Dimensionality Reduction and Subspace Clustering: Example for the Expert-in-the-Loop

a.holzinger@hci-kdd.org
http://hci-kdd.org/machine-learning-for-health-informatics-course
1) Classification vs Clustering
2) Feature spaces, feature engineering
   - Feature selection, feature extraction
3) The curse of dimensionality
4) Dimensionality reduction
   - PCA, ICA, FA, MDS, LDA – Isomap, LLE, Autoencoder
5) Subspace clustering and analysis
6) Projection Pursuit: “What is interesting?”
Always with a focus/application in health informatics

- **CONCEPTS**
  - Curse of Dim
  - NfL-Theorem
  - Overfitting
  - KL-Divergence
  - Non-Parametric
  - Exp. & Eval.

- **THEORIES**
  - Bayesian p(x)
  - Complexity
  - Info Theory

- **PARADIGMS**
  - unsupervised
  - supervised
  - Semi-supv.
  - online

- **MODELS**
  - Gaussian P.
  - Graphical M.
  - NN
  - SVM
  - Linear Models

- **METHODS**
  - Regularization
  - Validation
  - Aggregation
  - iML

- **TOOLS**
  - Python
  - Julia
  - Etc.
  - Azure
  - Privacy ML

- **Maths**

- **Cognition**

- **Visualization**

- **Data structure**

- **Perception**

- **Preprocessing**

- **Decision**

- **Interaction**

- **Integration**

- **Challenges**
Key Challenges

- Uncertainty, Validation, Curse of Dimensionality
- Large spaces gets sparse
- Distance Measures get useless
- Patterns occur in different subspaces
- “What is interesting?”
1) Classification vs. Clustering
1) The data is not labeled (A/C)?
2) Identify structure/patterns (A/C)?
3) Predicting an item set, identifying to which set of categories a new observation belongs (A/C)?
4) Assigning a set of objects into groups (A/C)?
5) Having many labelled data points (A/C)?
6) Using the concept of supervised learning (A/C)?
7) Grouping data items close to each other (A/C)?
8) Used to explore data sets (A/C)?
Classification vs. Clustering

- **Classification** (Supervised learning, Pattern Recogn., Prediction)
  - Supervision = the training data (observations, measurements, etc.) are accompanied by labels indicating the class of the observations;
  - New data is **classified based on the training set**
  - Important for clinical decision making
  - Example: Benign/Malign Classification of Tumors

- **Clustering** (Unsupervised learning, class discovery, )
  - The class labels of training data is unknown
  - Given a set of measurements, observations, etc. with the aim of **establishing the existence of clusters** in the data;
Why do we need Classification?

Typical questions include:
- Is this protein functioning as an enzyme?
- Does this gene sequence contain a splice site?
- Is this melanoma malign?

Given object \( x \) – predict the class label \( y \):
- If \( y \in \{0,1\} \) → binary classification problem
- If \( y \in \{1, \ldots, n\} \) and is \( n \in \mathbb{N} \) → multiclass problem
- If \( y \in \mathbb{R} \) → regression problem

- Naïve Bayes (NB) – see Bayes’ theorem with independent assumptions (hence “naïve”)
- Decision Trees (e.g. C4.5)
- NN – if $x_1$ is most similar to $x_2 \implies y_1 = y_2$

$$x_j = \arg \min_{x \in D} \|x - x_i\|^2 \implies y_i = y_j$$

- SVM – a plane/hyperplane separates two classes of data – very versatile for classification and clustering – also via the Kernel trick in high-dimensions

- Uses a **nonlinear mapping** to transform the original data (input space) **into a higher dimension** (feature space).

- = classification method for both **linear and nonlinear data**;
- Within the new dimension, it searches for the linear optimal separating **hyperplane** (i.e., “decision boundary”);
- By nonlinear mapping to a sufficiently high dimension, data from two classes can always be separated with a hyperplane;
- The SVM finds this hyperplane by using **support vectors** (these are the “essential” training tuples) and **margins** (defined by the support vectors);
**SVM**
- Deterministic algorithm
- Nice generalization properties
- Hard to learn – learned in batch mode using quadratic programming techniques
- Using kernels can learn very complex functions

**ANN**
- Nondeterministic algorithm
- Generalizes well but doesn’t have strong mathematical foundation
- Can easily be learned in incremental fashion
- To learn complex functions—use multilayer perceptron (nontrivial)
Clinical use: SVM are more accurate than ANN

