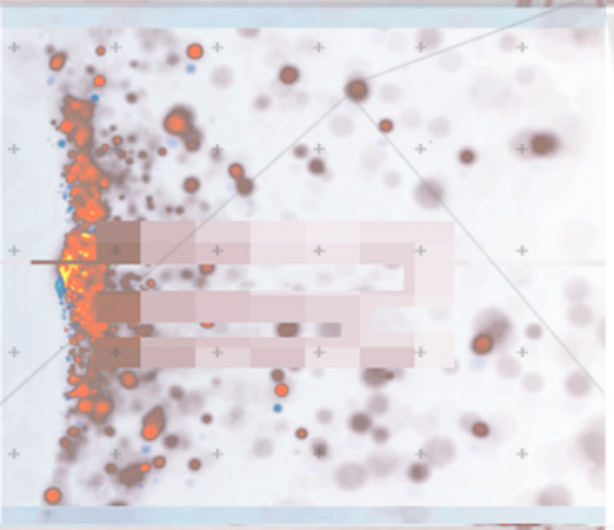
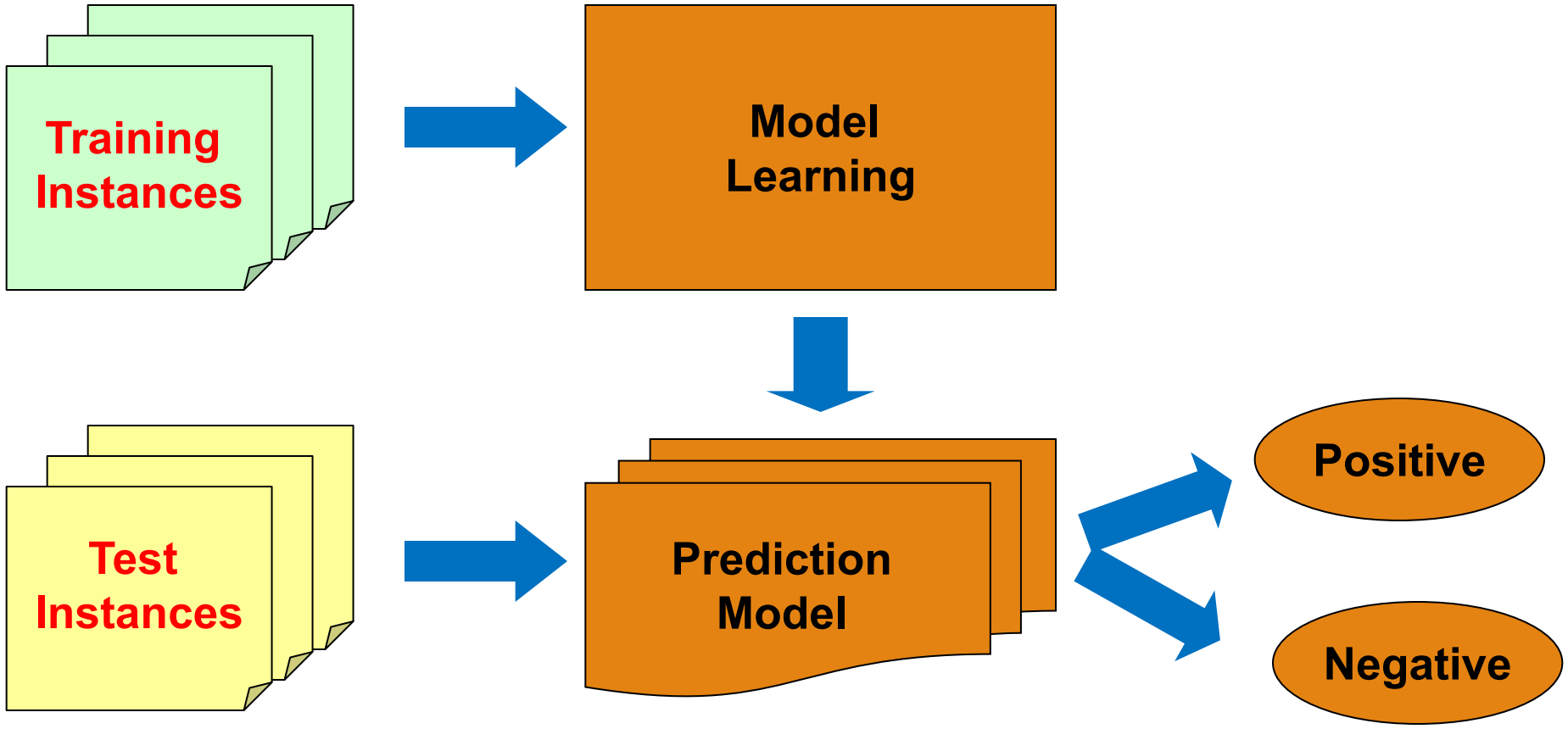
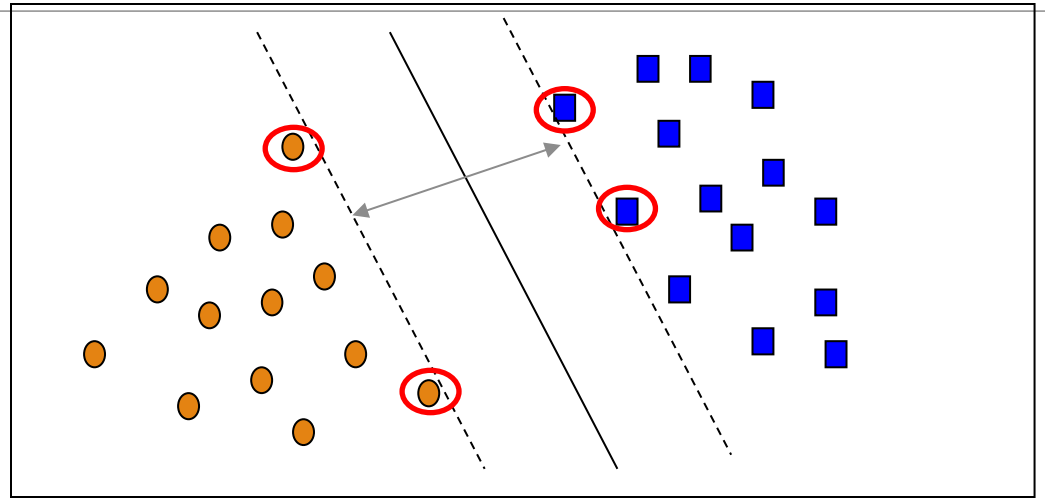


