**Lecture Note**

**IN6244 Graph**

|  |  |  |
| --- | --- | --- |
| **~~Homework - Individual~~**  ~~50%~~  ~~Written report and Code~~ | **~~Group Project&Presantation~~**  ~~30%~~  ~~Real-world Problem~~  ~~3-5 People~~ | **~~Participation~~**  ~~20%~~ |

**Lecture#1 Introduce**

Content of IN6244 for the first Class:

1. Types of Graphs
2. Weighted / Unweighted
3. Directed / Undirected
4. Homogenous / Heterogeneous [Homogenous: all nodes are of the same type ]
5. Unsigned Graph / Signed Graph
6. Other Special Types

* Bipartite Graph - Projection ( 投影 )

1. Data Structure
2. Edge list
3. Adjacency llist
4. Adjacency Matrix
5. Common Graph Algorithms
6. BFS
7. DFS
8. Dijkstra's Algorithm
9. Pagerank Algorithm
10. Graph Learning (ML + Graph)
11. Node-level (Graph Learning Tasks: Node Classification)
12. Edge-level
13. Graph / Subgraph-level
14. Graph Learning Tasks
15. Node Classification
16. Link Prediction
17. Graph Classification / Prediction
18. Basic Graph Learning Algorithm
19. Graph Embedding
20. Deepwalk
21. Node2Vec
22. GNNs
23. Graph Neural Network
24. Graph Convolutional Network (GCN)
25. GraphSAGE
26. Graph Attention Networks (GATs)

**Lecture#2 Basic Concepts**

1. Graph Basic
2. Nodes (Vertices)
3. Edges
4. Attributes (Node and Edge features)
5. Graph Property
6. Degree
7. Average Degree
8. Density (Complete - Dense - sparse)
9. Path: A sequence of vertices connected by edges.
10. Length: The length of a path is the number of edges that it uses.
11. Distance: The length of the shortest path between two vertices
12. Cycle - Directed acyclic graph (DAG)
13. Features for Graph
14. Node-level Features

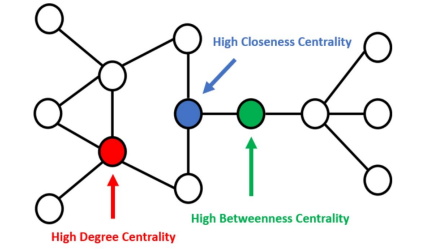
* Degree Centrality
* Closeness Centrality

（接近中心性）是一种衡量图中一个节点与其他节点之间的紧密程度的指标。它反映了一个节点在图中“靠近”其他节点的程度，是图论和网络分析中常用的中心性指标之一.

* Betweenness Centrality

图中所有节点对之间通过节点 的最短路径的数量占所有最短路径数量的比例

* : 节点 到节点 的所有最短路径的数量
* : 节点 到节点 的最短路径中通过节点 的路径数量



1. Edge-level Features

Common neighbors

1. Graph-level Features

Similarity Measures

* Neighbourhood overlap
* Jaccard Coefficient
* Adamic-Adar Index
* Resource Allocation Index

Graphlet（图子结构）

是指嵌入在更大图中的小型无向子图，通常由少量节点和边组成。

**Lecture#3 Graph Analytics**

1. Graph Analytics
2. Node Importance
3. Link Prediction
4. Community Detection
5. Graph Prediction
6. Graph Algorithm
7. BFS
8. DFS
9. Dijkstra Algorithm
10. [PageRank Algorithm](https://ulqqdencc9.feishu.cn/docx/Bu7ydqu3Xogo80x3k2ycdZtanle#share-IJqWdAiaWousgnxLKDmcVr1bnHb)

Limitations:

* Uniform teleportation probability
* Doesn't consider user preferences or context
* Purely link-based and doesn’t consider content quality
* Can be manipulated

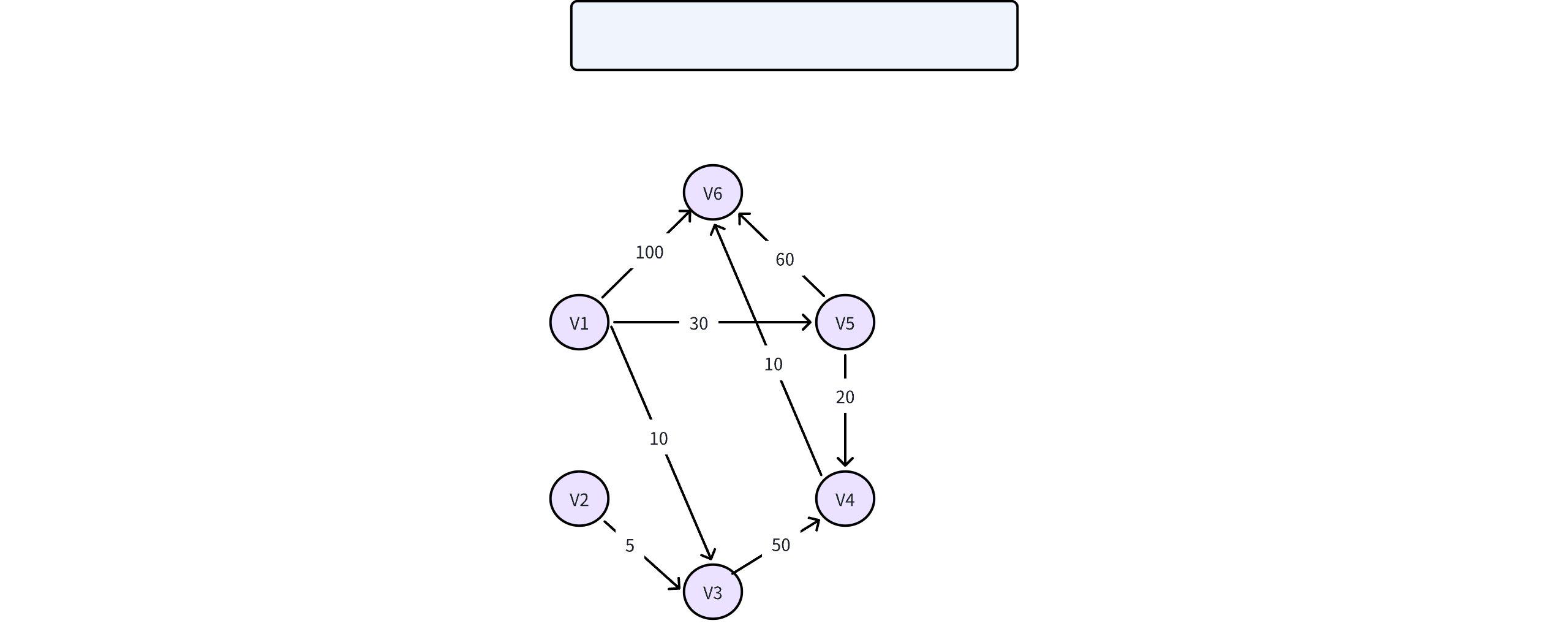
1. NetworkX

Graph Tools in Python [ Comprehensive library for working with graphs in Python ]

