Course Information

Course: CS 570 Data Mining
Instructor: Dr. Wei Jin
University: Emory University
Semester: Spring 2025
Author: Xinliu Zhong

1 Preprocessing

Data integration *Redundancy*

1. **Chi-Square (χ^2) Test (Categorical)**: $\chi^2 = \sum_i^n \frac{(O_i - E_i)^2}{E_i}$
   H_0: Variable distributions are independent. Higher χ^2 suggests correlation (**not** causality).
   df = (#row - 1)(#col - 1)
2. **Variance & Covariance (Numerical)**: $\hat{\sigma}^2 = \frac{1}{n}\sum(x_i - \hat{\mu})^2$,
   $\hat{\sigma}_{12} = \frac{1}{n}\sum(x_{i1} - \hat{\mu}_1)(x_{i2} - \hat{\mu}_2)$
   Expectation form: $\sigma_{12} = E[X_1X_2] - E[X_1]E[X_2]$
3. Covariance matrix $\sigma_{12} = E[(X_1 - \mu_1)(X_2 - \mu_2)^T]$
4. **Pearson Correlation**: Normalized Covariance, range: [-1,1]. Linear relation only. $\hat{\rho}_{12} = \frac{\hat{\sigma}_{12}}{\hat{\sigma}_1\hat{\sigma}_2} = \frac{\sum(x_{i1}-\hat{\mu}_1)(x_{i2}-\hat{\mu}_2)}{\sqrt{\sum(x_{i1}-\hat{\mu}_1)^2\sum(x_{i2}-\hat{\mu}_2)^2}}$

1.1 Data Reduction

- *Parametric*: Regression
- *Non-parametric*: Histogram, Clustering, Sampling

1.2 Data Transformation

Min-Max: $v' = \frac{v - min_A}{max_A - min_A}(new\_max - new\_min) + new\_min$
Z-Score: $v' = \frac{v - \mu_A}{\sigma_A}$
Decimal Scaling: $v' = \frac{v}{10^j}$, where j is the smallest integer s.t. $\max(|v'|) < 1$

2 Dimensionality Reduction

Curse of Dimensionality
Data becomes sparse, distance/density lose meaning, subspace combinations grow exponentially: $1 - (1 - \epsilon)^d$

PCA (Principal Component Analysis)
Unsupervised, linear. Maximizes variance preservation: $L = g^T Sg - \lambda(g^T g - 1)$

Sample covariance matrix: $S = \frac{1}{N-1}\sum_{i=1}^N (x_i - \bar{x})(x_i - \bar{x})^T$

Find first p eigenvectors $\{g_i\}_{i=1}^p$ from: $|S - \lambda I| = 0, \quad Sg = \lambda g$

Construct transformation matrix $G = [g_1, g_2, \ldots, g_p]$ and transform: $x \in \mathbb{R}^d \to G^T x \in \mathbb{R}^p$

Attribute Subset Selection
Choose best attribute by significance tests under independence assumption. Selection or Elimination.

3 Pattern Mining

3.1 Concept

Support (Count): Freq. of itemset X in transactions. **Relative Support**: Fraction of transactions containing X. An itemset X is *frequent* if X's support is no less than a minimum support threshold.
Frequent Itemset: X is frequent if $\text{Supp}(X) \geq$ min support. **Association Rule**: $X \to Y$, where X, Y are itemsets.

$$\text{Supp}(X \to Y) = \frac{|T(X \cup Y)|}{|T|}$$

$$\text{Conf}(X \to Y) = \frac{|T(X \cup Y)|}{|T(X)|}$$

3.2 Apriori

BFS, iterative DB scans.
Downward Closure: Any *subset* of a frequent itemset must be frequent. So we can prune *supersets* of itemset found infrequent. $\forall X, Y : (X \subseteq Y) \Rightarrow supp(X) \geq supp(Y)$
Candidate generation: a. $F_{k-1} \times F_1$ b. $F_{k-1} \times F_{k-1}$
Rule generation: low confidence rule $\text{conf}(ABC \to D) \geq \text{conf}(AB \to CD) \geq \text{conf}(A \to BCD)$

3.3 FPGrowth

DFS, 2 scans. Grow long patterns from short ones using **local** frequent items. 1. **FPTree Construction**: Find frequent 1-itemsets, sort into f-list, insert transactions into FP-Tree maintaining freq. counts. 2. **FPTree Mining**: Construct conditional FP-Tree for every suffix until only a single path remains.