Example: Multiclass cancer diagnosis (for Exercise)

Counterexample: Move problem to a feature space $\mathcal{H}$

Summary: The 10 top algorithms - Quiz

- **C4.5**
  - for generation of decision trees used for **classification**, (statistical classifier, Quinlan (1993));

- **k-means**
  - simple iterative method for partition of a dataset in a user-specified n of **clusters**, k (Lloyd (1957));

- **Apriori**
  - for finding frequent item sets using candidate generation and **clustering** (Agrawal & Srikant (1994));

- **EM**
  - Expectation–Maximization algorithm for finding maximum likelihood estimates of parameters in models (Dempster et al. (1977));

- **PageRank**
  - a search ranking algorithm using hyperlinks on the Web (Brin & Page (1998));

- **Adaptive Boost**
  - one of the most important ensemble methods (Freund & Shapire (1995));

- **k-Nearest Neighbor**
  - a method for **classifying** objects based on closest training sets in the feature space (Fix & Hodges (1951));

- **Naive Bayes**
  - can be trained efficiently in a supervised learning setting for classification (Domingos & Pazzani (1997));

- **CART**
  - **Classification** And Regression Trees as predictive model mapping observations about items to conclusions about the goal (Breiman et al 1984);

- **SVM** support vector machines offer one of the most robust and accurate methods among all well-known algorithms (Vapnik (1995));

Why do we need Clustering?

- Group similar objects into clusters together, e.g.
  - For image segmentation
  - Grouping genes similarly affected by a disease
  - Clustering patients with similar diseases
  - Cluster biological samples for category discovery
  - Finding subtypes of diseases
  - Visualizing protein families

- Inference: given $x_i$, predict $y_i$ by learning $f$
- No training data set – learn model and apply it
Partite a data set into k clusters so that intra-cluster variance is a minimum

- $V$ ... variance (objective function)
- $S_i$ ... cluster
- $Y_i$ ... mean
- $D$ ... set of all points $x_j$
- $k$ ... number of clusters

$$V(D) = \sum_{i=1}^{k} \sum_{x_j \in S_i} (x_j - \mu_i)^2$$
Algorithm 1: Example for a classical weight balanced k-means algorithm

Input: $d,k,n \in \mathbb{N}$, $X := \{x_1, \ldots, x_n\} \subset \mathbb{R}^d$, $S := \{s_1, \ldots, s_k\} \subset \mathbb{R}^d$

Output: Clustering $C = (C_1, \ldots, C_k)$ of $X$ and the arithmetic means $c_1, \ldots, c_k$ as sites

1. Partition $X$ into a clustering $C = (C_1, \ldots, C_k)$ by assigning $x_j \in X$ to a cluster $C_i$ that is closest to site $s_i \in S$.
2. Update each site $s_i$ as the center of gravity of cluster $C_i$; if $|C_i| = 0$, choose $s_i = x_l$ for a random $l \leq n$ with $x_l \neq s_j$ for all $j \leq k$. If the sites change, go to (1.).

Merely an increase in awareness of physicians on risk factors for ARA in children can be sufficient to change their attitudes towards antibiotics prescription. Our results can also be useful when preparing recommendations for antibiotics prescription and to guide the standardized health data record.

Advantages of k-Means?

What is the computational time of k-means?

\[ O(np^k t) \]

- compute kn distances in p dimensions
- number of iterations

Can be small if there's indeed a cluster structure in the data

The Basics: Centroid and Medoid

- **Centroid**: mean of the points in the cluster.
  \[ \mu = \frac{1}{|C|} \sum_{x \in C} x \]

- **Medoid**: point in the cluster that is closest to the centroid.
  \[ m = \arg \min_{x \in C} d(x, \mu) \]
(“Applied ML is basically feature engineering. Andrew Y. Ng”).

2) Feature Engineering
- Feature:= specific measurable property of a phenomenon being observed.
- Feature engineering:= using domain knowledge to create features useful for ML. ("Applied ML is basically feature engineering. Andrew Ng").
- Feature learning:= transformation of raw data input to a representation, which can be effectively exploited in ML.
Feature Space Basic Definitions

- Intuitively: a domain with a distance function
- Formally: Feature Space $\mathcal{F} = (\mathcal{D}, d)$
  - $\mathcal{D} = \text{ordered set of features}
  - $d: \mathcal{D} \times \mathcal{D} \to \mathbb{R}^+_0$ ... a total distance function; true for
    - $\forall p, q \in \mathcal{D}, p \neq q: d(p, q) > 0$ (strict)
    - and must be reflexive and symmetric

![Diagram](image.png)

Image credit to Pascal Vincent
A **Metric Space** is a pair \((X, d)\) where \(X\) is a set and \(d : X \times X \to \mathbb{R}^+\), called the metric, s.t.

