Sandipan Dey

Q1: What is the difference between Supervised and Unsupervised methods? Explain with examples. 7pts

## Answer

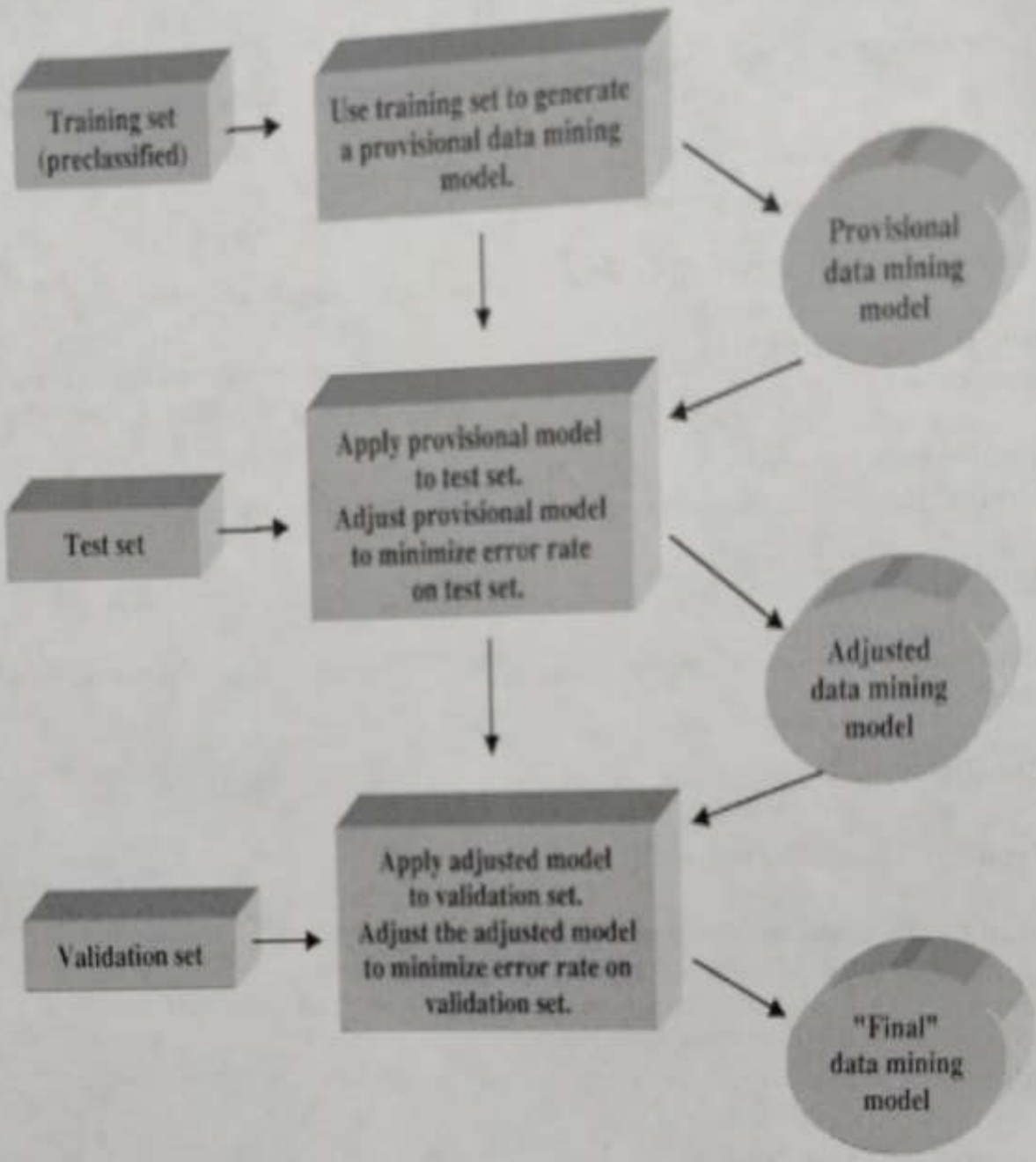
### Supervised and Unsupervised methods in Data Mining

Data Mining is extraction of hidden, nontrivial, unknown and useful information (knowledge / pattern) from data, i.e., learning from the data, that data comes in two flavors:

1. Supervised learning methods.
2. Unsupervised learning methods.

### Supervised methods

- Supervised learning is a machine learning technique for learning a function from training data.
- **Classification** is a supervised technique.
- Typically it learns by **example**.
- Must have a set of examples **correctly labeled** with **predefined** (class) **labels** or concepts (also called **training data set**) – this data set will be used to train the method, as learning step. Labels for **training tuples** must be **known**.
- It's known **a priori** what the desired output (class labels) should be.
- Generally a 3 **step process**:
    - **Training (learning) phase** – A **model** (with a **set of rules**) is built from the correctly labeled examples (**training data set**).
    - **Validation phase** – **Accuracy** of the **model** built is **tested** with another holdout dataset (also called **validation data set**) for which **labels** are known but kept hidden from the method, instead the method is asked to **predict the labels** using the **model** built and the patterns/structures learn. Then the **predicted labels** are **validated** against **the original labels** and accordingly the structure of the model learnt is adjusted, if required.
    - **Test phase** - The model learnt is actually applied on the unlabeled data.
- Following figure represents the schematic diagram of a supervised method:




Methodology for supervised modeling.