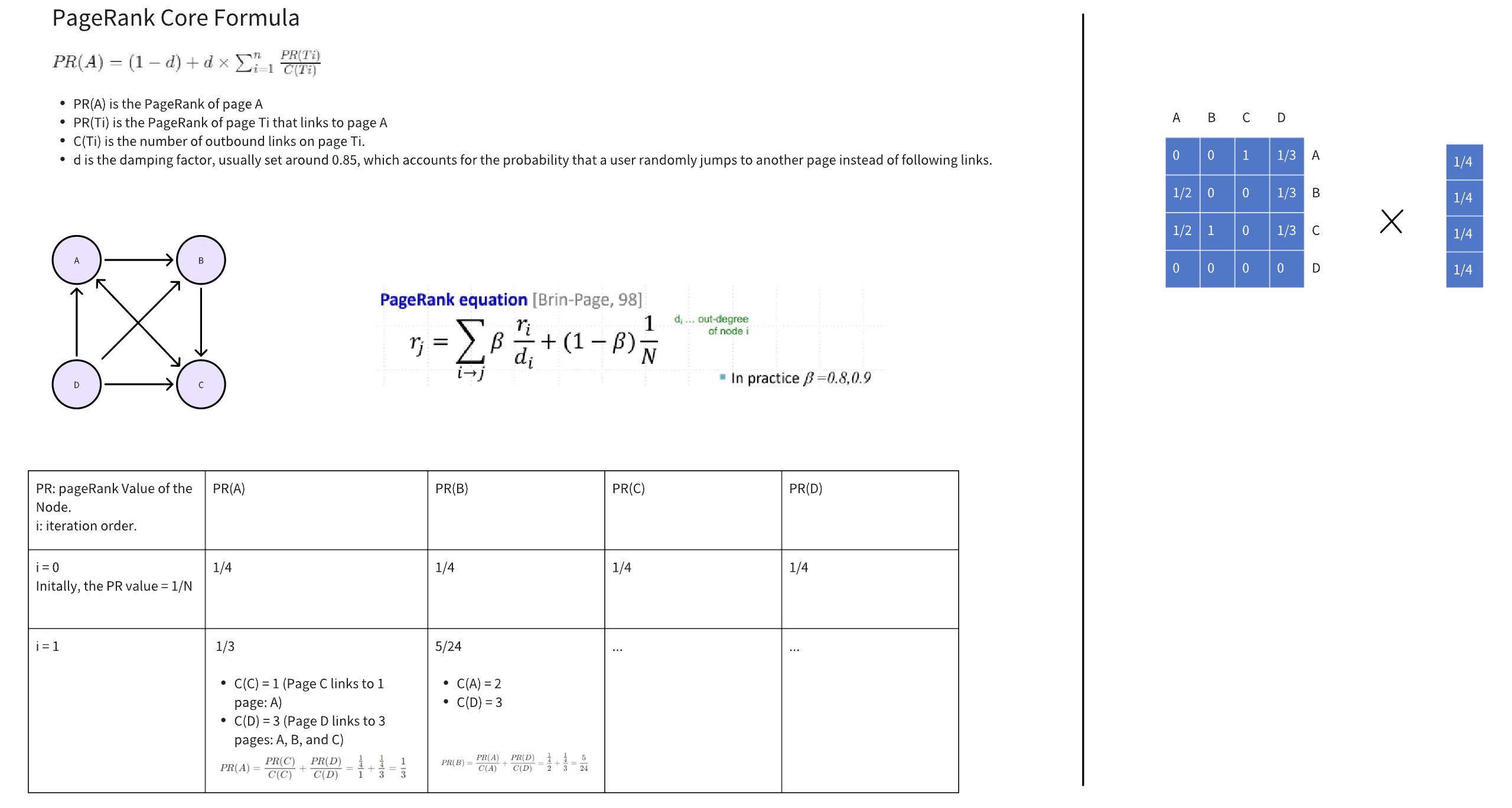
**Dijkstra Algorithm**

Time Complexity:

[最短路径问题---Dijkstra算法详解-CSDN博客](https://blog.csdn.net/qq_35644234/article/details/60870719)



**PageRank Algorithm**



|  |
| --- |
| Plain Text Algorithm PageRank(G, d, epsilon):  Initialize:  N = number of nodes in G  PR[i] = 1/N for all i in G    Repeat:  prev\_PR = PR.copy()    for each node i in G:  PR[i] = (1 - d)/B + d \* sum(prev\_PR[j]/out\_degree(j)) for all j pointing to i)    diff = sum(|PR[i] = prev\_PR[i]| for all i in G)    until diff < epsilon    return PR |

**Lecture#4 Graph Analytics**

* Graph Algorithm

1. Personalized Pagerank

|  |  |
| --- | --- |
|  |  |

1. Pagerank with Restart

Random walk with restart (RWR)

Applications:

* Personalized Search
* Recommendation Systems
* Social Networks
* Graph Big Data
* Graph Data Management
* Neo4J [Graph Database]

**Lecture#5 Graph Learning**

* Graph Learning

Application

* Node classification
* Link prediction
* Community detection
* Graph prediction/generation
* Graph Embedding

**Types of Graph Embedding Methods**

* Random walk-based methods (e.g., DeepWalk, node2vec) 随机游走
* Matrix factorization-based methods 矩阵分解
* Approximately factorize the adjacency matrix:
* Z: Low-dimensional node embeddings
* Objective: Minimize regularization terms.
* Neural network-based methods (e.g., Graph Neural Networks) 图神经网络
* Generative model-based methods
* [Node Embedding](https://ulqqdencc9.feishu.cn/docx/FsS3dfrAQoxMkUxCsz3ctyren6d?from=from_copylink)
* Introduction
* Encoder: maps each node to a low-dimensional vector.

: node in the input graph

: d-dimensional embedding

* Similarity function:
* Algorithm
* Word2vec (NLP)

Two common model architectures

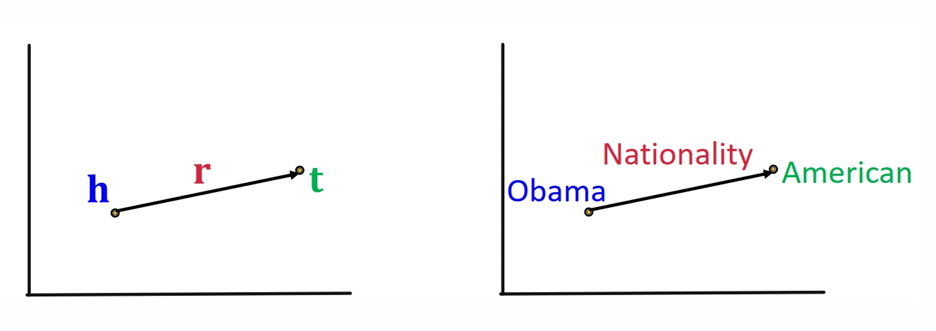
* Continuous Bag-Of-Words (CBOW)
* Skip-gram
* [DeepWalk](https://www.geeksforgeeks.org/deepwalk-algorithm/) [[ Example ]](https://ulqqdencc9.feishu.cn/docx/HKUJdJSKZo39dhxgMSLcZS27nDx?from=from_copylink)
* node2vec

**Recommanding Reading**

|  |  |  |
| --- | --- | --- |
| * Stanford CS224W * Deepwalk * Node2vec * PageRank | * GNN * GCN * Graph-SAGE * GIN | * GAT * Trans-E * Trans-R |

**Lecture#6 Knowledge Graph**

* Basic Concepts of Knowledge Graphs (知识图谱)
* Adding edges' attributions.
* Knowledge Graph Embedding
* TransE
* Basic idea: for a triple
* Score function:



* TransR
* Model entities as vectors in the entity space and model each relation as vector in relation space with as the projection matrix.