Classification: Basic Concepts

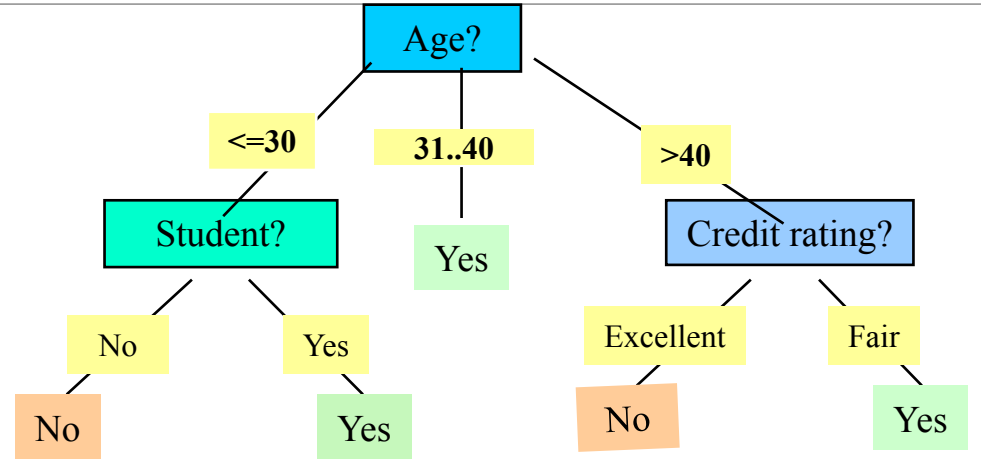
What Is Classification?