- **Training** step can be viewed as the **learning of a mapping** or **function** $f$ such that $y = f(X)$, where **y** is the **class label** of a given data tuple $X$.
- The training data consist of pairs of input objects (typically vectors), and desired outputs (e.g., class labels).
- The output of the function can be a **continuous value** (called **regression**), or can predict a **class label** of the input object (called **classification**).

- The task of the supervised learner is to **predict** the value of the function for any valid input object after having seen a number of **training examples** (i.e. pairs of input and target output).

- To achieve this, the supervised learner has to **generalize** from the presented data to unseen situations in a reasonable way.

- The supervised learner must use a set of assumptions to **predict** outputs given inputs that it has **not encountered,** these set of assumptions are commonly known as the **inductive bias** of the learning algorithm.

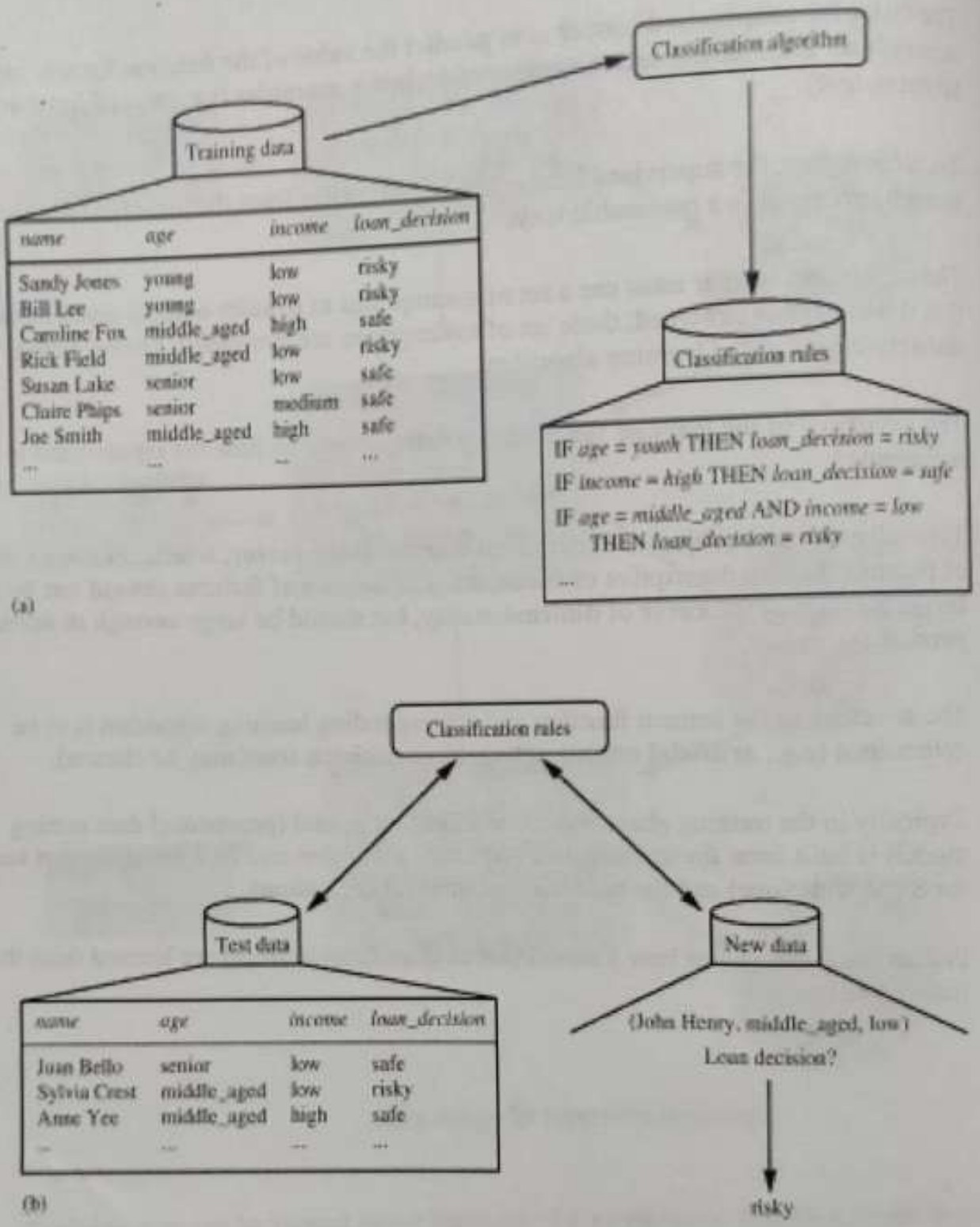
- The **accuracy** of the **learned function** depends strongly on how the input object is represented.