1. For all \(x, y, z \in X\), \(d(x, y) \leq d(x, z) + d(z, y)\).
2. For all \(x, y \in X\), \(d(x, y) = d(y, x)\).
3. \(d(x, y) = 0\) if and only if \(x = y\).

**Remark 1.** One example is \(\mathbb{R}^d\) with the Euclidean metric. Spheres \(S^n\) endowed with the spherical metric provide another example.

\[
d : X \to \mathbb{R} \\
d(x, x) = 0 \\
d(x^1, x^2) = d(x^2, x^1) \quad \text{symmetry} \\
d(x^1, x^2) \leq d(x^1, x^3) + d(x^3, x^2) \quad \text{triangle inequality}
\]
Let do a Quiz again: Similarities of feature vectors

Look at the examples below, which distance measures would you select?

Euclidian norm  |  Manhattan norm  |  Maximums norm
Feature Selection: Overview

- Filter approaches
  - Features
  - Predictor
- Embedded approaches
  - Features
  - Predictor
- Wrapper approaches
  - Features
  - Predictor
- Subset selection:
  - forward selection
  - backward selection
  - floating selection

Image credit to Chloe Azencott
Feature selection is just selecting a subset of the existing features without any transformation.

Feature extraction is *transforming* existing features into a lower dimensional space.

3) Curse of Dimensionality
Remember: The curse of dimensionality


http://www.iro.umontreal.ca/~bengioy/yoshua_en/research.html
Examples for High-Dimensional Biomedical Data

- **Medical Image Data (16 - 1000+ features)**

  [Image of medical image data]

  http://qsota.com/melanoma/

- **MEG Brain Imaging**

  120 locations x 500 time points
  x 20 objects

  [Image of MEG brain imaging]

  Nature 508, 199–206
  doi:10.1038/nature13185
Examples for High-Dimensional Biomedical Data

- Biomedical Signal Data (10 - 1000+ features)

http://www.nature.com/articles/srep21471#f1

http://www.mdpi.com/1424-8220/14/4/6124/htm

http://www.clinicalgaitanalysis.com/data/
Metabolome data (feature is the concentration of a specific metabolite; 50 – 2000+ features)

http://www.nature.com/ncomms/2015/151005/ncomms9524/fig_tab/ncomms9524_F5.html
Microarray Data (features correspond to genes, up to 30k features)

Text > $10^9$ documents $\times 10^6$ words/n-grams features correspond to words or terms, between 5k to 20k features

Text (Natural Language) is definitely very important for health:
- Handwritten Notes, Drawings
- Patient consent forms
- Patient reports
- Radiology reports
- Voice dictations, annotations
- Literature !!!

https://www.researchgate.net/publication/255723699_An_Answer_to_Who_Needs_a_Stylus_on_Handwriting_Recognition_on_Mobile_Devices
Example: UMLS – Unified Medical Language System
Why are many features problematic?

- Hyperspace is large – all points are far apart
- Computationally challenging (in time and space)
- Complexity grows with n of features
- Complex models less robust – more variance
- Statistically challenging – hard to learn
- Hard to interpret and hard to visualize
- Problem with redundant features and noise
- Question: Which algorithms will provide worse results with increasing irrelevant features?
- Answer: Distance-based algorithms generally trust all features of equal importance
Space and Time: Simple example on gait analysis

- Aspect 1: Optimization Problem
- Aspect 2: Concentration Effect
- Aspect 3: Irrelevant Attributes
- Aspect 4: Correlated Attributes

Example: Neonatal Screening (2/3)

3) Dimensionality Reduction
Why should we reduce the dimensionality?