Typical Classification Methods

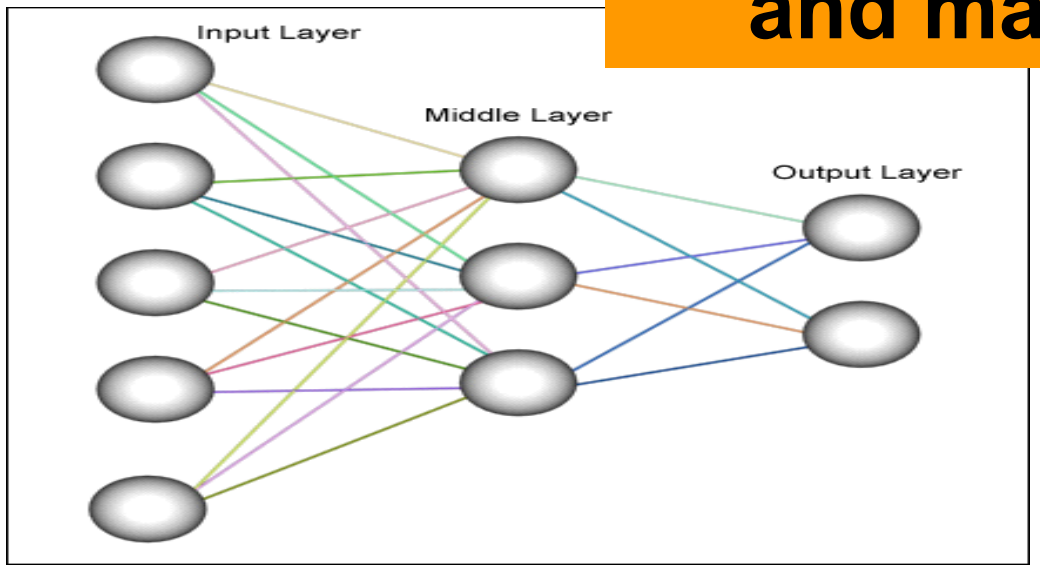


Support Vector Machi

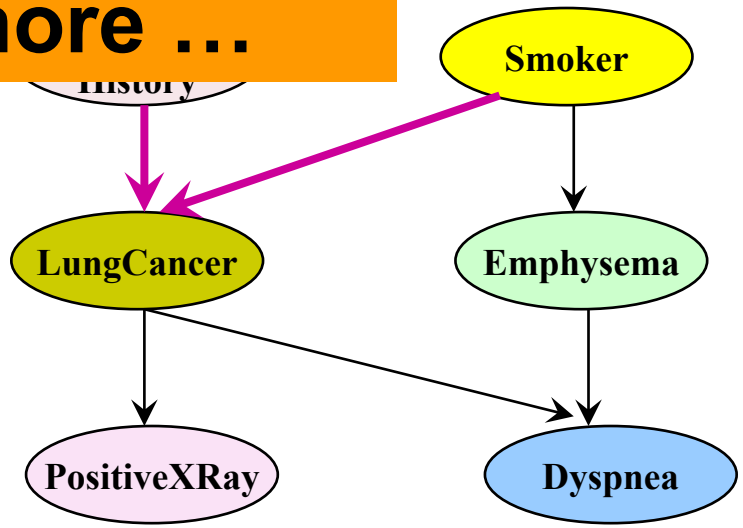


Decision Tree

and many more …

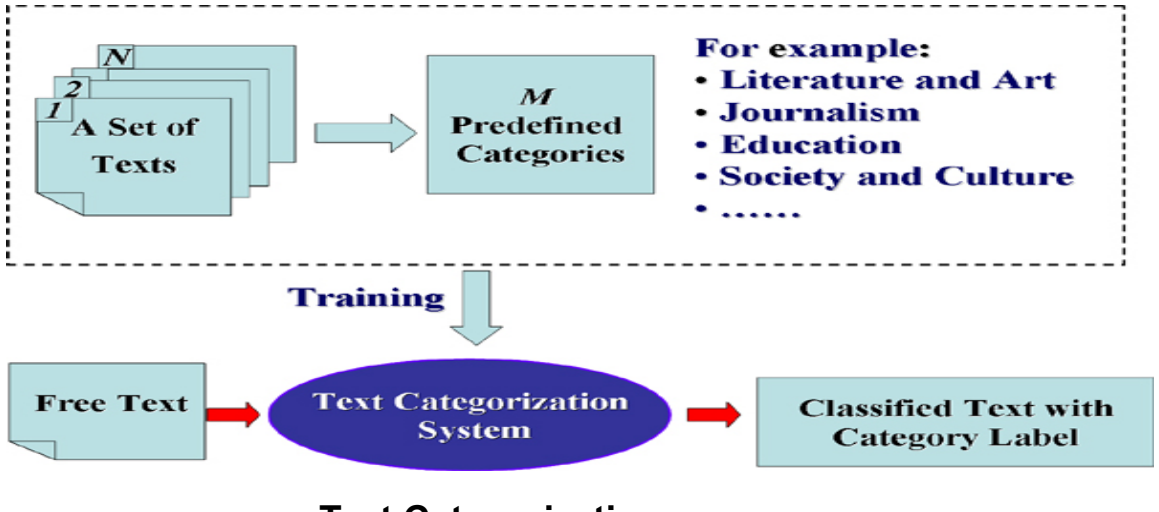


Neural Network

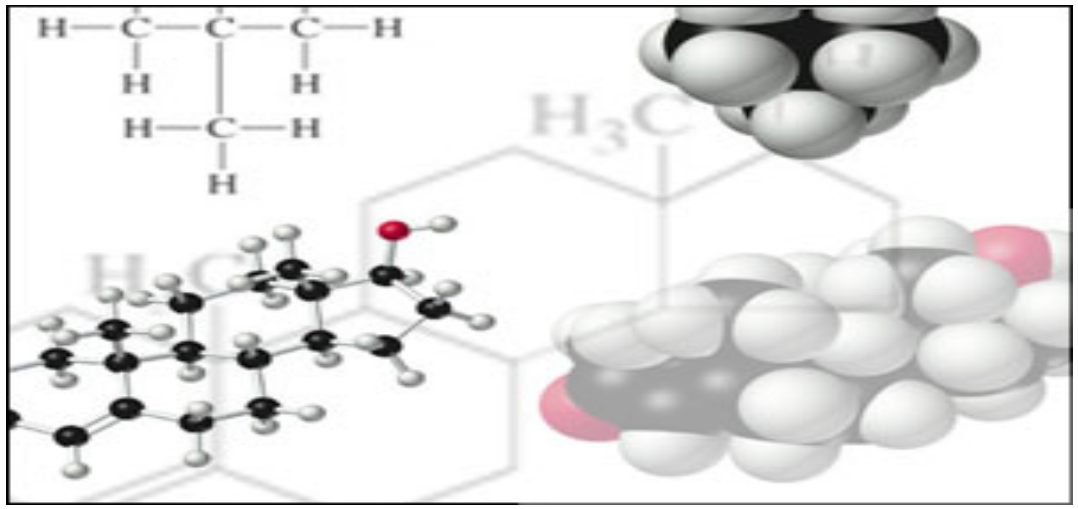


Bayesian Network

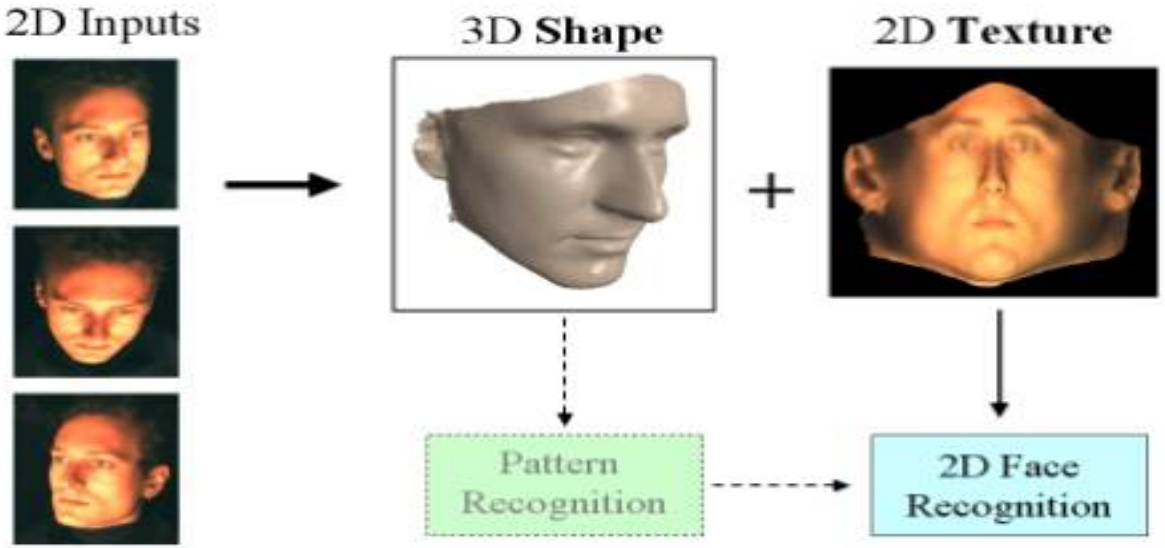
Numerous Classification Applications



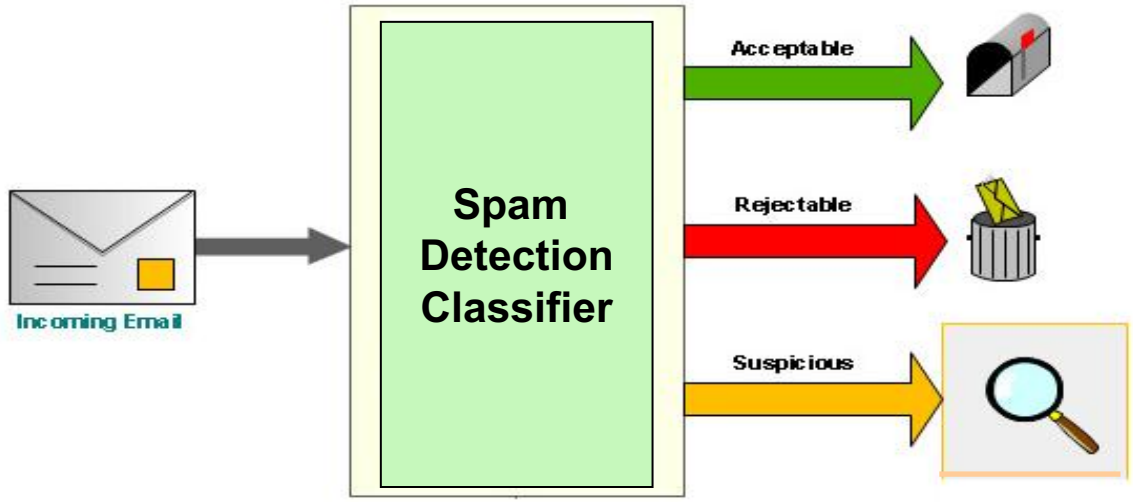
Text Categorization



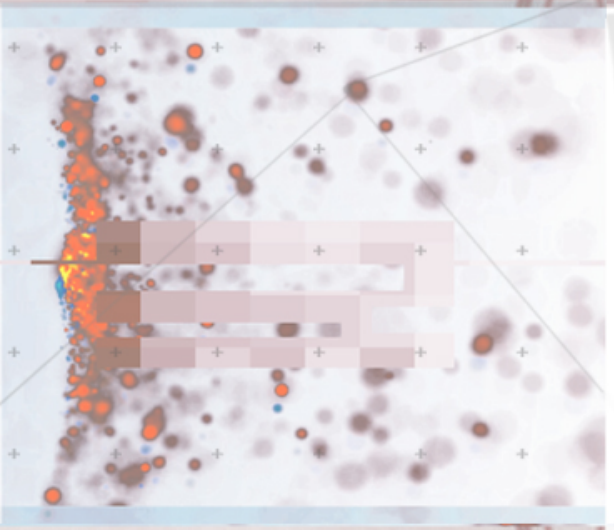
Drug Design



Face Recognition