- Typically, the input object is transformed into a **feature vector**, which contains a number of features that are descriptive of the object. The number of features **should not be too large**, because of the **curse of dimensionality**, but should be **large enough** to **accurately predict** the output.

- The structure of the learned function and corresponding learning algorithm is to be determined (e.g., **artificial neural networks** or **decision trees** may be chosen).

- Typically in the training phase some **classification model** (provisional data mining model) is built from the training data (typically a decision tree or a set of support vectors for SVM technique) and the necessary parameters are estimated.

- Following figure shows how a model (set of classification rules) are learned from the training phase:



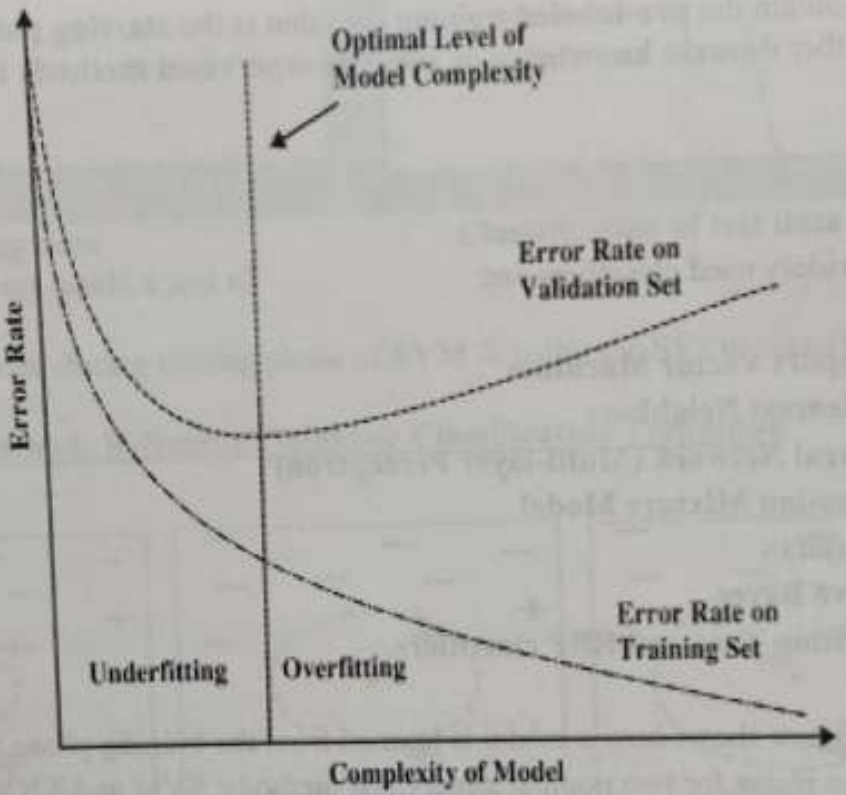

The data classification process: (a) *Learning*: Training data are analyzed by a classification algorithm. Here, the class label attribute is *loan_decision*, and the learned model or classifier is represented in the form of classification rules. (b) *Classification*: Test data are used to estimate the accuracy of the classification rules. If the accuracy is considered acceptable, the rules can be applied to the classification of new data tuples.

- The large amount of data that is usually present in supervised data mining tasks allows splitting of the data file in **three groups**: **training cases, validation cases** and test cases.

- **Validation** helps to see whether the model obtained with one chosen sample (training dataset) may be **generalizable** to other data - avoiding the phenomenon of **overfitting**.

- **Overfitting** results when provisional model tries to account for every possible trend or structure even the idiosyncratic ones – it essentially reduces the accuracy.

- There is an eternal tension in model building between **model complexity** (increasing the **accuracy** in **training set** and **generalizability** to validation and test set - increasing model complexity in order to increase accuracy on the training set essentially eventually leads to degradation in generalizability of the provisional model to the validation and test data set.




The optimal level of model complexity is at the minimum error rate on the validation set.

- Hence, **accuracy** of classification using the provisional model is usually **not as high** on **validation set** or **test set** as on training data set.

- Parameters of the learning algorithm may be **adjusted** by optimizing performance on a **subset** (called a **validation set**) of the training set, or via **cross-validation**.

- After parameter adjustment and learning, the performance of the algorithm may be measured on a **test data set** that is separate from the training set.

- Supervised learning is often also called **directed data mining**.

- In the variables under investigation can be split into two groups: **explanatory variables** (typically the class label) and one (or more) **dependent variables** (attributes other than class label), by **discriminant** analysis.

- The target of the analysis is to specify a **relationship** between the **explanatory variables** the **dependent variables**. To apply directed data mining techniques the **values** of the **dependent variable** must be known for a sufficiently large part of the data set.

- Supervised learning requires that the **target variable** (class label) is **well defined** and that a sufficient number of its values are **given**.

- In order to obtain the **pre-labeled** training data that is the **starting point** for a **supervised method**, either **domain knowledge** or some **unsupervised methods like clustering** can be used.