个人理解: TransE 和 TransR 的区别在于 TransE 是将 entity 和 relationship 投影到同一个 space，但是 TransR 则将她们投影到不同的 spaces 然后用一个映射的矩阵 M 进行连接.

* Prejection Matrix

, ----

Use to project from entity space to to relation space .

* DistMult
* Entities and relations using vectors in .
* Score function: ,
* Intuition of the score function can be viewed as a cosine similarity between and

个人理解: DistMult 与 TransE 和 TransR 的区别在于改进了 score function.

* Reasoning on Knowledge Graphs
* One-hop Queries
* Path queries
* Conjunctive Queries
* Applications of Knowledge Graphs

**Lecture#7**

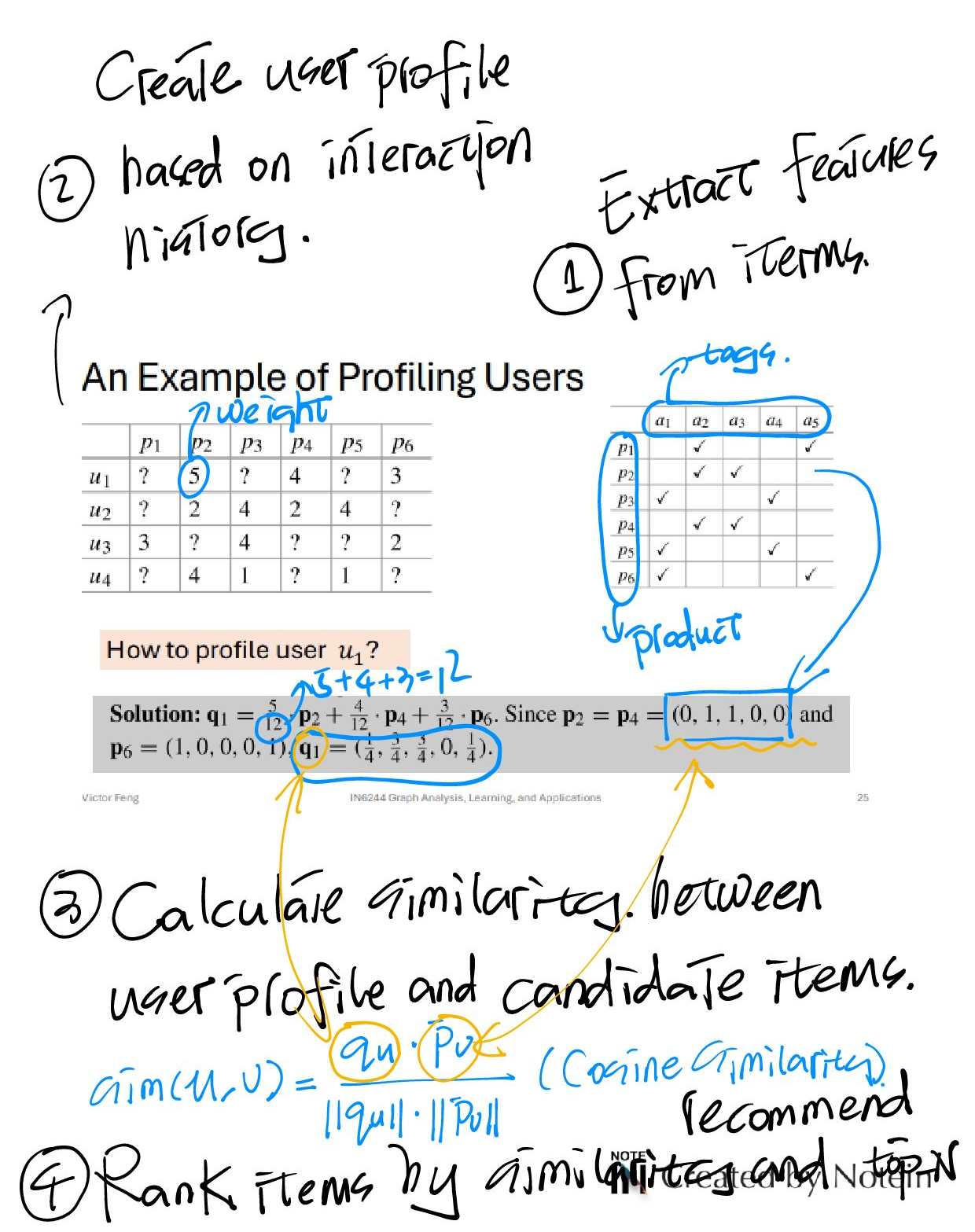
**Lecture#8 Recommender Systems**

1. Introduction of Recommender Systems

Types

* Content-based Filtering "Show me more of the same what I've like". [Similar items]
* Collaborative Filtering "What's popular among my peers". [Similar Peers]
* Knowledge-based Systems "Fits based on my needs."
* Hybrid Systems "Combinations"