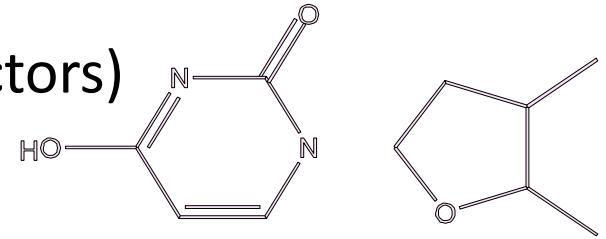
Spam Detection



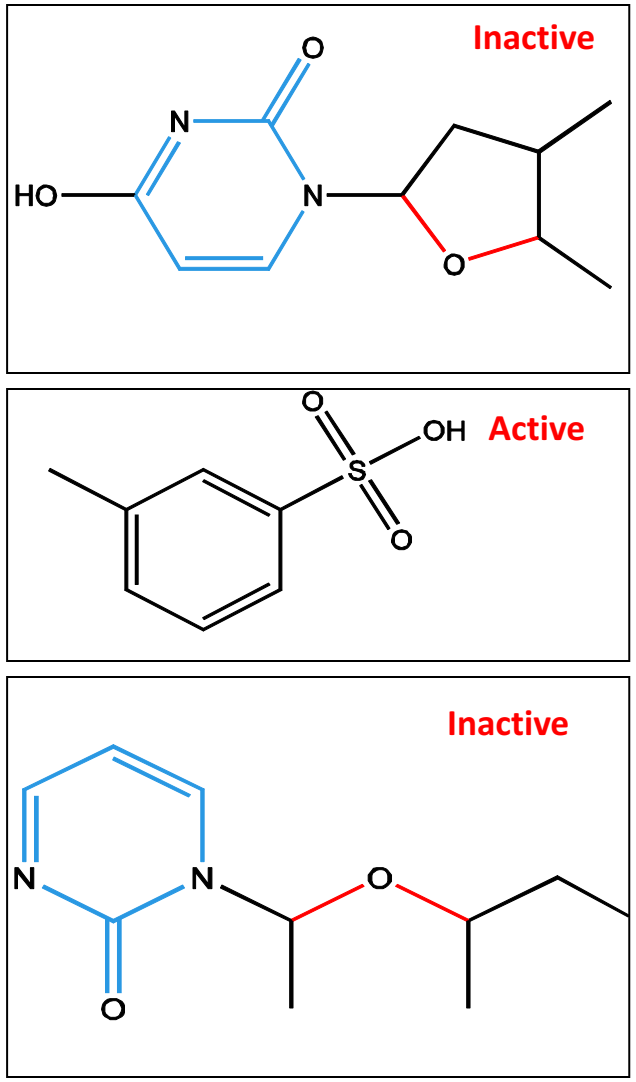
Pattern-Based Classification

Why Pattern-Based Classification?

- **Pattern-based classification:** An integration of both themes
- **Why pattern-based classification?**
  - **Feature construction**
    - Higher order; compact; discriminative
    - E.g., single word → phrase (apple pie, Apple iPad)
  - **Complex data modeling**
    - Graphs (no predefined feature vectors)
    - Sequences
    - Semi-structured/unstructured data