- Data visualization only possible in $\mathbb{R}^2$ (R3 cave)
- Human interpretability only in $\mathbb{R}^2/\mathbb{R}^3$
  (visualization can help sometimes with parallel coordinates)
- Simpler (=less variance) models are more robust
- Computational complexity (time and space)
- Eliminate non-relevant attributes that can make it more difficult for algorithms to learn
- Bad results through (many) irrelevant attributes?
- *Note again: Distance-based algorithms generally trust that all features are equally important.*
- Given \( n \) data points in \( d \) dimensions
- Conversion to \( m \) data points in \( r < d \) dimensions
- Challenge: **minimal loss of information \( \ast \)**

\( \ast \) this is always a grand challenge, e.g. in k-Anonymization – see later in this
- Very dangerous is the “modeling-of-artifacts”
- Linear methods (unsupervised):
  - PCA
  - FA
  - MDS
- Supervised methods:
  - LDA
- Non-linear methods (unsupervised):
  - Isomap (Isometric feature mapping)
  - LLE (locally linear embedding)
  - Autoencoders
Example 1: PCA

- Subtract mean from data (center X)
- (Typically) scale each dimension by its variance
  - Helps to pay less attention to magnitude of dimensions
- Compute covariance matrix $S$
  $$S = \frac{1}{N}X^\top X$$
- Compute $k$ largest eigenvectors of $S$
- These eigenvectors are the $k$ principal components

Example 2 ICA (Motivation: Blind Source Separation)

- Suppose that there are $k$ unknown independent sources
  \[ \mathbf{s}(t) = [s_1(t), \ldots, s_k(t)]^T \text{ with } E\mathbf{s}(t) = 0 \]

- A data vector $\mathbf{x}(t)$ is observed at each time point $t$, such that
  \[ \mathbf{x}(t) = \mathbf{A}s(t) \]
  where $\mathbf{A}$ is a $n \times k$ full rank scalar matrix

Factor analysis describes the variability of observations in terms of unobserved latent variables, called factors, and noise

- factors explain correlation between the variables
- remaining variance is explained by Gaussian noise

Factor analysis is a generative approach and models both the noise of the observations and their correlation

assumptions on the distribution of factors and noise
Example 4: MDS: Find projection that best preserves $d$

- Find a set of points whose pairwise distances match a given distance matrix

<table>
<thead>
<tr>
<th></th>
<th>p1</th>
<th>p2</th>
<th>p3</th>
<th>p4</th>
<th>p5</th>
</tr>
</thead>
<tbody>
<tr>
<td>p1</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>p2</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>p3</td>
<td>2</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>p4</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>p5</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

$x_i$ Point in $d$ dimensions
$y_i$ Corresponding point in $r < d$ dimensions
$\delta_{ij}$ Distance between $x_i$ and $x_j$
$d_{ij}$ Distance between $y_i$ and $y_j$

- Define (e.g.) $E(y) = \sum_{i,j} \left( \frac{d_{ij} - \delta_{ij}}{\delta_{ij}} \right)^2$
- Find $y_i$'s that minimize $E$ by gradient descent
- Invariant to translations, rotations and scalings
Seeking Life’s Bare (Genetic) Necessities

COLD SPRING HARBOR, NEW YORK—How many genes does an organism need to survive? Last week at the genome meeting here, two genome researchers with radically different approaches presented complementary views of the basic genes needed for life. One research team, using computer analyses to compare known genomes, concluded that today’s organisms can be sustained with just 250 genes, and that the earliest life forms required a mere 128 genes. The other researcher mapped genes in a simple parasite and estimated that for this organism, 800 genes are plenty to do the job—but that anything short of 100 wouldn’t be enough.

Although the numbers don’t match precisely, those predictions are not all that far apart,” especially in comparison to the 75,000 genes in the human genome, notes Sti Andersson of Uppsala University in Sweden, who arrived at the 800 number. But coming up with a consensus answer may be more than just a genetic numbers game, particularly as more and more genomes are completely mapped and sequenced. “It may be a way of organizing any newly sequenced genome,” explains Aracdy Mushegian, a computational molecular biologist at the National Center for Biotechnology Information (NCBI) in Bethesda, Maryland. Comparing an

“Stripping down. Computer analysis yields an estimate of the minimum modern and ancient genomes.