1. Content-based Recommendation

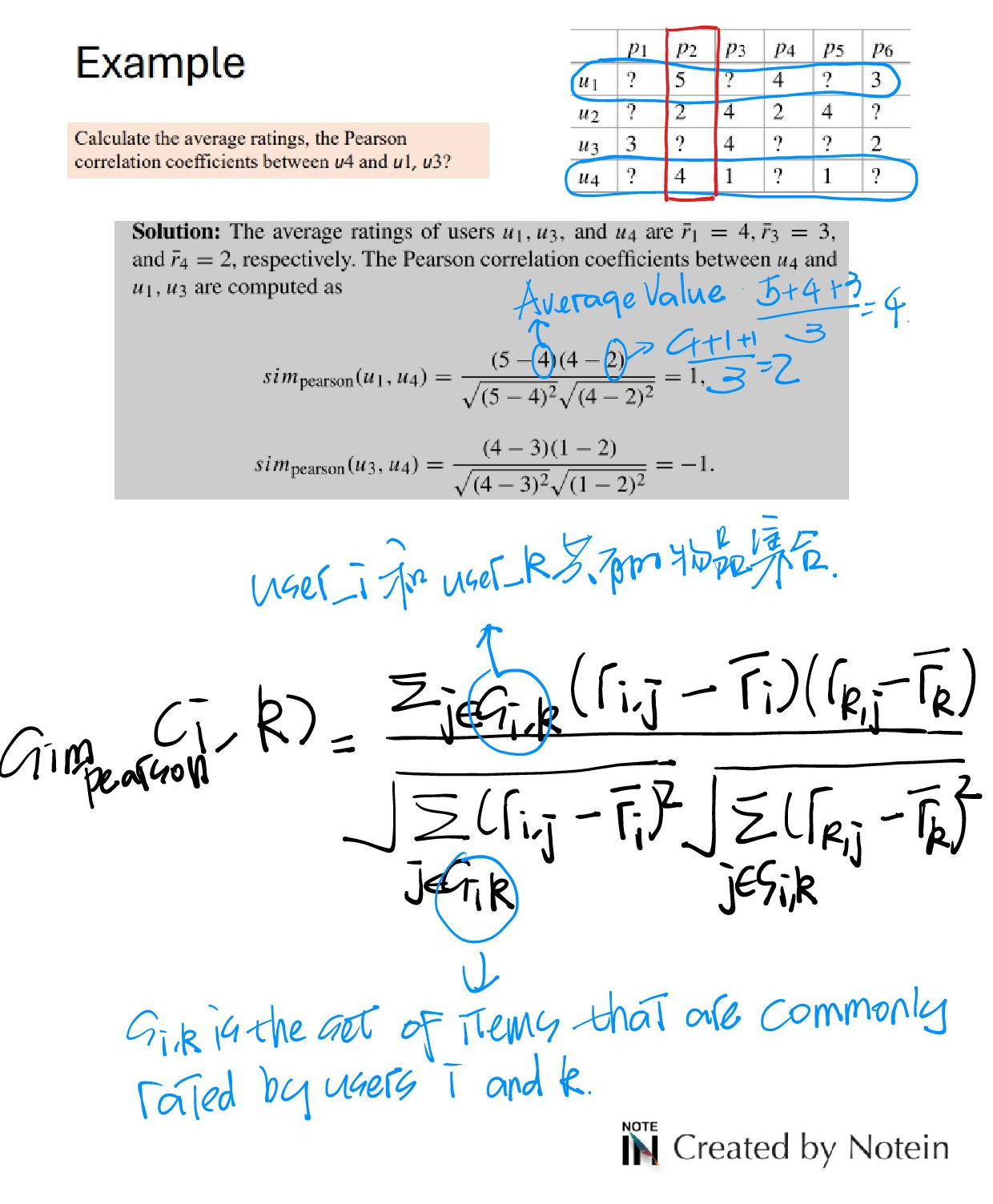


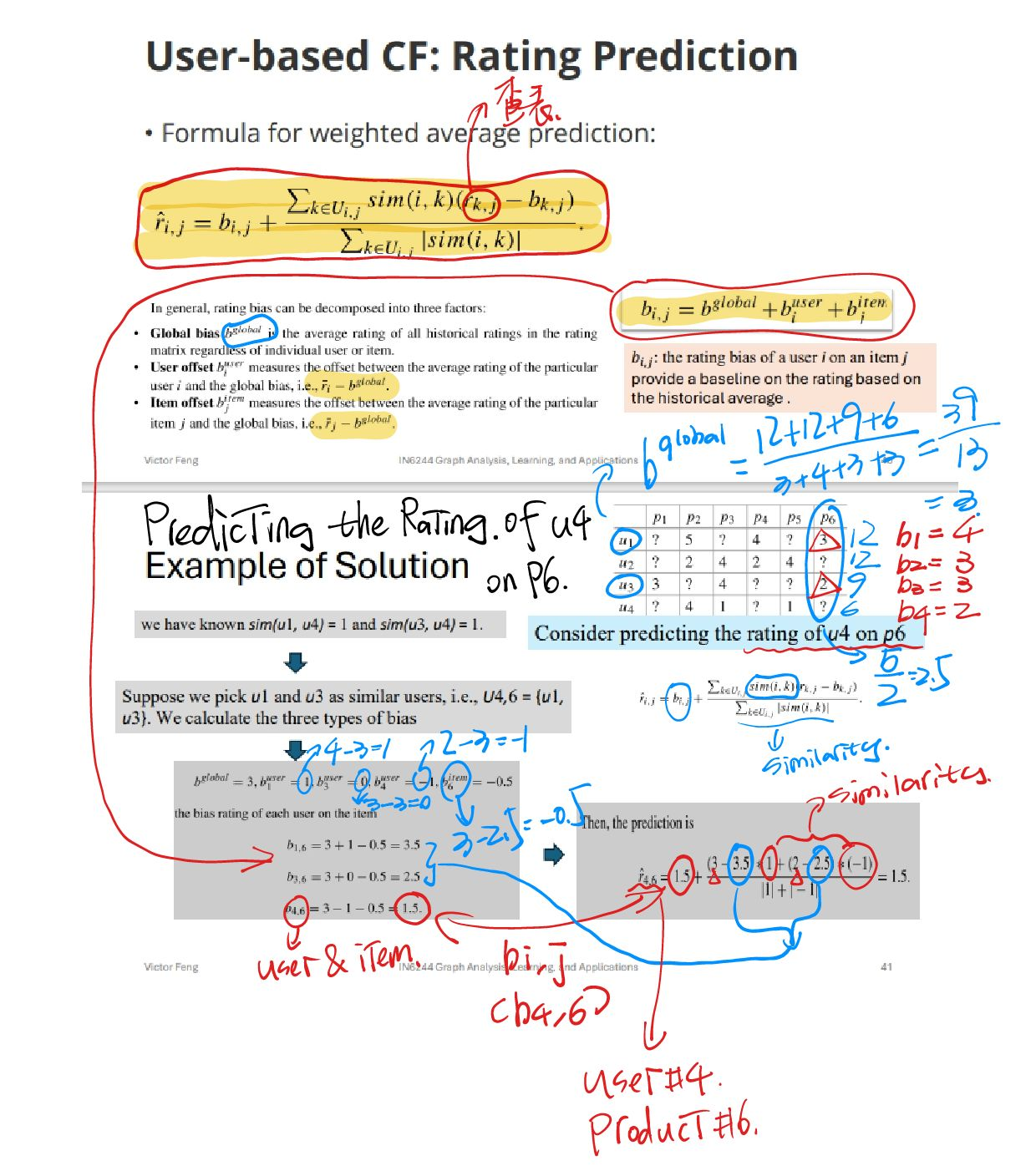
1. Collaborative Filtering
2. **Memory-based CF**
3. User-based CF: Recommends items based on similar users.
4. Item-based CF: Recommends items similar to what the user has already liked.
5. Differences between item-based CF and content-based: item-based CF build an item-item similarity.
6. **Content-Based Recommendation**: Primarily utilizes a **single user's data**, focusing on their preferences and the attributes of the items. It generates recommendations based on what the user has liked in the past.
7. **Item-Based Collaborative Filtering**: Leverages data from **multiple users to find patterns** in item interactions. It identifies relationships between items based on how different users rate or engage with them, thus relying on collective user behavior.
8. **Model-based CF**: Uses machine learning models to learn latent features in user-item interactions.

Key Methods

Matrix Factorization, Factorization Machines, Latent Factor Models, Deep Learning models

1. **User-based CF [Memory-based CF]**

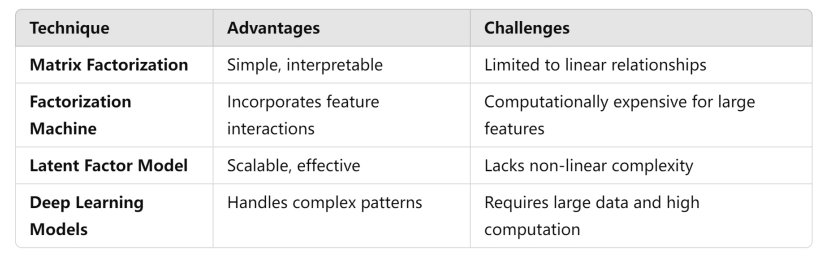




1. **Item-based CF [Memoriy-based CF]**
2. **Matrix Factorization [Model-based CF]**
3. **Factorization Machines [Model-based CF]**

A generalization of matrix factorization that models interactions between all pairs of features, not limited to users and items.

