCA via SVD
- Scalable PCA in Spark

Singular Value Decomposition (SVD)

$$A_{[n \times d]} = U_{[n \times r]} \Sigma_{[r \times r]} (V_{[d \times r]})^T$$

- r: the rank of the matrix A
- U: $n \times r$ matrix, column orthonormal, $U^TU = I$
- Σ : $r \times r$ diagonal matrix, strength of each factor
- V: $d \times r$ matrix, column orthonormal, $V^TV = I$



svd-matrices.png (800×339) (intoli.com)

Example on a Document x Term

Term Document	data	information	retrieval	brain	lung
CS-TR1	1	1	1	0	0
CS-TR2	2	2	2	0	0
CS-TR3	1	1	1	0	0
CS-TR4	5	5	5	0	0
MED-TR1	0	0	0	2	2
MED-TR2	0	0	0	3	3
MED-TR3	0	0	0	1	1

- d = 5 but r=2 → two bases [1 1 1 0 0] & [0 0 0 1 1]
- U: document-to-concept similarity matrix
- V: term-to-concept similarity matrix
- Σ: its diagonal elements → strength of each concept

Interpretation

Term Document	data	information	retrieval	brain	lung
CS-TR1	1	1	1	0	0
CS-TR2	2	2	2	0	0
CS-TR3	1	1	1	0	0
CS-TR4	5	5	5	0	0
MED-TR1	0	0	0	2	2
MED-TR2	0	0	0	3	3
MED-TR3	0	0	0	1	1

doc-to-concept similarity matrix

CS-concept

MD-concept

strength of CS-concept

CS-concept

term-to-concept similarity matrix

$$
\begin{array}{cc}
 & \begin{array}{ccccc} \text{data} & \text{inf.} & \text{retrieval} & \text{brain} & \text{lung} \end{array} \\
\begin{array}{c} \text{CS} \\ \\ \\ \\ \text{MD} \\ \\ \\ \end{array} & \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 & 3 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}
\end{array}
=
\begin{bmatrix} 0.18 & 0 \\ 0.36 & 0 \\ 0.18 & 0 \\ 0.90 & 0 \\ 0 & 0.53 \\ 0 & 0.80 \\ 0 & 0.27 \end{bmatrix}
\times
\begin{bmatrix} 9.64 & 0 \\ 0 & 5.29 \end{bmatrix}
\times
\begin{bmatrix} 0.58 & 0.58 & 0.58 & 0 & 0 \\ 0 & 0 & 0 & 0.71 & 0.71 \end{bmatrix}
$$

SVD – Dimensionality Reduction

- To reduce the dimensionality further (3 zero singular values have already been removed)
  - Best rank-1 approximation →

$$
\begin{bmatrix}
1 & 1 & 1 & 0 & 0 \\
2 & 2 & 2 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 \\
5 & 5 & 5 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0
\end{bmatrix}
$$

$$
\begin{bmatrix}
1 & 1 & 1 & 0 & 0 \\
2 & 2 & 2 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 \\
5 & 5 & 5 & 0 & 0 \\
0 & 0 & 0 & 2 & 2 \\
0 & 0 & 0 & 3 & 3 \\
0 & 0 & 0 & 1 & 1
\end{bmatrix}
=
\begin{bmatrix}
0.18 & \cancel{0} \\
0.36 & \cancel{0} \\
0.18 & \cancel{0} \\
0.90 & \cancel{0} \\
0 & \cancel{0.53} \\
0 & \cancel{0.80} \\
0 & \cancel{0.27}
\end{bmatrix}
\text{x}
\begin{bmatrix}
9.64 & 0 \\
0 & \cancel{5.29}
\end{bmatrix}
\text{x}
\begin{bmatrix}
0.58 & 0.58 & 0.58 & 0 & 0 \\
\cancel{0} & \cancel{0} & \cancel{0} & \cancel{0.71} & \cancel{0.71}
\end{bmatrix}
$$

Week 9 Contents / Objectives

- Principal Component Analysis
- Singular Value Decomposition (SVD)
- PCA via SVD
- Scalable PCA in Spark