3.4 Evaluation

$Lift(X \to Y) = \frac{p(X \cup Y)}{p(X)p(Y)}$ Lift($X \to Y$) vs. Lift($X \to \neg Y$)
Closed Pattern: X is frequent and has no super-pattern $Y \supset X$ with the same support. **Max-Pattern**: X is frequent and has no frequent super-pattern $Y \supset X$.

3.5 Sequential Pattern Mining

Sequence Data: sequence $\to$ element (transaction) $\to$ event (item). **k-sequence**: A sequence with k events/items. **Support of subsequence** w: Fraction of data sequences that **contain** w. **Generalized Sequential Pattern (GSP) Algorithm** $\sim$ Apriori: Merging $k-1$ patterns to form k-sequences: - Merge two frequent $(k-1)$-sequences w_1, w_2 if the first event of w_1 matches the last event of w_2 after deletion. - The resulting k-sequence extends w_1: - If the last two events in w_2 share an element, append its last event to w_1's last element. - Else, append it as a new element.

4 Similarity Search

4.1 Proximity measure

Minkowski Distance: $dist = \left(\sum_{k=1}^n |p_k - q_k|^r\right)^{\frac{1}{r}}$ - $r = 1$ (Manhattan, L_1 norm), $r = 2$ (Euclidean), $r \to \infty$ ($L_{\max}$ norm).
Mahalanobis Distance: Penalizes covariance matrix.
SMC vs. Jaccard: $SMC = \frac{M_{11}+M_{00}}{M_{11}+M_{00}+M_{10}+M_{01}}$, $J = \frac{M_{11}}{M_{11}+M_{10}+M_{01}}$ - One-hot encoding $\to$ Jaccard.
- Cosine Similarity. - Correlation: standardize $\to$ dot product (fails for nonlinear).

4.2 Similarity Search

Collaborative Filtering:
User-based NN: $pred(u,i) = \bar{r}_u + \frac{\sum_{n \in neighbors(u)} sim(u,n) \cdot (r_{ni} - \bar{r}_n)}{\sum_{n \in neighbors(u)} sim(u,n)}$
Item-based NN: $pred(u,i) = \frac{\sum_{j \in ratedItems(u)} sim(i,j) \cdot r_{uj}}{\sum_{j \in ratedItems(u)} sim(i,j)}$
k-D Trees: Partition data recursively: 1. Choose dim. w/ max variance. 2. Split at median. 3. Partition into two halves. 4. Repeat for next highest variance.

4.3 Hashing for Dim Reduction

Locality Sensitive Hashing (LSH) Similar items $\Rightarrow$ High probability of same bucket
MinHash: Increases the probability of collision when documents are similar. Generate a **random permutation** of the columns. For data d_i, the MinHash value is the **first permuted column** that has a "1". $\Rightarrow$ Reduces the number of pairwise comparisons.

5 Classification

5.1 Decision Tree

Induction Algorithm (overfitting $\to$ pre- or post-pruning), **Prediction:** Majority voting
Entropy (Expected information needed to classify a tuple in D): $Info(D) = -\sum_{i=1}^m p_i \log_2(p_i)$ **Information needed** after splitting D using attribute A: $Info_A(D) = \sum_{j=1}^v \frac{|D_j|}{|D|} \times Info(D_j)$ **Information Gain** (reduction in entropy): $Gain(A) = Info(D) - Info_A(D)$
Stopping Conditions: All nodes belong to same class, no attributes left, or no samples left.