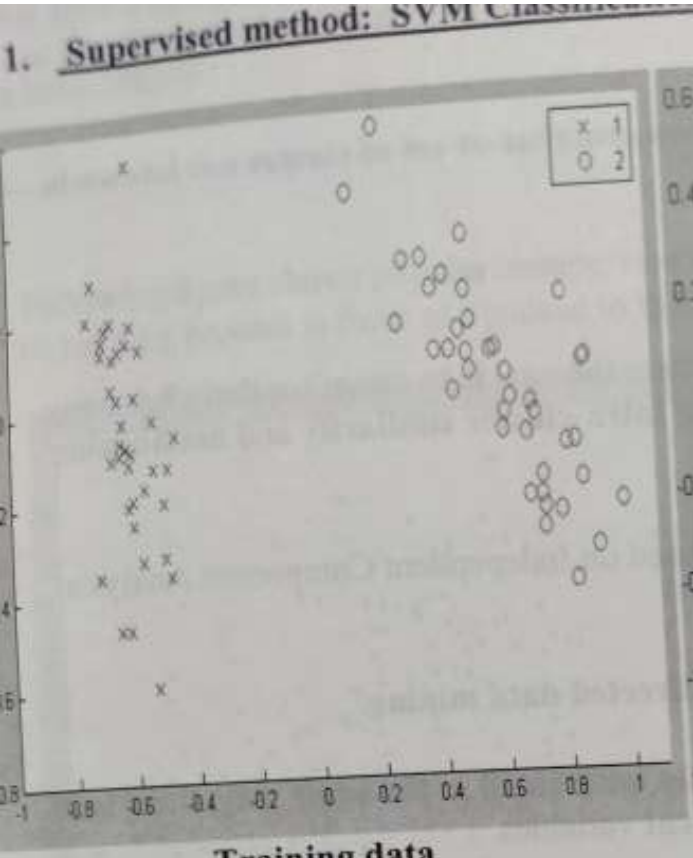
- **Examples:**

  The most widely used classifiers are:

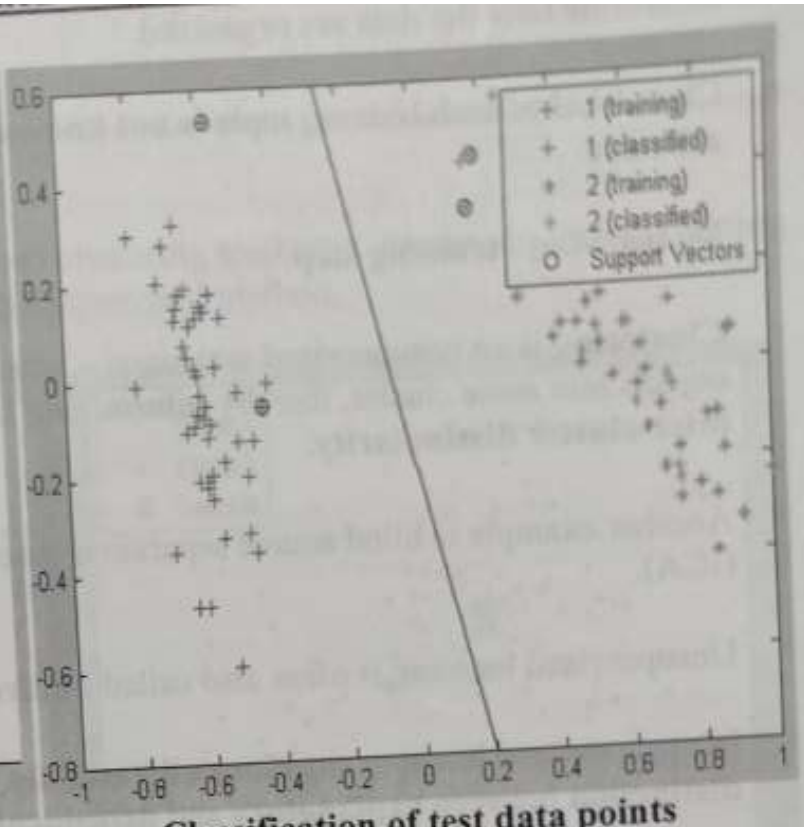
  - **Support Vector Machines**
  - **k-Nearest Neighbors**
  - **Neural Network (Multi-layer Perceptron)**
  - **Gaussian Mixture Model**
  - **Gaussian**
  - **Naive Bayes**
  - **Decision Tree and RBF classifiers.**

- Following figure shows how a model is learned from the training phase and then used in classification phase for two popular supervised methods: SVM and KNN Classification techniques.