Pattern-Based Classification on Graphs



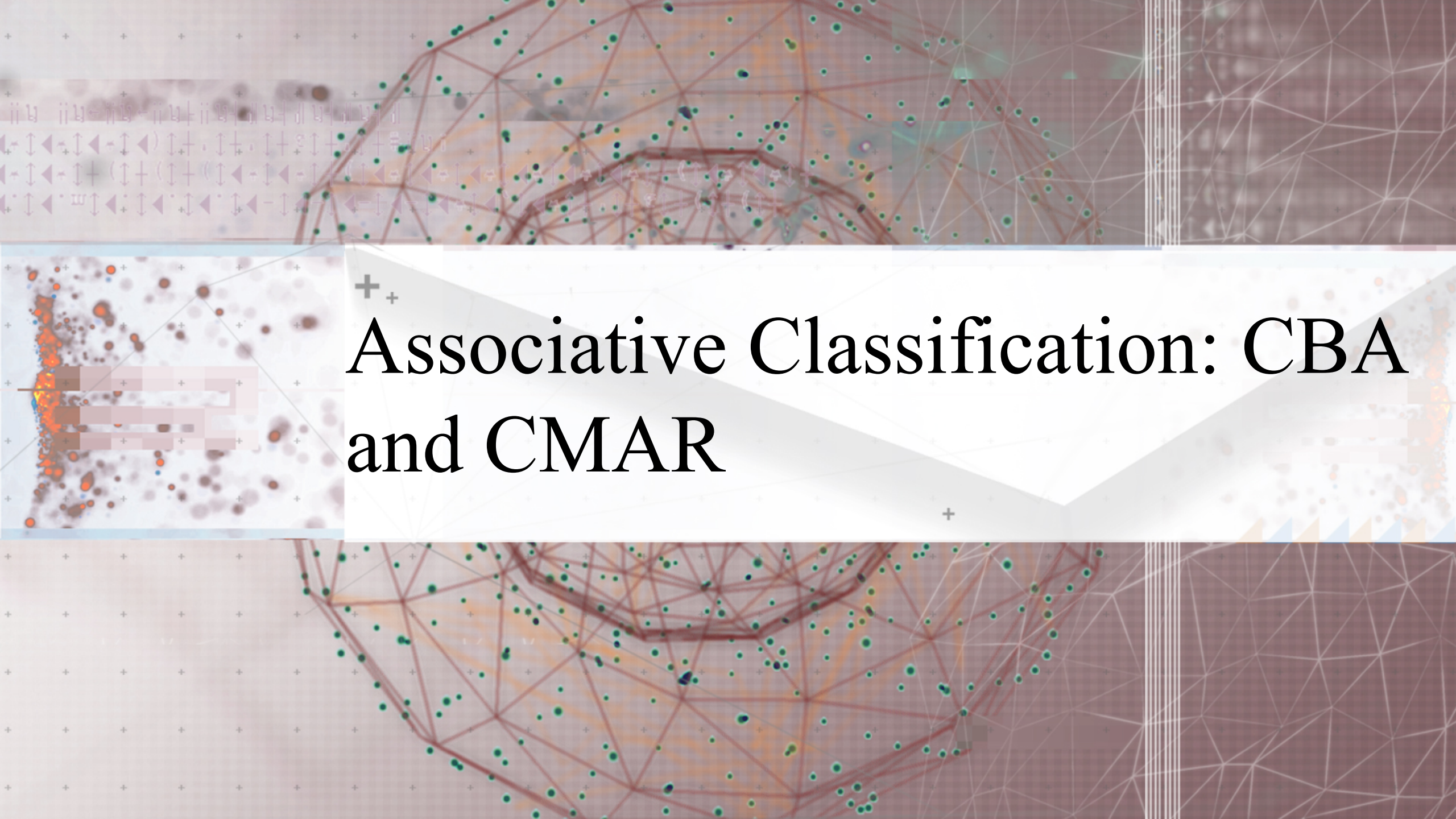
Use frequent patterns as features for classification

g_1	g_2	Class
1	1	0
0	0	1
1	1	0

Associative or Pattern-Based Classification

- **Data:** Transactions, microarray data, … → **Patterns or association rules**
- **Classification methods** (some interesting work):
  - CBA (Liu, Hsu, & Ma, KDD'98): Use high-confidence, high-support *class association rules* to build classifiers To be discussed here
  - Emerging patterns (Dong & Li, KDD'99): Patterns whose support changes significantly between the two classes
  - CMAR (Li, Han, & Pei, ICDM'01): Multiple rules in prediction To be discussed here
  - CPAR (Yin & Han, SDM'03): Beam search on multiple prediction rules
  - RCBT (Cong, Tan, Tung, & Xu, SIGMOD'05): Build classifier based on mining top-k covering rule groups with row enumeration (for high-dimensional data)
  - Lazy classifier (Veloso, Meira, & Zaki, ICDM'06): For a test t, project training data D on t, mine rules from D_t, predict on the best rule
  - Discriminative pattern-based classification (Cheng, Yan, Han, & Hsu, ICDE'07)

To be discussed here



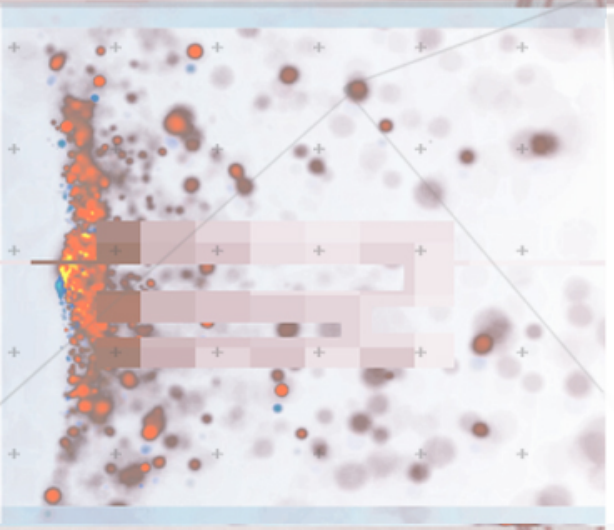
Associative Classification: CBA and CMAR

CBA: Classification Based on Associations

- CBA (Liu, Hsu, and Ma, KDD'98)
- Method
  - Mine high-confidence, high-support class association rules
  - LHS: Conjunctions of attribute-value pairs; RHS: Class labels
    p_1 ^ p_2 … ^ p_l → "$A_{class\text{-}label}$ = C" (confidence, support)
  - Rank rules in descending order of confidence and support
  - Classification: Apply the first rule that matches a test case; otherwise, apply the default rule
  - Effectiveness: Often found more accurate than some traditional classification methods, such as C4.5
  - Why? – Exploring high confident associations among multiple attributes may overcome some constraints introduced by some classifiers that consider only one attribute at a time

CMAR: Classification Based on Multiple Association Rules

- Rule pruning whenever a rule is inserted into the tree
  - Given two rules R_1 and R_2, if the antecedent of R_1 is more general than that of R_2 and $conf(R_1) \geq conf(R_2)$, then prune R_2
  - Prunes rules for which the rule antecedent and class label are not positively correlated based on the χ^2 test of statistical significance
- Classification based on generated/pruned rules
  - If only *one rule* satisfies tuple X, assign the class label of the rule
  - If a *rule set* S satisfies X
    - Divide S into groups according to class labels
    - Use a weighted χ^2 measure to find the strongest group of rules based on the statistical correlation of rules within a group
    - Assign X the class label of the strongest group
- CMAR improves model construction efficiency and classification accuracy