5.2 Bayesian classification

Generative classifier $P(Y, X)$, incremental (incorporate prior knowledge)
Naïve Assumption: Attributes are conditionally independent
Categorical: $p(X|C_i) = \prod_k p(x_k|C_i)$
Continuous-valued $p(x_k = v_k|C_i) = N(x_k|\mu_{C_i}, \sigma_{C_i}) = \frac{1}{\sqrt{2\pi\sigma_{C_i}^2}} e^{\frac{-(x-\mu_{C_i})^2}{2\sigma^2}}$

5.3 Linear Classifier

Discriminative Classifier $P(Y|X)$
Logistic Regression Turns linear predictions into probabilities.
Sigmoid function: $S(x) = \frac{1}{1+e^{-x}} = \frac{e^x}{e^x+1}$ **Likelihood Function**: $l(w) = \sum_{i=1}^N y_i \log p(Y=1|x_i; w) + (1 - y_i)\log(1 - p(Y=1|x_i; w))$
Optimization: No closed-form solution $\to$ Use gradient descent $a_{n+1} = a_n - \gamma \nabla F(a_n)$.

kNN: Non-parametric, lazy learner.

5.4 Bayesian belief networks

Chain Rule of Probability: Prob. factorization based on Directed Acyclic Graph (DAG). $P(x_1, x_2, ..., x_N) = \prod_i P(x_i|Parent(x_i))$

5.5 SVM

Maximize margin $\frac{2}{\|\mathbf{w}\|} \to$ Minimize $E(\mathbf{w}) = \frac{\|\mathbf{w}\|^2}{2}$ Subject to:

$$y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1, \quad i = 1, 2, \ldots, N$$

Lagrange Formulation

$$L(\mathbf{w}) = \frac{\|\mathbf{w}\|^2}{2} - \sum_i \lambda_i [y_i(\mathbf{w}^\top \mathbf{x}_i + b) - 1]$$

Derivatives:
$\frac{\partial L}{\partial \mathbf{w}} = 0 \Rightarrow \mathbf{w} = \sum_i \lambda_i y_i \mathbf{x}_i, \quad \frac{\partial L}{\partial b} = 0 \Rightarrow \sum_i \lambda_i y_i = 0$
Constraints:

$$\lambda_i \geq 0, \quad \lambda_i [y_i(\mathbf{w}^\top \mathbf{x}_i + b) - 1] = 0$$

(Complementary slackness)
Quadratic Programming:
$\max_\lambda L(\lambda) = \sum_i \lambda_i - \frac{1}{2}\sum_{i,j} \lambda_i \lambda_j y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j), \quad \lambda \geq 0$
Support Vectors: Points where $\lambda_i \neq 0$.
Kernel Trick
Gain linear separation by mapping to a higher dimension:
Applying Kernel:

$$L_d = \sum \alpha_i - \frac{1}{2}\sum \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$$

1 Clustering

Unsupervised learning.

1.1 Partitioning Approach

K-means: Each cluster is represented by the mean of its points.

Algorithm: Select K initial centroids. **Repeat:** a. Assign each point to the closest centroid. b. Update centroids as cluster means. **Until** centroids don't change.

Complexity: $O(nKId)$ (n: points, K: clusters, I: iterations, d: dimensions) **P1. initialization of centroids** $k-means++$: For each point x_i, compute its distance to the nearest centroid: $d(x_i, C)^2 = \min_{x_c \in C} d(x_i, x_c)^2$ Choose the next centroid x_k with probability: $P(x_k) = \frac{d(x_k,C)^2}{\sum_i d(x_i,C)^2}$

P2. Choosing K: **Cross-validation**.

P3. Non-globular clusters → Use many clusters.

P4. Sensitive to outliers → Use **K-Medoids** (medoid instead of mean).

P5. Empty clusters → Select the point contributing most to SSE as a new centroid.

*Bisecting K-Means (Hierarchical) Split the cluster using basic K-means. → Add resulting clusters with the highest SSE.

1.2 Hierarchical Approach

Irreversible, $O(N^2)$ space, $O(N^3)$ time.

Divisive (Top-Down): Better global structure, early stopping avoids over-clustering. *(Can use K-means for efficiency)*

Agglomerative (Bottom-Up): Clear merging patterns, useful for structured data.

Inter-Cluster Similarity:

MIN (Single-Linkage): Handles non-elliptical shapes, sensitive to noise/outliers.

MAX (Complete-Linkage): Less noise-sensitive, splits large clusters, favors globular clusters.

Group Average: Reduces noise impact, favors globular clusters.

Centroid-Based Distance: Noise-robust, simple, not suitable for categorical data.