1. **Latent Factor Model [Model-based CF]**
2. **Deep Learning Models [Model-based CF]**



1. Graph-based Algorithms for Recommender Systems

Graph Neural Networks (GNN) for Recommendation

Learning node embeddings through message passing

Popular GNN architectures

* Graph Convolutional Networks (GCN)
* GraphSAGE
* Graph Attention Networks (GAT)

1. Evaluation
2. Accuracy-based Evaluation Metrics

[Best suited for explicit feedback systems (e.g., star ratings)]

* Mean Squared Error (MSE)
* Mean Absolute Error (MAE)

1. Ranking-based Evaluation Metrics P-73

* Precision@K
* Recall@K
* F1 Score
* Mean Reciprocal Rank (MRR)
* Normalized Discounted Cumulative Gain (NDCG)

1. Novelty and Diversity Metrics

* Intra-list Diversity
* Coverage
* Novelty

1. Online Evaluation Techniques

* A/B Testing

1. Applications, Challenges, and Advances

P46

**IN6227 Data Mining**

|  |  |  |
| --- | --- | --- |
| ~~Assignment #1 25%~~  ~~Assignment #2 25%~~  ~~Assignment #3 20%~~  (Individual) | Online Quiz (25%) | ~~Class Participation (5%)~~ |

**Lecture#1 Introduce**

Content for IN6227 Data Mining

1. Classification (Supervised Learning - with ground truth)
2. Regression (Supervised Learning - with ground truth)
3. Assiociation Rule Discovery (no ground truth)
4. Sequential Rule Discovery (no ground truth)
5. Cluster (no ground truth)

**Lecture#2 Data**

Data Type (Attributes - Discrete/Continuous)

1. Nominal (the value of the number has no meaning - ID)
2. Ordinal (Quantitative number - {tall, medium, short})
3. Interval
4. Ratio (meaningful number - length)

Types of data sets (Presenting various dataset samples)

1. Record
2. Graph
3. Ordered (Time Series)

Issues related to data quality

1. Noise
2. Outliers

离群值（或称为异常值）是指在数据集中，其特征与大多数其他数据对象有显著差异的数据对象。这些数据点通常偏离正常范围，可能是由于异常情况、测量误差或特殊条件引起的

1. Missing Values
2. Duplicate Data

Important Characteristics of Structured Data

1. Dimensionality - 维度
2. Sparsity - 稀疏性
3. Resolution - 分辨率

**Data Preprocessing** - 数据预处理 (What have been done in my final project of undergraduate period.)

1. Aggregation

通过对数据进行汇总或组合来简化数据集的过程 - 将每日销售数据汇总为每月或每年的数据，以便更容易观察长期趋势

1. Sampling

从大规模数据集中选择一个子集的过程，这个子集应该能够代表整个数据集的分布和特征 (随机抽样、系统抽样和分层抽样)

[Key Problem: determine the size of sample - Confidence Level]

1. Dimensionality Reduction

减少数据集中特征（维度）数量的过程 (主成分分析（PCA）、线性判别分析（LDA）等)

* Principal Component Analysis (PCA)
* Singular Value Decomposition (SVD)
* Non-linear Dimensionality Reduction - ISOMAP
* Others: supervised and non-linear techniques

1. Feature Subset Selection

是从原始数据集中选择最相关的特征子集，以减少数据的维度和复杂性 (过滤法、包裹法和嵌入法)

* Filter approaches
* Brute-force approach
* Embedded approaches
* Wrapper approaches

1. Feature Creation

通过组合、转换或生成原始数据的特征，来创造出新的、更有意义的特征。这些新特征可以更好地表示数据的模式，从而提高模型的性能。特征创建可以包括数学变换、交互特征的构建，或者根据领域知识生成新的特征

* Feature Extraction
* Mapping Data to New Space
* Feature Construction

1. Discretization and Binarization

离散化是将连续数据转化为离散类别的过程，而二值化是将数据转换为二进制值（0和1）的过程

1. Attribute Transformation

过数学变换或其他方法进行重新表示的过程

* 标准化（将数据缩放到一个特定的范围，如0到1）
* 归一化（将数据转换为单位向量）
* 对数变换（处理具有幂律分布的数据）

( 属性转换可以帮助消除数据中的异常值或极端值，提高模型的稳定性和性能 )



Step of PCA

The core objective of PCA is to find a **projection** that maximizes the variance of the data in a lower-dimensional space.

1. Data Standardization
2. Compute Covariance Matrix
3. Calculate Eigenvalues and Eigenvectors
4. Select Principal Components
5. Project Data

Measure

Similarity and Dissimilarity

1. Simple Attributes

* Euclidean Distance
* Minkowski Distance

(r = 1: Norm; r = 2: Norm; r = ∞ Norm)

* Mahalanobis Distance - 衡量了一个点到数据集均值的距离
* 是待测数据点。
* 是数据集的均值向量。
* 是数据集的协方差矩阵

1. Similarity Between Binary Vectors

* SMC
* Jaccard

1. Cosine Similarity

Entropy - 熵

Euclidean density = number of points per unit volume

**Lecture#3 Data Exploration**

|  |
| --- |
| 3 Topics for this Lecture:   1. **Exploratory Data Analysis (EDA)** 2. **Data Visualisation** 3. **Online Analytical Processing (OLAP)** |

**Summary Statistics**

Summary Statistics are numbers that summarize properties of the data

1. Frequency: count/percentage
2. Location: mean / median / weighted average

Other measures:

1. Spread: standard deviation, range

**Visualization Techniques**

1. Histograms - 直方图
2. Box Plots - 箱线图
3. Scatter Plots - 散点图
4. Contour Plots
5. Matrix Plots
6. Parallel Coordinates
7. Star Plots

**OLAP (Database)**

**Comparation between OLAP & RDBMS**

The difference between **Online Analytical Processing (OLAP)** and **relational databases (RDBMS)**

|  |  |  |
| --- | --- | --- |
|  | **Online Analytical Processing (OLAP)** | **relational databases (RDBMS)** |
| Purpose | 1. Primarily used for **data analysis** and **decision support**, handling historical or **large multidimensional datasets**. 2. It enables users to perform **complex queries** across different dimensions of data to support business decision-making. 3. OLAP is common in business intelligence (BI) systems for reporting, trend analysis, and data mining. | 1. Mainly used for day-to-day transaction processing (OLTP, Online Transaction Processing). |
| Data Structure | Uses a multidimensional data model. Data is organized by dimensions (e.g., time, location, product) and stored in fact tables and dimension tables, forming a "data cube" that allows users to slice and dice data from various perspectives. | Based on a two-dimensional tabular structure where data is stored in rows and columns. |
| Query Methods | Designed for complex, multidimensional analytical queries. Common operations include drill-down, roll-up, slice, and dice, which help users navigate data cubes. OLAP queries often involve aggregate functions like sum, average, max, and min, and require fast response times. | Focuses on simple query operations on rows and columns, suitable for transactional tasks (CRUD: Create, Read, Update, Delete). |
| Summary | Focused on data analysis, multidimensional queries, and is optimized for handling complex analytical tasks over large datasets. | Designed for transaction-based applications, emphasizing data storage, real-time updates, and operational processing. |

**Example**

|  |  |
| --- | --- |
|  |  |

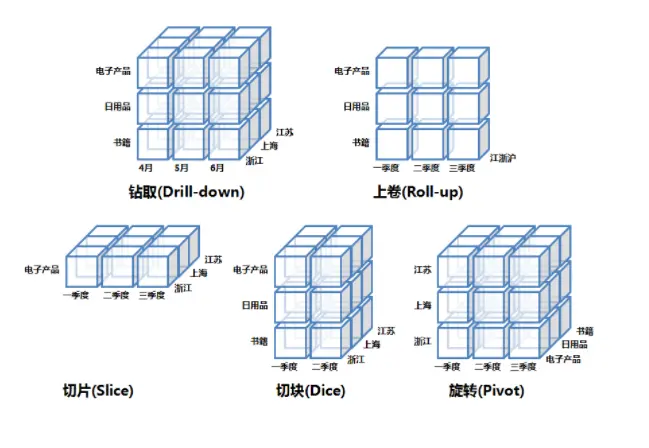
**Operations**

1. Slice
2. Dice
3. Roll-up (聚合)

选定某些维度，根据这些维度来聚合事实

1. Drill-down

选定某些维度，将这些维度拆解出小的维度（如年拆解为月，省份拆解为城市），之后聚合事实



**Lecture#4 Classification**

Classification Techniques

1. Decision Tree-based Methods
2. Rule-based Methods
3. Memory-based reasoning
4. Neural Networks
5. Naïve Bayes and Bayesian Belief Networks
6. Support Vector Machines (SVM)

**Decision Tree-based Methods**

Algorithm

1. [Hunt’s Algorithm](https://ulqqdencc9.feishu.cn/docx/HaAWdd1dkoBVS2xob0ScL8iynLg?from=from_copylink)
2. CART
3. ID3, C4.5
4. SLIQ, SPRINT

**Conceptions Introduction**

**Underfitting and Overfitting Example**

|  |  |
| --- | --- |
|  |  |

**Generalization errors** indicate the error in the real scenario, which means it can be only estimated instead of calculating directly.