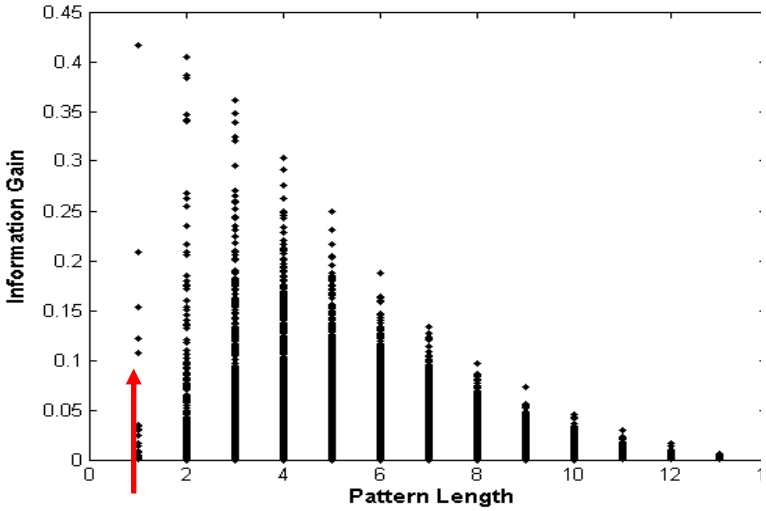
Discriminative Pattern-Based Classification

Discriminative Pattern-Based Classification

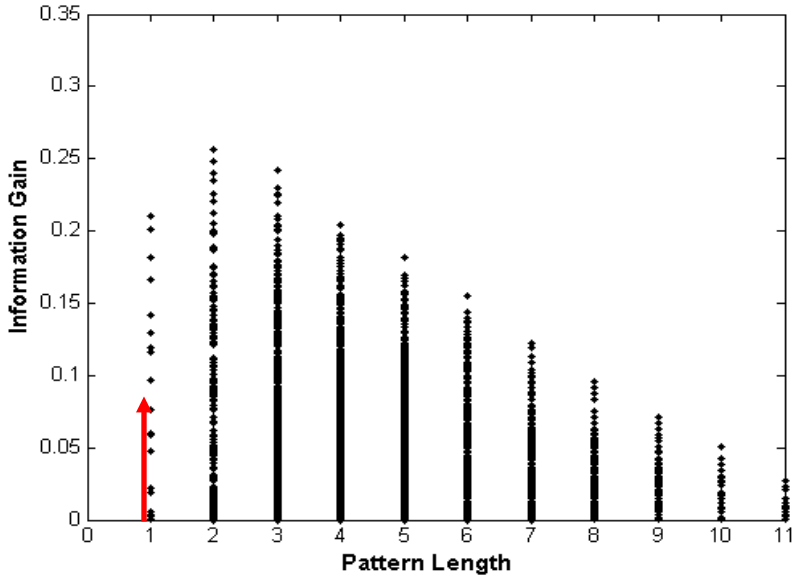
- Discriminative patterns as features for classification (Cheng et al., ICDE'07)
- **Principle:** Mining discriminative frequent patterns as high-quality features and then apply any classifier
- **Framework (PatClass)**
  - Feature construction by *frequent itemset mining*
  - Feature selection (e.g., using Maximal Marginal Relevance (MMR))
    - Select discriminative features (i.e., that are relevant but minimally similar to the previously selected ones)
    - Remove redundant or closely correlated features
  - Model learning
    - Apply a general classifier, such as SVM or C4.5, to build a classification model

On the Power of Discriminative Patterns

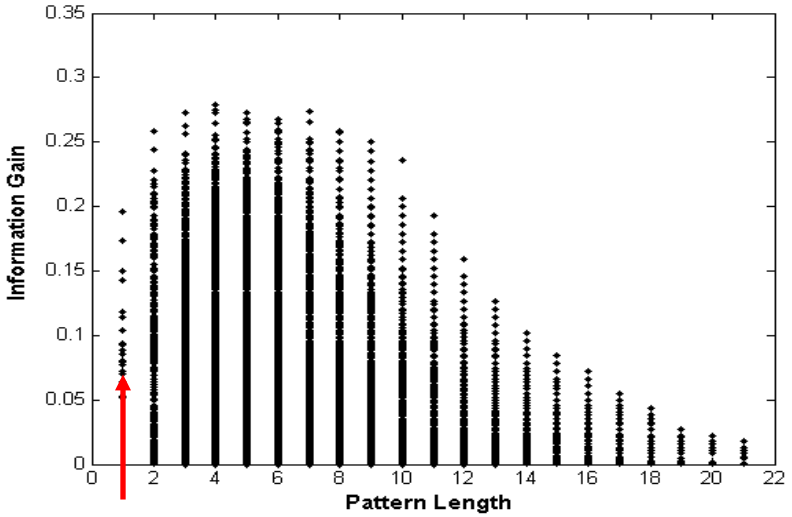
- K-itemsets are often more informative than single features (1-itemsets) in classification
- Computation on real datasets shows: The discriminative power of k-itemsets (for k > 1 but often ≤ 10) is higher than that of single features