SVD ←→Eigen-decomposition

- SVD of $X = U \Sigma V^T$
- Eigen-decomposition of $X^TX = W \Lambda W^T$
  - Because X^TX is *real* and *symmetric*
- U, V: orthonormal → $U^TU = I$, $V^TV = I$
- Σ, Λ: diagonal
- Relationship:
  - $X^TX = V \Sigma^T U^T (U \Sigma V^T) = V \Sigma \Sigma^T V^T = V \Sigma^2 V^T$
  - $X^TXV = (V \Sigma^2 V^T)V = V\Sigma^2$
- Columns of V are eigenvectors of X^TX ($W = V$)
  - Singular values are square roots of eigenvalues ($\Lambda = \Sigma^2$)

PCA via SVD

- Better PCA algorithm:
  - $X_0 \leftarrow n \times d$ data matrix, data point $\rightarrow$ row vector x_i
  - X: subtract mean x from each row vector x_i in X_0
  - $U\,\Sigma\,V^T \leftarrow$ SVD of X
  - Compute top r right singular vectors V of X $\rightarrow$ the PCs
  - The singular values in Σ = the square roots of the eigenvalues of X^TX

- We can do this without computing the full eigen-decomposition of X^TX

Week 9 Contents / Objectives

- Principal Component Analysis
- Singular Value Decomposition (SVD)
- PCA via SVD
- Scalable PCA in Spark

Three PCA APIs in Spark

- DataFrame-based API – PCA (source code, Scala doc)
  - `pyspark.ml.feature.PCA(k=None, inputCol=None, outputCol=None)`
- RDD-based API – RowMatrix (source code, Scala doc)
  - `computePrincipalComponents(k)`
  - **Scalable**: `computeSVD(k, computeU=False, rCond=1e-09)`

```scala
  @Since("1.6.0")
  def computePrincipalComponentsAndExplainedVariance(k: Int): (Matrix, Vector) = {
    val n = numCols().toInt
    require(k > 0 && k <= n, s"k = $k out of range (0, n = $n]")

    if (n > 65535) {
      val svd = computeSVD(k)
      val s = svd.s.toArray.map(eigValue => eigValue * eigValue / (n - 1))
      val eigenSum = s.sum
      val explainedVariance = s.map(_ / eigenSum)
```

SVD in Spark MLlib (RDD)

- U: $m \times k$; $\Sigma : k \times k$; V: $n \times k$
- Assumption: n (dimensionality) < m (# samples)
- Different methods based on computational cost:
  - If n is small (n<100) or k is large compared with n (k>n/2):
    - Construct $\mathrm{X}^{\mathrm{T}}\mathrm{X}$ first, then compute its top eigenvalues and eigenvectors locally on the driver node
  - Otherwise:
    - Run ARPACK on the driver node to compute eigenvalues/eigenvectors
    - ARPACK makes calls to Spark to compute $(\mathrm{X}^{\mathrm{T}}\mathrm{X})v$ – for different vectors v – which in Spark computes in a distributed wa

Selection of SVD Computation

```scala
      if (n < 100 || (k > n / 2 && n <= 15000)) {
        // If n is small or k is large compared with n, we better compute the Gramian matrix first
        // and then compute its eigenvalues locally, instead of making multiple passes.
        if (k < n / 3) {
          SVDMode.LocalARPACK
        } else {
          SVDMode.LocalLAPACK
        }
      } else {
        // If k is small compared with n, we use ARPACK with distributed multiplication.
        SVDMode.DistARPACK
      }
    case "local-svd" => SVDMode.LocalLAPACK
    case "local-eigs" => SVDMode.LocalARPACK
    case "dist-eigs" => SVDMode.DistARPACK
    case _ => throw new IllegalArgumentException(s"Do not support mode $mode.")
```

Acknowledgement & References

- Acknowledgement
  - Some slides are adapted from the [MMDS book] slides
- References
  - [Chapter 11] of the [MMDS book]

Thank You

Announcements:

- The Mock Test has been released on Blackboard
- A new Stanage reservation for completing the assignment `com6012-11` will be available from 1pm on 26 April to 1pm 3rd May
  - This DOES NOT mean you should wait until 26 April to start on your assignment
  - Use your university account to access assignment
- Please fill out the TellUS survey for this module