SCIENCE • VOL. 272 • 24 MAY 1996
Example 6: Isomap

A Global Geometric Framework for Nonlinear Dimensionality Reduction

Joshua B. Tenenbaum,1,2 Vin de Silva,2 John C. Langford3

Scientists working with large volumes of high-dimensional data, such as global climate patterns, stellar spectra, or human gene distributions, regularly confront the problem of dimensionality reduction: finding meaningful low-dimensional structures hidden in their high-dimensional observations. The human brain confronts the same problem in everyday perception, extracting from its high-dimensional sensory inputs—30,000 auditory nerve fibers or 106 optic nerve fibers—a manageable small number of perceptually relevant features. Here we describe an approach to solving dimensionality reduction problems that uses easily measured local metric information to learn the underlying global geometry of a data set. Unlike classical techniques such as principal component analysis (PCA) and multidimensional scaling (MDS), our approach is capable of discovering the nonlinear degrees of freedom that underlie complex natural observations, such as human handwriting or images of a face under different viewing conditions. In contrast to previous algorithms for nonlinear dimensionality reduction, ours efficiently computes a globally optimal solution, and, for an important class of data manifolds, is guaranteed to converge asymptotically to the true structure.

Goal: Find projection onto nonlinear manifold

1. Construct neighborhood graph $G$:
   - For all $x_i, x_j$
     - If distance$(x_i, x_j) < \epsilon$
       - Then add edge $(x_i, x_j)$ to $G$

2. Compute shortest distances along graph $\delta_G(x_i, x_j)$
   (e.g., by Floyd’s algorithm)

3. Apply multidimensional scaling to $\delta_G(x_i, x_j)$

http://isomap.stanford.edu/

**Example 7: LLE**

Locally linear embedding (LLE) computes low-dimensional, neighborhood-preserving embeddings / representations. LLE performs nonlinear mappings. The objective is

$$\varepsilon(W) = \sum_i \left\| x_i - \sum_{j=1}^{k} W_{ij} x_j \right\|^2 \quad \sum_{j=1}^{k} W_{ij} = 1$$

Optimized by constrained least squares using neighbors $x_j$ of $x_i$.

The solutions of this problem are invariant to rotations, rescalings, and translations of $x_i$.

Down-projection optimizes

$$\Phi(Y) = \sum_i \left\| y_i - \sum_{j=1}^{k} W_{ij} y_j \right\|^2$$

where the $W_{ij}$ are fixed.

The representation of $x_i$ by its neighbors is transferred to $y_i$

$$\Phi(Y) = \sum_{ij} M_{ij} y_i^T y_j$$

$$M_{ij} = \delta_{ij} - W_{ij} - W_{ji} + \sum_k W_{ki} W_{kj}$$

$$\delta_{ij} : 1 \text{ for } i=j, \ 0 \text{ otherwise}$$

Optimal embedding: bottom $d$ eigenvectors of $M$, except the last one.
Example 8: Autoencoders

- **History:** Dim-reduction with NN: Learning representations by back-propagating errors
- **Goal:** output matches input


Autoencoders are “old”


  \[ \Delta(y, x) = ||y - x||^2_2 \]


  \[ \min_{A,B} \sum_x ||AXBx - x||^2_2 \]

*) David Rumelhart (1942-2011) was Cognitive Scientist working on math. Psychology
Autoencoders -> Restricted Boltzmann Machines

- Based on Information processing in dynamical systems: Foundations of harmony theory by Smolensky (1986): Stochastic neural networks where the unit activation $i = \text{probabilistic}$

\[
Pr(o_i = 1) = \frac{1}{1 + e^{-w_{i0} + \sum_j o_j w_{ij}}}
\]

- $o_i = 0 \text{ or } 1$

Right: A restricted Boltzmann machine with binary hidden units and softmax visible units

Goal: Having \( m < p \) features

Feature selection via

- A) Filter approaches
- B) Wrapper approaches
- C) Embedded approaches (Lasso, Electric net, see Tibshirani, Hastie ...)

Feature extraction

- A) Linear: e.g. PCA
- B) Non-linear: Autoencoders (map the input to the output via a smaller layer)
4) Subspace Clustering* and Analysis

* Two major issues
(1) the algorithmic approach to clustering and
(2) the definition and assessment of similarity versus dissimilarity.
What is subspace clustering?

- K clusters
- N data points
- $D$ dimensions (original space)
- $d$ dimensions (latent subspace)
- SC = clustering data whilst reducing the $d$ of each cluster to a cluster-dependent subspace


### Large Amount of Records

<table>
<thead>
<tr>
<th>Year</th>
<th>Type</th>
<th>Data Source</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2020</td>
<td>Type A</td>
<td>Hospital</td>
<td>Large dataset of patient records</td>
</tr>
<tr>
<td>2019</td>
<td>Type B</td>
<td>Survey</td>
<td>Large survey dataset</td>
</tr>
<tr>
<td>2018</td>
<td>Type C</td>
<td>Social Media</td>
<td>Large dataset of user interactions</td>
</tr>
</tbody>
</table>

### High-Dimensional Data

Geben Sie eine Formel ein.