Methods for estimating generalization errors:

1. Optimistic approach [ 乐观方法 ]: 假设模型在训练数据上的好表现能够在测试数据上得到类似的结果
2. Pessimistic Approach [ 悲观方法 ]:

Example: ‒ For a tree with 30 leaf nodes and 10 errors in training (out of 1000 instances)

Tip: 0.5 - const value.

1. Reduced Error Pruning, REP [ 减少误差剪枝 ]: 专门用于决策树模型的剪枝技术，旨在减少决策树的复杂性并提高其泛化能力 [ Passed in Slides ]

**Occam's Razor**

**奥卡姆剃刀（Occam's Razor）** 是一种哲学原则和方法论工具，用于在面对多个解释时选择最简单的解释。

**Minimum Description Length (MDL)**

[ 最小描述长度 ]

是一种信息理论中的模型选择方法，用于在众多模型中选择描述数据最简洁的模型。MDL原理基于两个主要的概念：模型的复杂性和数据的拟合度。简而言之，MDL方法试图找到在给定数据的情况下，描述数据所需的总信息量最小的模型。

在MDL方法中，我们的目标是找到一个总成本最低的模型，即最小化下列公式中的总成本：

* : Model cost [ 模型成本 ]
* : Data cost [ 数据成本 ]

**How to address overfitting?**

**Address Overfitting**

* Stop the algorithm before it becomes a fully-grown tree
* Typical stopping conditions for a node:
* Stop if all instances belong to the same class
* Stop if all the attribute values are the same
* More restrictive conditions:
* Stop if expanding the current node does not improve impurity measures (e.g., Gini or information gain).
* Stop if number of instances is less than some user-specified threshold
* Stop if class distribution of instances are independent of the available features

**Handling Missing Values**

......

Finding an optimal decision tree is NP-hard

**Evaluation for Classification Tasks**

1. Metrics for Performance Evaluation

How to evaluate the performance of a model?

1. Methods for Performance Evaluation

How to obtain reliable estimates?

1. Methods for Model Comparison

How to compare the relative performance among competing models?

Overfitting

* What is it?
* How to judge?
* Why occur?
* How to address?

**Lecture#5 Classification**

* [K-Nearest Neighbors](https://ulqqdencc9.feishu.cn/docx/XNTbd9vN1o8DyUxlvmgcXBY6nUg?from=from_copylink) (KNN) - Lazy learning

Core Intuition: Given an unknown point, we assign it to a group by observing what group its nearest neighbors belong to.

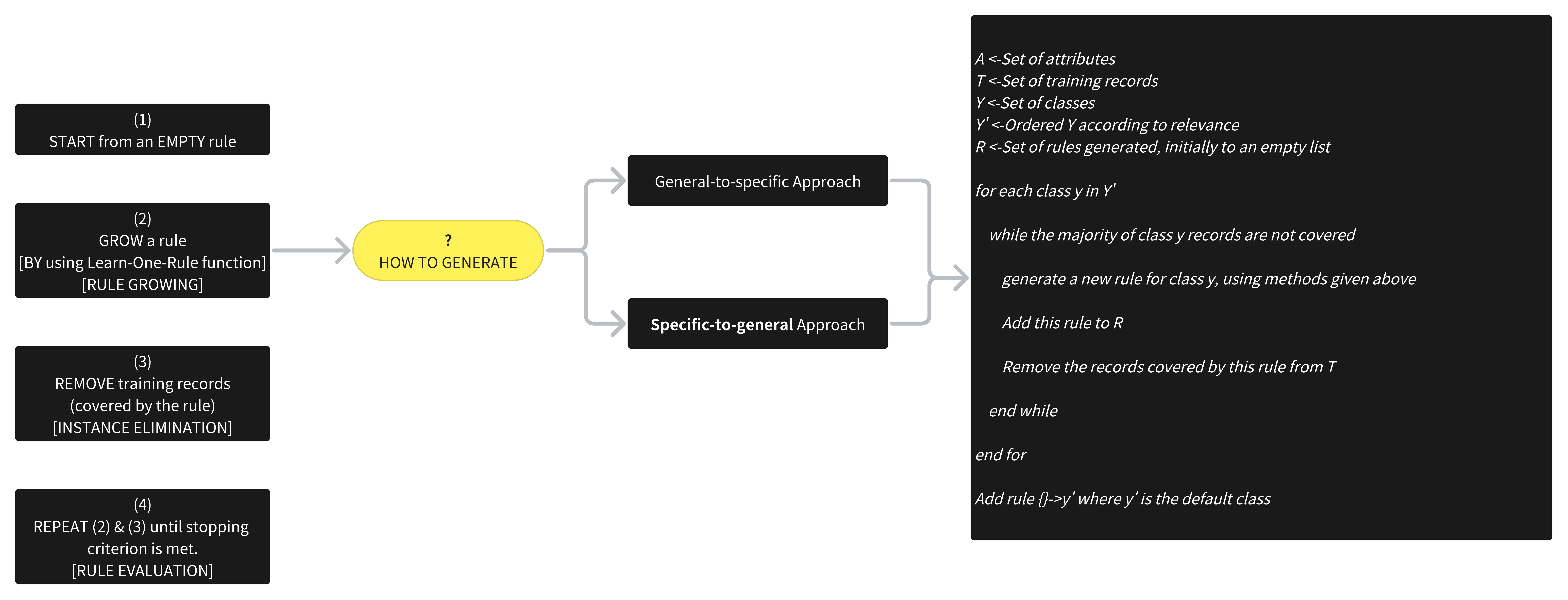
* Rule-Based Classifier - Eager learning

Core Intuition: Generate "IF-THEN" rules for classification task.

Quality of Rules: Coverage and Accuracy

Key Problem:

* HOW TO GENERATE RULES?
* HOW TO APPLY WHEN MULTI-RULES TARGETED? - Give priority [Individual rules are ranked based on their quality]



Rule-Based Classifer

DISADVANTAGE: Not well-suited for handling missing values in the test set.