(a) Austral



(b) Cleve

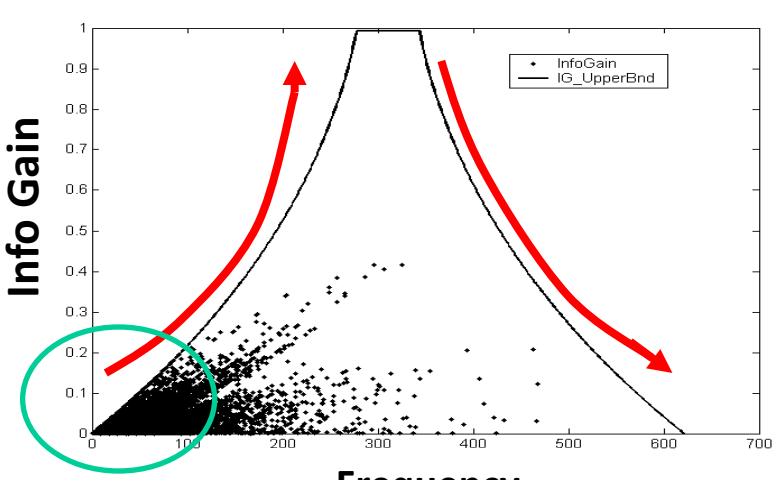


(c) Sonar

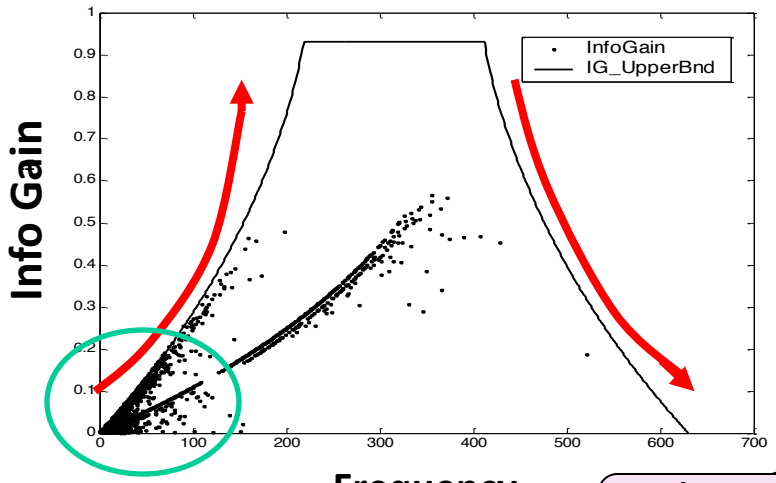
Information Gain vs. Pattern Length

Information Gain vs. Pattern Frequency

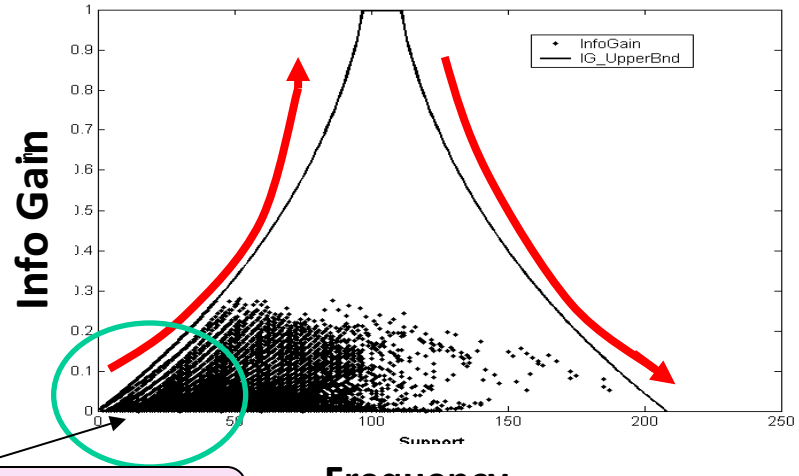
- Computation on real datasets shows: Pattern frequency (but not too frequent) is strongly tied with the discriminative power (information gain)
- Information gain upper bound monotonically increases with pattern frequency



(a) Austral



(b) Breast



(c) Sonar

Low support, low info gain

Information Gain Formula: $IG(C \mid X) = H(C) - H(C \mid X)$

Conditional entropy of study focus

Entropy of given data

$$H(C) = -\sum_{i=1}^{m} p_i \log_2(p_i)$$

$$H(C \mid X) = \sum_j P(X = x_j) H(Y \mid X = x_j)$$

Discriminative Pattern-Based Classification: Experimental Results

Table 1. Accuracy by SVM on Frequent Combined Features vs. Single Features

Data	Single Feature			Freq. Pattern	
	Item_All	*Item_FS*	*Item_RBF*	*Pat_All*	*Pat_FS*
anneal	**99.78**	**99.78**	99.11	99.33	99.67
austral	85.01	85.50	85.01	81.79	**91.14**
auto	83.25	84.21	78.80	74.97	**90.79**
breast	97.46	97.46	96.98	96.83	**97.78**
cleve	84.81	84.81	85.80	78.55	**95.04**
diabetes	74.41	74.41	74.55	77.73	**78.31**
glass	75.19	75.19	74.78	79.91	**81.32**
heart	84.81	84.81	84.07	82.22	**88.15**
hepatic	84.50	89.04	85.83	81.29	**96.83**
horse	83.70	84.79	82.36	82.35	**92.39**
iono	93.15	94.30	92.61	89.17	**95.44**
iris	94.00	**96.00**	94.00	95.33	**96.00**
labor	89.99	91.67	91.67	94.99	**95.00**
lymph	81.00	81.62	84.29	83.67	**96.67**
pima	74.56	74.56	76.15	76.43	**77.16**
sonar	82.71	86.55	82.71	84.60	**90.86**
vehicle	70.43	72.93	72.14	73.33	**76.34**
wine	98.33	99.44	98.33	98.30	**100**
zoo	97.09	97.09	95.09	94.18	**99.00**

Table 2. Accuracy by C4.5 on Frequent Combined Features vs. Single Features

Dataset	Single Features		Frequent Patterns	
	Item_All	*Item_FS*	*Pat_All*	*Pat_FS*
anneal	98.33	98.33	97.22	**98.44**
austral	84.53	84.53	84.21	**88.24**
auto	71.70	77.63	71.14	**78.77**
breast	95.56	95.56	95.40	**96.35**
cleve	80.87	80.87	80.84	**91.42**
diabetes	**77.02**	**77.02**	76.00	76.58
glass	75.24	75.24	76.62	**79.89**
heart	81.85	81.85	80.00	**86.30**
hepatic	78.79	85.21	80.71	**93.04**
horse	83.71	83.71	84.50	**87.77**
iono	92.30	92.30	92.89	**94.87**
iris	**94.00**	**94.00**	93.33	93.33
labor	86.67	86.67	**95.00**	91.67
lymph	76.95	77.62	74.90	**83.67**
pima	75.86	75.86	76.28	**76.72**
sonar	80.83	81.19	**83.67**	**83.67**
vehicle	70.70	71.49	**74.24**	73.06
wine	95.52	93.82	96.63	**99.44**
zoo	91.18	91.18	95.09	**97.09**

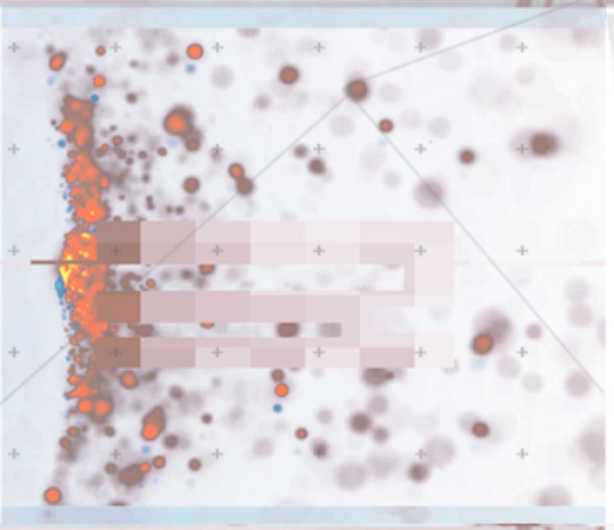
Discriminative Pattern-Based Classification: Scalability Tests