Large Amount of Dimensions

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>123</td>
<td>456</td>
</tr>
<tr>
<td>789</td>
<td>012</td>
</tr>
<tr>
<td>321</td>
<td>654</td>
</tr>
</tbody>
</table>

Holzinger Group
- Irrelevant Dimensions
- Correlated and Redundant Dimensions
- Conflicting Dimensions
- Challenging Interpretation of data and analysis results

- NN problem: Given \( n \) data points and a query point in an \( m \) -dimensional metric space
- find the data point closest to the query point.

- Concentration Effect
  - Discriminability of similarity gets lost
  - Impact on usefulness of a similarity measure

- High-Dimensional Data is Sparse

Optimization Problem and Combinatorial Issues
Feature selection and dimension reduction
$2^d - 1$ possible subsets of dimensions ($\rightarrow$ subspaces)
Example: Full Space Clustering of High-Dimensional Data

Normalized Distance between records / clusters

Data Records
Patterns may be found in subspaces (dimension combinations)
Patterns may be complementary or redundant to each other
Example of 12D Data -> 4095 subspaces (296 interesting)
Motivation

Single Distance Function: $d(\bullet, \triangle) \rightarrow R$, based on \textbf{Fixed} dimensions [shape, color, size, rotation]

Motivation

k-Nearest Neighbors: Ranked list of most similar objects
- **Attention**: Similarity measures lose their discriminative ability
- **Noise**, irrelevant, redundant, and conflicting dimensions appear

![Diagram of k-Nearest Neighbor Query](image)

- Query Object
- **k-Nearest Neighbor Query**
  - distance function
  - set of dimensions
Application in a Clinical Scenario

(1) Relevant subspaces *depend on patient* and are *unknown* beforehand

(2) *Multiple* subspaces might be relevant

(3) Subspaces helps to *interpret* the nearest neighbors (*semantic* meaning)

Sex, Age, Blood Type, Blood Pressure, Former Diseases, Medication, ...
1. Detect all previously unknown subspaces that are relevant for a NN-search
2. Determine the respective set of NN within each relevant subspace

Characteristics:
• Search for different NN’s in different subspaces
• Consider local similarity (instead of global)
• Subspaces are query dependent
• Subspaces are not an abstract concept but helps to semantically interpret the nearest neighbors
Subspace clustering aims at finding clusters in different axis-parallel or arbitrarily-oriented subspaces [1]

Subspace Outlier Detection search for subspaces in which an arbitrary, or a user-defined object is considered as outlier [2].


Relevance of Nearest Neighbors

A set of objects $a, b, c$ are NN of the query $q$ in a subspace $s$, iff $a, b, c$ are similar to $q$ in all dimensions of $s$.

Relevance of a Subspace

A subspace is considered relevant, iff it contains relevant nearest neighbors.

Advantages of Subspace Modelling

- **Interpretability**: reflects the semantic meaning
  - In which way are NN’s similar to the query?
  - In all dimensions of the subspace

- **Fulfills the downward-closure property**
  - Make use of *Apriori-like algorithms* for subspace search

- **No global distance function necessary**
  - Heterogeneous subspaces can be described
  - Compute the nearest neighbors in every dimension separately (with an appropriate distance function)
  - Compute subspace by intersection
Query Based Interestingness Measure for Dimensions

Non-Characteristic Dimension

Characteristic Dimension

Data Distribution
Query Based Interestingness Measure for Dimensions

query = butter

query = gauda cheese
Initial experiments

Supplementary Material
- http://files.dbvis.de/sisap2015

Dataset
- USDA National Nutrition Database

Experiment
- Full Space (Eucl. distance, 50 dim.)
- Subspaces (our model)
(1) Determine Nearest Neighbors per Dimension
(2) Efficient Search Strategy
(3) Query-Based Interestingness for Dimensions
(4) Subspace Quality Criterion (Depends on Analysis Task)
(5) Evaluation Methods and Development of Benchmark Datasets
(6) Multi-input Subspace Nearest Neighbor Search
(7) Visualization and User Interaction
- Variety of different algorithms, e.g. PROCLUS [1], CLIQUE [2], RESCUE [3]
- Example CLIQUE:

- Challenges
- Exponential # of possible subspaces
- Result highly depend on parameters
- Highly redundant results (clusters + subspaces)
Example Clust Nails Tatu et al (2012)

Which dimensions occur more often in clusters?
Which occur often together?
Which values do records in a specific cluster have?