* [Bayes Classifiers](https://ulqqdencc9.feishu.cn/docx/BDRudx6DFo5RAOxLcHXcPGVjngJ?from=from_copylink) [Probabilistic Classification Models] [More details can be seen at [Conclusion](https://ulqqdencc9.feishu.cn/docx/UeXBdEKhdoL4IJxLrSLc1xd7nqP?from=from_copylink)]
* Ensemble Classifiers
* Bias-Variance Tradeoff
* Bias: Bias refers to the error introduced by overly simplistic models that make strong assumptions about the data. High bias leads to **underfitting**, where the model is too simple to capture the underlying patterns in the data.
* Variance refers to the error introduced by models that are overly complex and sensitive to small fluctuations in the training data. High variance leads to **overfitting**, where the model captures noise and random variations in the training data, rather than the true underlying distribution.
* Bagging (Bootstrap Sampling)

The idea is to train multiple models on different subsets of the training data and then average their predictions (for regression) or use majority voting (for classification) to make the final prediction.

It is designed to reduce the **variance** of machine learning models.

Steps

1. Bootstrap Sampling: Bagging uses **bootstrap sampling**, which is a technique where multiple new training datasets are created by sampling the original training data **with replacement**. [Example: **[A, B, C, D] ----> [A, B, B, D]**]

* Boosting: "Samples that are classified correctly will have their **weights** decreased, otherwise increased"
* Random Forest
* Artificial Neural Networks

|  |  |
| --- | --- |
| * Feedforward Neural Networks，FNN * Recurrent Neural Networks，RNN * Convolutional Neural Networks，CNN * Long Short-Term Memory Networks，LSTM * Generative Adversarial Networks，GANs |  |

* Support Vector Machines
* Nonlinear Support Vector Machines - Do a non-linear transformation

**Lecture#6 Association Analysis**

**Conception**

* **Association Rule Mining**: Given a set of transactions, find rules that will predict the occurrence of an item based on the occurrences of other items in the transaction.
* **Frequent Itemset**:
* Itemset: {Milk, Bread, Diaper}
* Support count
* Support
* Frequent Itemset
* Support
* Confidence

**Algorithm**

* Brute-force
* List all possible association rules
* Compute the support and confidence for each rule
* Prune rules that fail the minsup and minconf thresholds
* Two-step approach
* Frequent Itemset Generation (brute-force

Strategies

* Reduce the number of candidates
* **Apriori** principle: If an itemset is frequent, then all of its subsets must also be frequent. (利用支持度对数据集进行剪枝)
* Reduce the number of transactions
* Reduce the number of comparisons
* Rule Generation (computationally expensive)
* **Apriori Algorithm**

The algorithm leverages the **Apriori Principle**, which states that if an itemset is frequent, all of its subsets must also be frequent. This principle allows the algorithm to prune large portions of the search space, improving efficiency.

* The algorithm starts by scanning the dataset to find itemsets that meet a minimum support threshold, which is the proportion of transactions containing a specific item or itemset.
* It **begins with individual items** and then iteratively explores larger itemsets (pairs, triples, etc.), keeping only those that meet the support threshold.
* Data structure for higher efficiency: Hash Tree

How Hash Trees Work in Association Rule Discovery?

1. **Building the Tree**:

* The algorithm starts by inserting candidate itemsets into the hash tree.
* A hash function (e.g., using the item ID % some number) directs each itemset to a specific branch of the tree.
* This creates a structured storage that speeds up searching.

1. **Counting Itemsets**:

* When scanning a transaction, only relevant parts of the hash tree are traversed, counting only the itemsets that actually appear in the transaction.

1. **Pruning**:

* Hash trees allow pruning of branches that contain itemsets failing to meet minimum support, so only promising paths are explored.
* Factors Affecting Complexity
* Choice of minimum support threshold
* Dimensionality (number of items) of the data set
* Size of database
* Average transaction width
* **FP-Growth Algorithm**

**[FP-Growth.mp4]**

|  |  |
| --- | --- |
|  |  |

**Key Conception**

* The **support** of an itemset is the number of rows (transactions) in which it appears.
* The **minimum support threshold** is set to 5 in this example. This means an itemset is considered "frequent" if it appears in at least 5 transactions (rows).
* A **frequent itemset** is one that meets the minimum support threshold.
* A **maximal frequent itemset** is a frequent itemset that is not contained within any larger frequent itemset. In other words, it is the largest possible subset of items that appears frequently. (e.g., if {A1, A2, A3} is frequent, but {A1, A2, A3, A4} is not, then {A1, A2, A3} might be maximal).
* **Closed Itemset**: An itemset is closed if none of its immediate supersets have the same support.
* Closed Itemset vs Maximal Itemset.

Many real data sets have skewed support distribution.

Multiple Minimum Support : minimum support for item i.

Pattern Evaluation Stage (Problem: Generate too many rules)

{A, B, C} -> {D} and {A, B} -> {D} have same support & confidence.

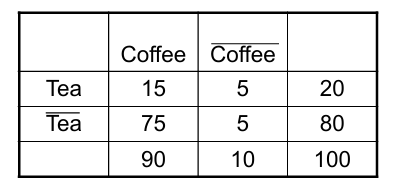
"Interestingness"

Drawback of confidence:

but

Although confidence is high, rules are misleading.

= 0.9375



Measure Table [ P59 ]

**Lecture#7 Advanced Association Analysis**

Outline:

1. Feature Representations
2. Multi-level Association Rules
3. Sequential Pattern Mining
4. Frequent Subgraph Mining

**Asymmetric Binary Attributes** refer to binary variables where the 2 possible values (typically 0 and 1) are not equally important or do not carry the same weight on analysis. For example, in a dataset where a binary attribute represents whether a patient has a disease (1) or does not have the diease (0), the value 1 is much more important for analysis and treatment than the value 0.

Categorical Attributes Representation: transform categorical attributes into asymmetric binary variables.

E.g., Browser - {IE, Chrome, Firefox} ==> "Browser is IE" - {0, 1}, "Browser is Firefox" - {0, 1}, "Browser is Chrome" - {0, 1}

Continuous Attributes Representation

1. [Discretization](https://ulqqdencc9.feishu.cn/docx/Bu7ydqu3Xogo80x3k2ycdZtanle#share-B9tidbVVIontDJxFnulchZaEnIb)

Issues

* Size of the discretized intervals affect support & confidence
* If intervals are too small - itemsets may not have enough support
* If intervals are too large - rules may not have enough confidence

1. Statistics-based
2. Non-discretization based

**minAprior**

Discretization (Continuous Attributes Representation)

1. Equal-width bining

In **equal-width binning**, the continuous attribute's range is divided into a specified number of equal-width intervals (or bins). Each bin has the same width, meaning that the range of values in each bin is constant.

E.g., values ranging from 0 to 100, divide it into 5 equal-width bins: [0–20), [20–40), [40–60), [60–80), and [80–100]

1. Equal-depth bining

In **equal-depth binning**, the continuous attribute is divided into bins such that each bin contains approximately the same number of data points. Unlike equal-width binning, the width of the intervals can vary.

E.g, [3, 5, 8, 12, 15, 18, 22, 30, 35, 40]

Since we want 2 bins, and there are 10 values, each bin should contain 5 values.

* Bin 1: [3, 5, 8, 12, 15].
* Bin 2: [18, 22, 30, 35, 40].

1. Clustering

In **clustering-based discretization**, clustering algorithms (such as k-means) are used to group continuous values into clusters or "bins" based on their similarity. Each cluster can then be treated as a bin for discretization.

P14 - 16

Statistics-based Methods

The slides are discussing statistics-based methods for determining whether an association rule is interesting, particularly in the context of continuous variables (like age).

