2.3 Featurereduktion

Idee: Anstatt Features einfach wegzulassen, generiere einen neuen niedrigdimensionalen Featureraum aus allen Features:

- Redundante Features können zusammengefasst werden
- Irrelevantere Features haben einen entsprechend kleineres Gewicht in den neuen Feature

Lösungsansätze:

- Referenzpunktansatz
- Hauptkomponentenanalyse (PCA)
- Single-Value-Decompusition (SVD)

1. Referenzpunkt Transformation

Idee:

Position eines Objekts kann häufig recht gut über den Abstand zu anderen Objekten beschrieben werden.

Wähle k Referenzpunkte und beschreibe Objekte über den k-dimensionalen Vektor der Abstände zu den Referenzpunkten.

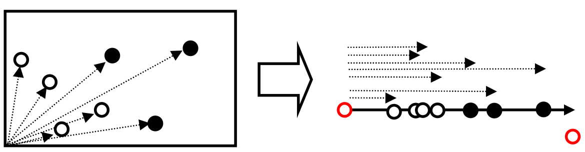
Gegeben: Vektorraum $F = D_1 \times .. \times D_n$ mit $D = \{D_{1},..,D_n\}$.

Gesucht: *k*-dimensionaler Raum *R*, der für ein gegebenes Data Mining Problem eine optimale Lösung erlaubt.

Methode: Für die Menge der Referenzpunkte $R = \{r_{1},..,r_k\}$ und Distanzmaß *d()*:.

Transformation von Vektor $x \in F$:

$$r = \begin{pmatrix} d(r_1, x) \\ \vdots \\ d(r_k, x) \end{pmatrix}$$



1. Referenz Punkt Transformation

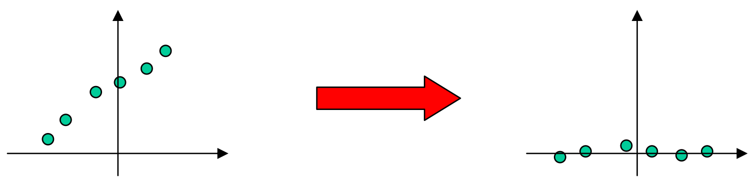
- Abstandsmaß ist meist durch Applikation gegeben.
- Auswahl der Referenzpunkte:
  - Wähle Centroide der Klassen oder Cluster-Centroide als Referenzpunkte
  - Häufig Wahl der Referenzpunkte am Rand und möglichst weit weg von allen Datenobjekten.

Vorteile:
- leicht umzusetzender Ansatz

Nachteil:
- selbst bei gleicher Featureanzahl ist die Abbildung nicht eindeutig
- Performanz stark von der Wahl der Referenzpunkte abhängig.

Hauptachsentransformation (PCA)

Ziel: Rotiere den Datenraum so, dass
- die Abhängigkeiten zwischen den Merkmalen verschwinden
- Abstände und Winkel der Vektoren erhalten bleiben



Gesucht ist also...
- eine orthonormale Abbildung,
- die die Richtung stärkster Varianz auf die erste Achse abbildet
- die Richtung zweitstärkster Varianz auf die zweite usw.

PCA

- Wir beginnen mit der Kovarianz-Matrix: $\Sigma = 1/n \sum_{x \in D}(x-\mu)(x-\mu)^T$
- Die Matrix wird zerlegt in
  - eine Orthonormalmatrix $V = [e_1,\ldots,e_d]$ (Eigenvektoren)
  - und eine Diagonalmatrix $\Lambda = \mathrm{diag}\,(\lambda_1,\ldots,\lambda_d)$ (Eigenwerte)
  - so dass gilt: $\Sigma = V \Lambda V^T$
- Bei Weglassen von k Basisvektoren e_j entsteht ein neuer Unterraum. Die Transformation der Vektoren aus X in diesen neuen Unterraum hat den quadratischen Fehler:

$$\chi^2 = \sum_{j=1}^{k} \lambda_j$$

$\Rightarrow$ Wähle die k Eigenvek leinsten Eigenwerten

PCA

Dimensionsreduktion via PCA

1. Berechne Kovarianzmatrix Σ
2. Berechne Eigenwerte und Eigenvektoren von Σ
3. Bestimme die k kleinsten Eigenwerte und lösche deren Eigenvektoren (V')
4. Die resultierenden Eigenvektoren bilden die Basis für den neuen Unterraum
5. Entwickle die Vektoren der Daten $X = \{x_1,\ldots,x_n\}$ nach dieser neuen Unterraumbasis:

$$y_i = V' \, x_i$$

$\Rightarrow$ Resultierende Daten $Y = \{y_1,\ldots,y_n\}$ sind $(d-k)$-dimensional