1. **Supervised method: SVM Classifica**



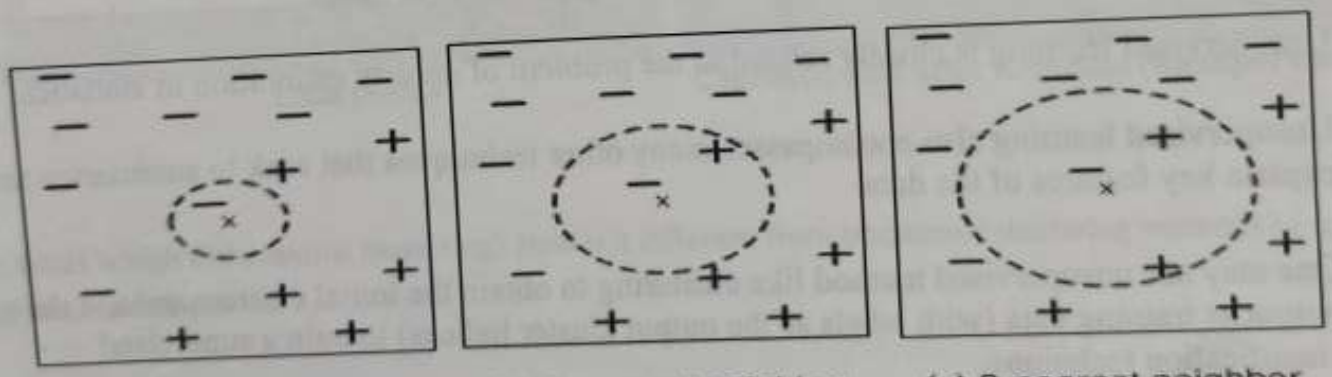

**Training data**
(2 pre-defined class labels **x** and **o**)




**Classification of test data points**

**Support vectors** obtained during raining phase of **SVM** & using the **SV**s to classify data points

2. **Supervised Method: K-Nearest Neighbor Classification Technique**



(a) 1-nearest neighbor (b) 2-nearest neighbor (c) 3-nearest neighbor

K-nearest neighbors of a record x are data points that have the k smallest distance to x

**Training data** (2 pre-defined class labels - and +) & **Classification of test data point x** (Class label to be decided by Euclidian proximity of the point from pre-labeled training points)

**Unsupervised methods**

- In machine learning, **unsupervised learning** is a class of problems in which one seeks determine how the data are organized.
- Class label of each training tuple is **not known, number or set of classes not known** advance.
- No **learning / training step**.
- **Clustering** is an unsupervised technique – where the goal is to group similarly behavin objects into same cluster, thereby **minimizing intra-cluster similarity** and **maximizin inter-cluster dissimilarity**.
- Another example is blind source separation based on Independent Component Analysis (ICA).
- Unsupervised learning is often also called **undirected data mining.**
- In unsupervised learning situations **all variables** are **treated** in **the same way**, there is **distinction** between **explanatory** and **dependent** variables.
- The dividing line between supervised learning and unsupervised learning is the same tha distinguishes **discriminant analysis** from **cluster analysis.**
- For unsupervised learning typically either the **target variable** (e.g., class label) is **unknown** or has only been recorded for too small a number of cases.
- Unsupervised learning is closely related to the problem of density estimation in statistics
- Unsupervised learning also encompasses many other techniques that seek to summarize explain key features of the data.
- One may use unsupervised method like clustering to obtain the initial clusters and use the output as training data (with labels as the output cluster indices) to train a supervised classification technique.
- As such, **unsupervised** methods like clustering does not use **previously assigned class labels**, **except** perhaps for **verification** of how well the clustering worked.
- Another way of determining the performance of unsupervised clustering method is to compute intra-cluster dissimilarities (e.g., from centroid) like **SSE** after clustering.