1.3 Density-Based Approach

Clusters based on density reachability, detecting outliers and non-globular shapes. **DBSCAN:** Label points as **core**, **border**, or **noise**. Remove noise, connect core points within ϵ, assign border points to nearest core cluster. **Parameters: Eps** (radius), **MinPts** (min neighbors). **Properties:** Noise-robust, handles arbitrary shapes, sensitive to Eps/MinPts. Sort points by k-th nearest neighbor distance, use elbow method for optimal ϵ.

1.4 Clustering Evaluation

Extrinsic (Supervised) BCubed Precision & Recall measure correctness of object relations: $\text{Precision}(o_i) = \frac{\#\text{objects in } C(o_i) \text{ with } L(o_i)-1}{\#\text{objects in } C(o_i)-1}$, $\text{Recall}(o_i) = \frac{\#\text{objects in } C(o_i) \text{ with } L(o_i)-1}{\text{Total } \#\text{ objects with } L(o_i)-1}$. BCubed Precision and Recall: $\frac{1}{n}\sum_{i=1}^{n} \text{Precision}(o_i)$, $\frac{1}{n}\sum_{i=1}^{n} \text{Recall}(o_i)$.

Intrinsic (Unsupervised) Compactness (Intra-cluster Distance): $a(o) = \frac{\sum_{o' \in C_i, o \neq o'} dist(o,o')}{|C_i|-1}$, measures how close o is to other points in the same cluster. Lower is better.

Separateness (Inter-cluster Distance): $b(o) = \min_{C_j : j \neq i} \frac{\sum_{o' \in C_j} dist(o,o')}{|C_j|}$, measures how far o is from the closest other cluster. Higher is better.

Silhouette Score: $s(o) = \frac{b(o)-a(o)}{\max(a(o),b(o))} \in [-1,1]$. $s(o) \approx 1$: Well-clustered, $s(o) < 0$: Incorrect clustering.

2 Outlier Detection

Types: Global, contextual (attribute-based), collective.

2.1 Statistical Methods

Parametric: Probability density function, Grubb's test.

Univariate: Fit normal distribution, outliers deviate by $\geq 3\sigma$. Grubb's Test: $z = \frac{|x-\mu|}{\sigma}$, $z \geq \frac{N-1}{\sqrt{N}} \times \frac{t_{\alpha/(2N),N-2}}{\sqrt{N-2+t^2_{\alpha/(2N),N-2}}}$

Multivariate: Mahalanobis Distance accounts for covariance: $MDist(o,\delta) = \sqrt{(o-\delta)^\top \Lambda (o-\delta)}$

2.2 Non-Parametric Methods

Histogram and Kernel Density Estimation (KDE) Gaussian kernel smoothing: $K(u) = \frac{1}{h\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{u}{h}\right)^2}$, Density estimation for dataset $\{x_1, x_2, ..., x_N\}$: $\hat{f}(o_i) = \frac{1}{N}\sum_{j=1}^{N} K\left(\frac{o_i - x_j}{h}\right)$

2.3 Proximity-based Methods

Distance-based: An object is an outlier if $\leq k$ objects exist within its r-neighborhood. $k = \pi \cdot |D|$. **Density-Based:** k-distance $\text{dist}_k(o)$: Distance from o to its k-th nearest neighbor. $N_k(o)$: Set of objects within k-distance. **Local Outlier Factor (LOF)** Density: $d(o) = \frac{|N_k(o)|}{\sum_{o' \in N_k(o)} \max(\text{dist}(o,o'), \text{dist}_k(o'))}$ LOF: $LOF(o) = \frac{1}{|N_k(o)|}\sum_{o' \in N_k(o)} \frac{d(o')}{d(o)}$ LOF ≈ 1: Not an outlier. LOF $\gg 1$: Local outlier.

2.4 Clustering-based Methods

DBSCAN: Detects objects not belonging to any cluster. **Distance-based Outliers:** Score $\propto \frac{\text{dist}(o,c_n)}{\text{avg dist}(c_n)}$. **CBLOF (Cluster-Based Local Outlier Factor):** Sort clusters by size. CBLOF: $|C| \times \text{similarity}(p, C_{\text{closest}})$. Example: If p in small cluster C_3 has low similarity to its closest large cluster C_2, then p is an outlier.