1. Understanding Association Rules with Continuous Variabless
2. Approach to Apply Statistics

To evaluate how "interesting" the rule is, they suggest a systematic approach

Approach to Apply Statistics:

* **Withhold the target variable** (in this case, age) from the dataset initially. 隐藏目标变量（从数据集中暂时排除目标变量）
* Apply standard **frequent itemset mining** to discover rules about categorical variables. 生成频繁项集（frequent itemset）在没有目标变量（年龄）的数据上，应用常规的**频繁项集挖掘**算法，生成一组频繁项集。比如从用户浏览器（Chrome）和操作系统（Mac）这样的变量中挖掘出常见的组合。
* For each rule generated, calculate **descriptive statistics** for the target variable (here, the mean age). 比如说，假设在频繁项集中我们得到了“使用 Chrome 且操作系统为 Mac 的用户”，可以计算这些用户的平均年龄是多少（比如23岁）

In IN6208, we learn "decriptive statistics" (e.g., mean, median, mode, standard deviation) & "inferential statistics" (t-test, ANOVA, chi-square test.)

* The result is an association rule where the **frequent itemset** (like "Browser = Chrome ∩ OS = Mac") is associated with the target variable (Age), characterized by its statistics (like the mean age). 生成完整的关联规则. 最后，将目标变量（年龄）重新引入。此时，频繁项集会与目标变量的统计值结合，形成完整的关联规则
* Determine the Interestingness of a Rule: To assess whether an association rule is **interesting**, we compare the statistics (e.g., mean age) of the group covered by the rule vs. the group **not covered** by the rule. 最后，使用统计检验（比如 z-test）来评估规则是否有趣。通过比较规则覆盖的用户群体和未覆盖的用户群体的目标变量（年龄）的统计值差异，确定这个规则是否具有显著性差异，从而判断它是否“有趣”。
* Z-test: a statistical test to determine whether 2 population means are different

Example: P15 -16

**Lecture-8 Clustering**

1. Clustering
2. **Partitional Clustering** vs **Hierarchical Clustering**
3. Several distinctions between different types of clustering methods
4. Non-exclusive vs exclusive (non-overlap, ONE point --> ONE cluster)
5. Fuzzy vs Non-fuzzy

In fuzzy clustering, each data point can belong to every cluster with a degree of membership (a value between 0 and 1), and the sum of the membership values across all clusters must equal 1. This allows for more flexible cluster assignments.

Non-fuzzy: Each data point is assigned to exactly one cluster with no ambiguity, similar to exclusive clustering.

1. Partial vs Complete

**Partial Clustering**: Sometimes, we only want to cluster a subset of the data rather than the entire dataset.

**Complete Clustering**: All data points are assigned to clusters, and no data points are left unclustered.

1. Heterogeneous vs Homogeneous

**Heterogeneous Clustering**: Clusters can contain data points of very different sizes, shapes, and densities, allowing for more diversity within clusters.

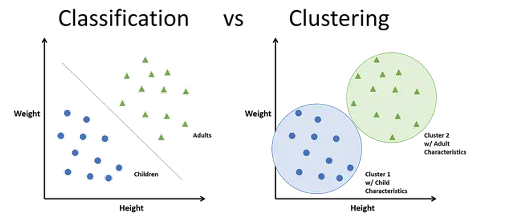
**Homogeneous Clustering**: Clusters are more uniform, containing data points that are more similar to each other in terms of size, shape, and density.

1. Types of Clusters
2. Well-separated clusters P11
3. Center-based clusters P12
4. Contiguous clusters
5. Density-based clusters
6. Property or Conceptual
7. Described by an Objective Function
8. Clustering Algorithms
9. K-means

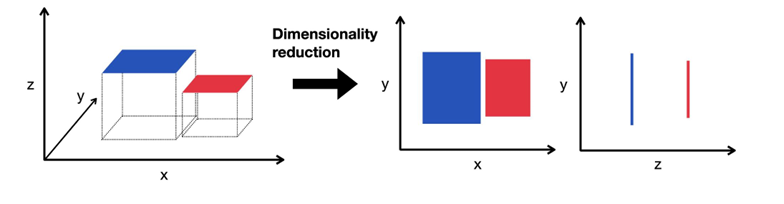
* Pre-processing P34
* Normalize the data
* Eliminate outliers
* Post-processing
* Bisecting K-means P 35
* Limitations P 38

1. Hierarchical clustering
2. Density-based clustering

Classification (supervised Learning): have class label information.



Dimensionality Reduction (e.g., PCA, t-SNE)



**Partitional Clustering** （划分聚类）vs **Hierarchical Clustering** （层次聚类）

1. **Partitional Clustering**

Partitional clustering is a method where the dataset is divided into **non-overlapping subsets (clusters)** such that each data point belongs to exactly one cluster. The goal is to optimize a given criterion (such as minimizing the sum of squared distances within clusters) by assigning points to clusters iteratively until convergence.

划分聚类是一种将数据集划分为\*\*非重叠子集（簇）\*\*的方法，其中每个数据点只属于一个簇。其目标是通过迭代地将数据点分配到不同簇中，优化某个标准（如最小化簇内平方距离之和），直到达到收敛 (**K-means**， where the number of clusters K is predefined, and the algorithm minimizes the distances between data points and the centroid of each cluster.)

1. Hierarchical Clustering

Hierarchical clustering builds a tree-like structure (dendrogram) of clusters, where each data point starts as its own cluster, and clusters are merged (or split) step by step. This process continues until a single cluster or a desired number of clusters is formed. Hierarchical clustering can be **agglomerative** (bottom-up) or **divisive** (top-down).

层次聚类构建一个类似**树形结构（树状图）的簇，每个数据点最初作为自己的簇，然后逐步将簇合并（或拆分）。这一过程一直持续到形成一个簇或达到预定的簇数量。层次聚类可以是凝聚型**（自下而上）或**分裂型**（自上而下）

Hierarchical Clustering

1. Agglomerative (Bottom-up) Clustering

* The process starts with each data point as its own individual cluster.
* 该过程从每个数据点作为一个独立的簇开始
* Then, it repeatedly merges the closest pairs of clusters based on a similarity metric (e.g., Euclidean distance) until all data points are grouped into a single large cluster or until a stopping criterion is met (such as a specified number of clusters).
* 根据某种相似性度量（例如欧几里得距离），反复将最近的簇对合并，直到所有数据点被分组为一个大的簇，或者达到指定的停止标准
* The result is a dendrogram, which shows the hierarchy of clusters and the points where they were merged.
* 最终结果是一个 **树状图（dendrogram）**

1. Divisive (Top-down) Clustering

* The process starts with all data points in a single cluster.
* 该过程从将所有数据点放入一个簇开始
* It then recursively splits the clusters into smaller subclusters until each data point is its own cluster or until a stopping criterion is met.
* 然后递归地将簇分割成更小的子簇，直到每个数据点成为一个簇，或达到指定的停止标准

**IN6208 Research Method**

**IN6206 Internet Programming**

|  |  |  |
| --- | --- | --- |
| **~~Term Paper~~**  ~~Individual~~  ~~30%~~ | **Technology Research Assignment**  Individual  20% | **~~Web Application Development Mini-Project~~**  ~~Group~~  ~~50%~~ |

**Mini Web App**

Eclipse/NetBeans + Web 2.0/3.0

[Internet Programming选题](https://ulqqdencc9.feishu.cn/docx/Z69YdJg2toZEu6xwnQzciX79nmf?from=from_copylink)

**IN6231 Security Policy & Strategy**

|  |  |  |
| --- | --- | --- |
| **Group Assignment (Presentation)**  Group  30% | **Assignment Report**  Individual  50% | **Test**  Individual  20% |