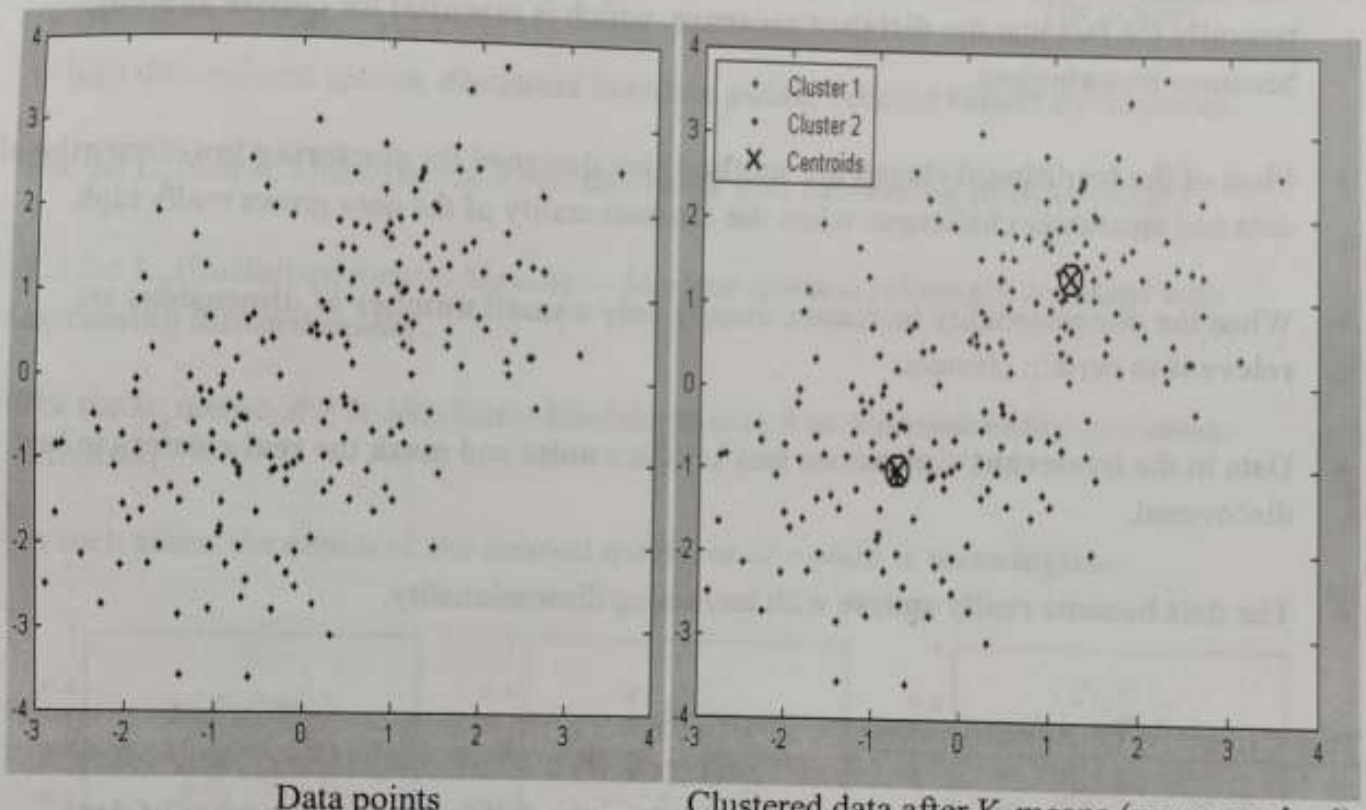
IS 733 Data Warehousing and Mining

Spring 2009 – Take home Final

Due Date: 5/18/09

---

- Following figure shows popular unsupervised clustering method K-means, no that no training or learning process is there as opposed to the supervised method.




Data points

Clustered data after K-means (**unsupervised**)

Q2: What is high dimensional clustering? How is it different from traditional clustering methods? Explain with examples **8 pts**

## Answer

- **Clustering** is the assignment of objects into groups (called clusters) so that objects from the same cluster are more similar to each other than objects from different clusters.

- **Minimize intra-cluster** (within cluster) **similarities** and **maximize inter-cluster** (across cluster) **dissimilarities**.

- Often **similarity** is assessed according to a **distance measure**.

- **Distance measures** give **poor similarity measures** with **increasing dimensions,** thereby creating problems in **high dimensional clustering**.

- The reason that **traditional clustering methods** don't scale well to **high dimensions** is typically the fact that the **distance measure**, which is **essential** for **cluster analysis**, becomes **meaningless**.

- Most of the **traditional clustering methods** are designed for **clustering low-dimensional** data and encounter challenges when the dimensionality of the data grows really high.

- When the dimensionality increases, usually only a **small number of dimensions** are **relevant** to certain clusters.

- Data in the **irrelevant** dimensions may produce **noise** and **mask** the **real clusters** to be discovered.

- The data become **really sparse** with increasing dimensionality.

## The Curse of Dimensionality

- Problems with **high dimensionality** result from the fact that a **fixed number** of **data points** become **increasingly sparse** as the dimensionality increase.

- Impossible to optimize a function of many variables by a brute force search on a discrete multidimensional grid.

- The number of **grids points** increases **exponentially** with **dimensionality**, i.e., with the **number of variables**.

- For **clustering** purposes, the most relevant aspect of the curse of dimensionality concerns the effect of **increasing dimensionality** on **distance or similarity**.

- Most clustering techniques depend critically on the measure of **distance** or **similarity**, and require that the objects **within clusters** are, in general, **closer** to each other than to objects in other clusters.