Further Subspace Cluster Visualization Techniques

- VISA by Assent et al. (2007)
- CoDa by Günnemann et al. (2010)
- Morpheus by Müller et al. (2008)
- Visual Analytics Framework by Tatu et al. (2012), see before
Visual Analytics for Subspace Steering

- Existing techniques: **exploration** of subspace clusters
- Visualizations to **make sense** of clusters and its subspaces
  
  Is the parameter setting appropriate for the data?

What happens if algorithms cannot scale with the #dimensions?

- We need methods to **steer algorithms** while computing relevant subspaces
  
  - Pruning of intermediate results
  - Adjust parameters to domain knowledge
  - ...
Data in only one dimension is relatively packed
Adding a dimension “stretch” the points across that dimension, making them further apart
Adding more dimensions will make the points further apart—high dimensional data is extremely sparse
Distance measure becomes meaningless—due to equidistance
Repeat some definitions

- **Dataset** - consists of a matrix of data values, rows represent individual instances and columns represent dimensions.
- **Instance** - refers to a vector of $d$ measurements.
- **Cluster** - group of instances in a dataset that are more similar to each other than to other instances. Often, similarity is measured using a distance metric over some or all of the dimensions in the dataset.
- **Subspace** - is a subset of the $d$ dimensions of a given dataset.
- **Subspace Clustering** – seek to find clusters in a dataset by selecting the most *relevant* dimensions for each cluster separately.
- **Feature Selection** - process of determining and selecting the dimensions (features) that are most relevant to the data mining task.
Interesting Clusters may ONLY exist in subspaces!!

Similar concept: Principal Component Analysis (PCA)
6) “What is interesting?”
Projection Pursuit
• **Projection pursuit**: Find a subset of coordinates of the data which display “interesting” features. Often the selection of the subset of coordinates is manual, but there are automated algorithms which can find these subsets automatically also. Finally one has to inspect each projection and decide if its “interesting”.

Projection pursuit:
least Gaussian ("interesting") projections of the data

how to define non-Gaussianity?

covariance and mean given: Gaussian distribution maximizes the entropy

Objective: minimize $H(t)$ for $t = w^T x$

- $t$ is normalized to zero mean and unit variance

This is difficult to optimize
  - finding unimodal super-Gaussians
  - finding multimodal distributions

Other criteria are given for ICA: kurtosis and different contrast functions which measure non-Gaussianity
145 diabetes patients

6 dimensional data set:

- 1) age,
- 2) relative weight,
- 3) fasting plasma glucose,
- 4) area under the plasma glucose curve for the three hour glucose tolerance test (OGTT),
- 5) area under the plasma insulin curve for the OGTT,
- 6) steady state plasma glucose response.

Method: Projection Pursuit (PP)

Result: \( \mathbb{R}^6 \rightarrow \mathbb{R}^3 \)

Given a point cloud data set $X$ and a covering $U$ 
$\Rightarrow$ simplicial complex

$$f : X \rightarrow \mathbb{R}$$

$$f : X \rightarrow \mathbb{Z}$$

$$U = \{ U_\alpha \}_{\alpha \in A}$$

$$f_\varepsilon(x) = C_\varepsilon \sum_y \exp\left(\frac{-d(x, y)^2}{\varepsilon}\right)$$

Future Outlook

- Time (e.g. entropy) and Space (e.g. topology)
- Knowledge Discovery from “unstructured” ;-) (Forrester: >80%) data and applications of structured components as methods to index and organize data -> Content Analytics
- Open data, Big data, sometimes: small data
- Integration in “real-world” (e.g. Hospital), mobile
- How can we measure the benefits of visual analysis as compared to traditional methods?
- Can (and how can) we develop powerful visual analytics tools for the non-expert end user?
Thank you!
Sample Questions

- Why would we wish at all to reduce the dimensionality of a data set?
- Why is feature selection so important? What is the difference between feature selection and feature extraction?
- What types of feature selection do you know?
- Can Neural Networks also be used to select features?
- Why do we need a human expert in the loop in subspace clustering?
- What is the advantage of the Projection Pursuit method?
- Why is algorithm selection so critical?