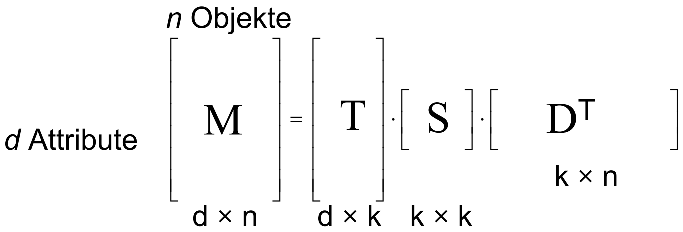
Single Value Decomposition (SVD)

Verallgemeinerung der PCA: Auch anwendbar wenn Kovarianz-Matrix singulär.

Im Textumfeld häufig als Latent Semantic Indexing (LSI) bezeichnet.

Grundidee:

Bestimme Zerlegung der Objekt-Feature-Matrix.



T : links-singuläre Vektoren, orthogonal
S : singuläre Werte, Diagonalmatrix
D : rechts-singuläre Vektoren, orthogonal

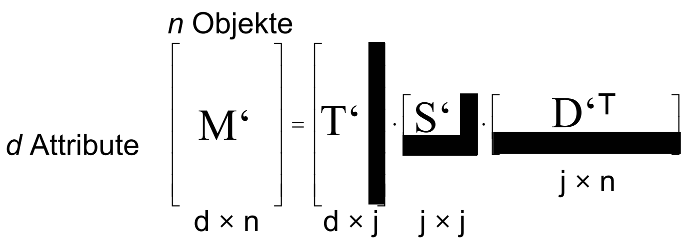
Zerlegung mittels numerischer Algorithmen zur Matrix-Faktorisierung.

(nicht Thema der Vorlesung)

SVD (2)

Feature-Reduktion auf $j < k$ Features:

- Sortiere TSD^T nach der größe der k Diagonaleinträge in S
- Streiche die k-j Zeilen mit den niedrigsten singulären Werten



- M‘ ist Näherung von M
- Anstatt der A Attribute werden die Objekte im Raum j-dimensionalen Raum der Singular Values betrachtet => die Matrix D‘ beschreibt die Objekte.

SVD(3)

Problem: Momentan sind nur Trainingsdaten umgewandelt.

Wie werden neue Objekte in den neuen Feature-Raum transformiert ?

Lösungsansatz: Folding-In

Neues Objekte o.

Durch Umformung erhält man aus M‘ =T‘ S‘ D‘^{T} folgende Umrechnung:

$$M' = T' \cdot S' \cdot D'^T \Rightarrow S'^T \cdot T'^T \cdot M' = D'^T$$

$$\boxed{S'^T \cdot T'^T \cdot o = r}$$ Umrechnungsformel

Vorteile:

- Anwendbar auf alle möglichen Objekt-Attribut Matrizen
- Reduktion auf die wichtigsten Konzepte

Nachteile:

- Matrixfakturisierung ist ein verhältnismäßig teures Verfahren
- basiert auf linearen Abbildungen zwischen Attributen und Objekten

Literatur

- A.Blum and P. Langley: *Selection of Relevant Features and Examples in Machine Learning*, Artificial Intelligence (97),1997
- H. Liu and L. Yu: *Feature Selection for Data Mining* (WWW), 2002
- L.C. Molina, L. Belanche, Â. Nebot: Feature Selection Algorithms: *A Survey and Experimental Evaluations*, ICDM 2002, Maebashi City, Japan
- P. Mitra, C.A. Murthy and S.K. Pal: *Unsupervised Feature Selection using Feature Similarity*, IEEE Transacitons on pattern analysis and Machicne intelligence, Vol. 24. No. 3, 2004
- S. Deerwester, S. Dumais, R. Harshman: *Indexing by Latent Semantic Analysis,* Journal of the American Society of Information Science,Vol. 41, 1990
- J. Dy, C. Brodley: *Feature Selection for Unsupervised Learning*, Journal of Machine Learning Research 5, 2004
- I. Guyon, A. Elisseeff: An Introduction to Variable and Feature Selection, Journal of Machine Learning Research 3, 2003
- M. Dash, H. Liu, H. Motoda*: Consistency Based Feature Selection*, 4th Pacific-Asia Conference, PADKK 2000, Kyoto, Japan, 2000