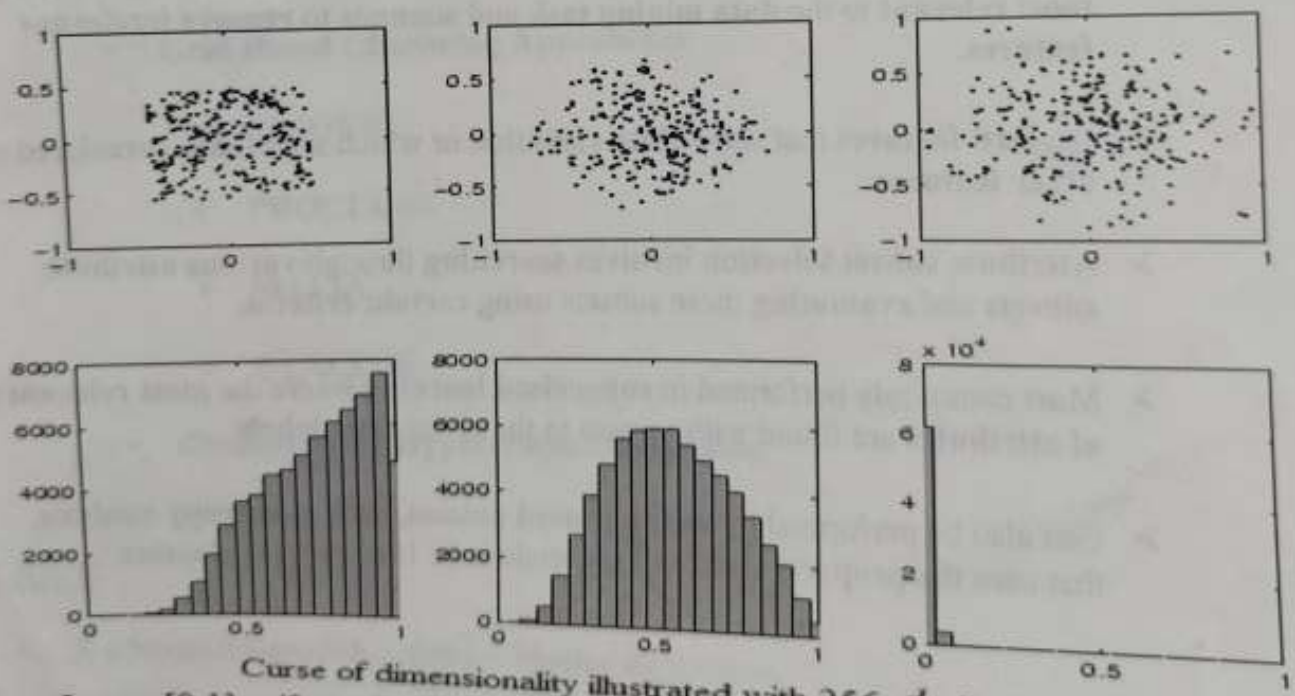
- It is shown, for certain data distributions (e.g., if all attributes are i.i.d.), that the relative difference of the distances of the closest and farthest data points of an independently

selected point goes to 0 as the dimensionality increases, i.e.,

$$\lim_{d \to \infty} \frac{MaxDist - MinDist}{MinDist} = 0$$

[BGRS99]

- In high dimensional spaces, **distances between points** become **relatively uniform**.
- For the $\mathbf{L}_1$ metric, *MaxDist – MinDist* **increases** with **increasing dimensionality**.
- For the $\mathbf{L}_2$ (Euclidian) metric, *MaxDist – MinDist* remains **relatively constant** with **increasing dimensionality**.
- For the $\mathbf{L}_d$ metric, $d \geq 3$, *MaxDist – MinDist* goes to **0** as **dimensionality increases**. [HAK00]
- In such cases, the notion of **the nearest neighbor** of a point is **meaningless**.



Curse of dimensionality illustrated with 256 $d$-dimensional points from a [0,1] uniform distribution with $d = 2$ (left), 4 (middle) and 32 (right). The top row shows the results of the 2D Principal Components Analysis (PCA). The bottom row shows how similarity (as a monotonically decreasing function of Euclidean distance) is distributed. As $d$ increases, projections approach Gaussian distributions. Also, an average pair of points' similarity decreases rapidly and similarities become approximately equal for most pairs with increasing $d$.

Spring 2009 – Take home Final

Due Date: 5/18/09

To overcome this difficulty, there are following two techniques:

- **Feature Transformation methods**
- **Feature Selection Methods**

- **Dimensionality reduction** by **Feature Selection Methods**
  - It is often possible to reduce the dimensionality of the data without losing important information.
  - Sometimes it is known ***apriori*** that only a **smaller number of variables** are o **interest**.
  - These variables (of interest) are **selected**, and the others discarded, thus reduc the dimensionality of the data set.
  - **Data analysis** (clustering or otherwise) is often preceded by a **feature selecti** or **attribute subset selection** step that finds the **subset of attributes** that are **most relevant** to the **data mining task** and attempts to **remove irrelevant features**.
  - **Discard features** that show **little variation** or which are **highly correlated w other features**.
  - **Attribute subset selection** involves **searching** through **various attribute subsets** and **evaluating** these subsets using **certain criteria**.
  - Most commonly performed in **supervised learning** where the **most relevant s of attributes** are found with respect to the given **class labels**.
  - Can also be performed by **unsupervised** process, such as **entropy analysis**, that uses the **property** that **entropy** tends to be **low** for **tight clusters**.

- **Subspace Clustering**
  - An extension to attribute subset selection.
  - Different subspaces may contain different, meaningful clusters.
  - Searches for groups of clusters within different subspa

- Problem becomes how to find such subspaces effectively and efficiently.

- **Feature Transformation methods**
  - **Transform** data to a **smaller space** while generally **preserving** the **relative distance** between objects.
  - Techniques like **Principal Component Analysis** (PCA) or **Singular Value Decomposition** (SVD) are feature transformation techniques.
  - **Project** points from a **higher dimensional space** to a **lower dimensional space**.
  - Often data can be **approximated** reasonably well even if only a relatively small number of dimensions are kept, and thus, little true information is lost.

- Examples of popular high dimensional clustering methods:
  - **Grid Based Clustering Approaches**
    - **CLIQUE**
    - **PROCLUSE**
    - **MAFIA**
    - **DENCLUE**
  - **Clustering via Hypergraph Partitioning**

**CLIQUE**

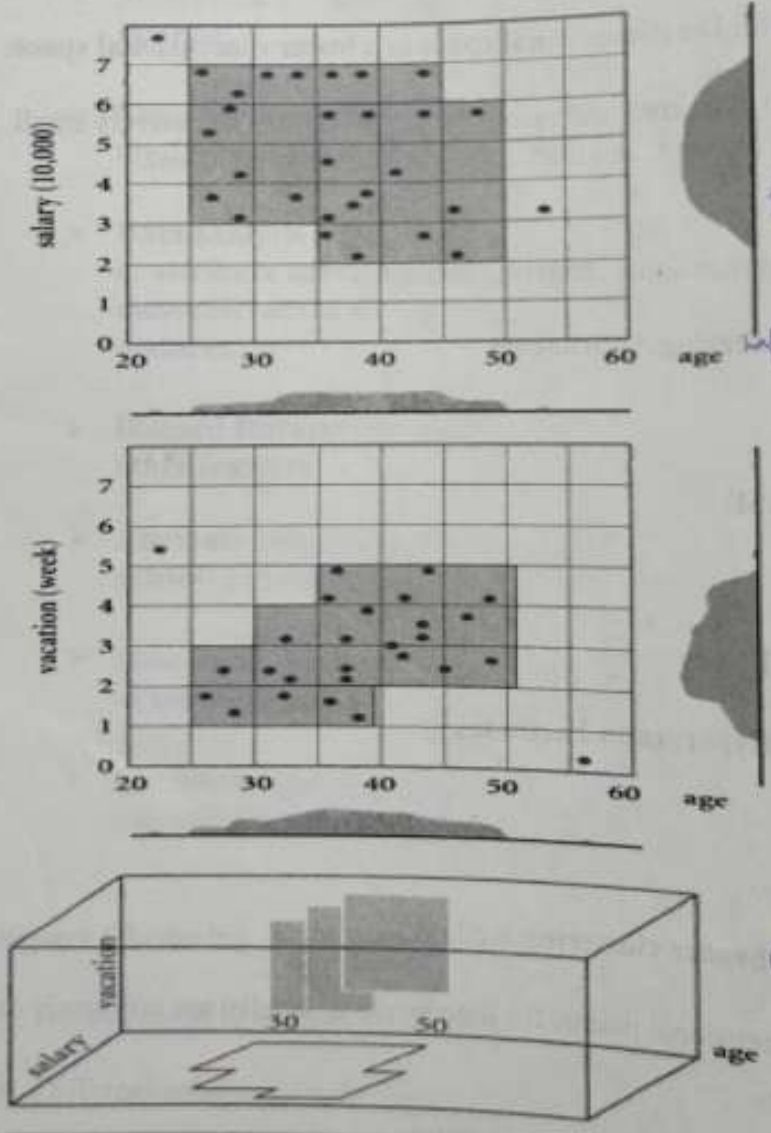
- A **dimension-growth subspace clustering** method.
- In a large set of multidimensional points, the data space is usually not uniformly occupied by the data points.
- Identifies sparse and dense areas in space (or **units**).

IS 733 Data Warehousing and Mining

Spring 2009 – Take home Final

Due Date: 5/18/09

- Cluster is defined as **maximal set** of **connected dense units**.
- Uses the following **apriori** property - a region that is **dense** in a **particular subspace** must create **dense** regions when **projected** onto **lower dimensional subspaces**.
- Potential dense units in **k-dimensional** space are generated from the dense units found in **k – 1** dimensional space.
- Resulting space searched is much smaller than the original space.




**Dense units found with respect to age for the dimensions salary and vacation are intersected in order to provide a candidate search space for dense units of higher dimensionality.**

① Predictive modeling
↳ Classification
↳ Regression
$g(x) = \begin{cases} 0, x \in A \\ 1, x \in B \end{cases}$
find $\hat{g}$ / separable plane / discriminant analysis (W, Y)
from training data by minimizing classification error.
by convex spaces (overlapping free)

Feature selection → select min# numbers of features for better generalization, simple models

SVM → maximize margin $\frac{2}{\|w\|_2}$
minimize misclassification errors.

Regression → obtain a true regression function by set of pre-defined function set

F.S. + S.V.M.

② Clustering → Define ~~cluster~~ function of clusters
grouping pts ~~minimize~~ maximize inter-cluster.
K Clusters ≤ find min# error

③ dependency Modelling → ~~es~~ compute pdf in semi-parametric approach by combination of K basis functions (Gaussian mixture model)

P(L) with $\mu_0, \sigma_1$