Table 3. Accuracy & Time on Chess Data

min_sup	#Patterns	Time (s)	SVM (%)	C4.5 (%)
1	N/A	N/A	N/A	N/A
2000	68,967	44.703	92.52	97.59
2200	28,358	19.938	91.68	97.84
2500	6,837	2.906	91.68	97.62
2800	1,031	0.469	91.84	97.37
3000	136	0.063	91.90	97.06

Table 4. Accuracy & Time on Waveform Data

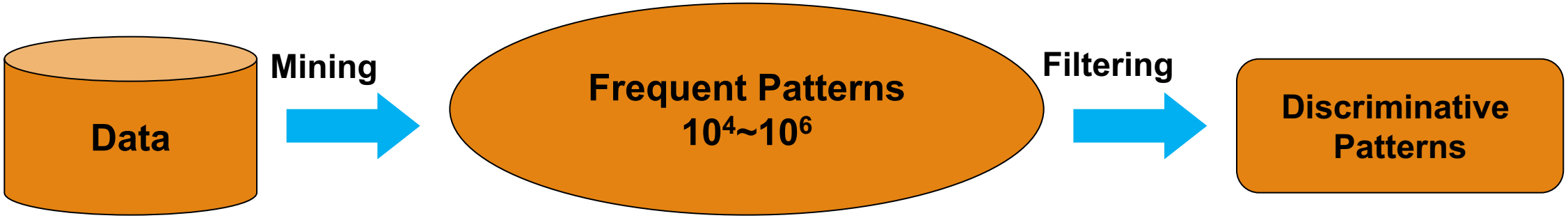
min_sup	#Patterns	Time (s)	SVM (%)	C4.5 (%)
1	9,468,109	N/A	N/A	N/A
80	26,576	176.485	92.40	88.35
100	15,316	90.406	92.19	87.29
150	5,408	23.610	91.53	88.80
200	2,481	8.234	91.22	87.32



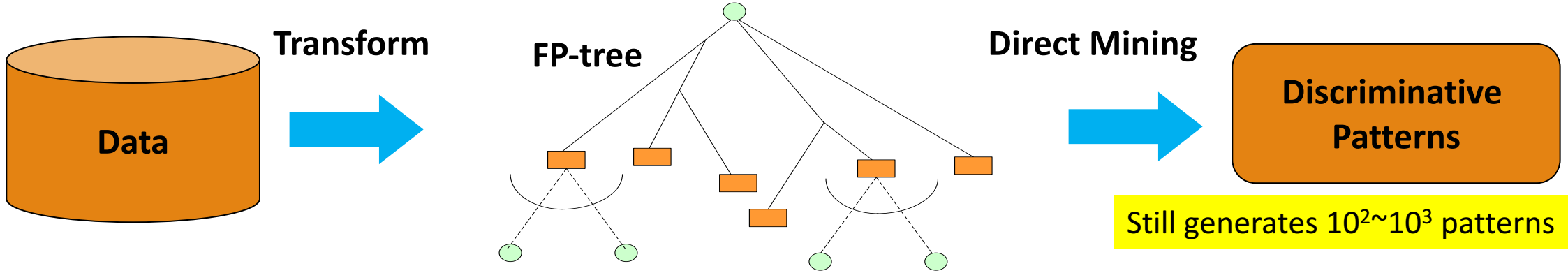
DPClass: Mining Concise Set of Discriminative Patterns for Classification

Mining Concise Set of Discriminative Patterns

Frequent pattern mining, then getting discriminative patterns: Expensive, large model

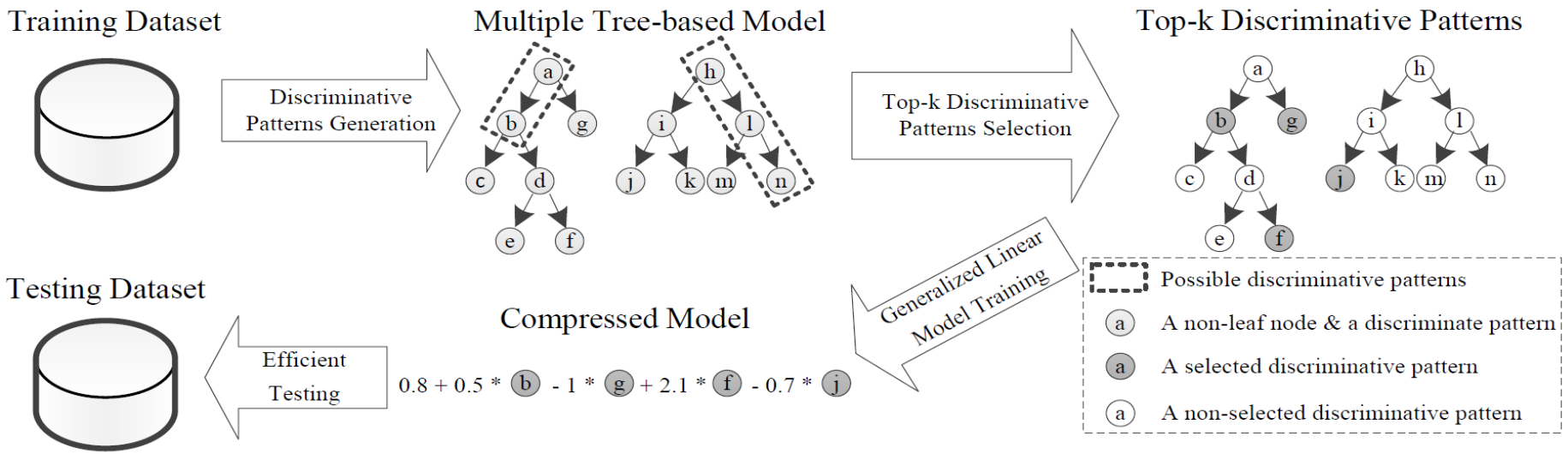


DDPMine (Cheng et al., ICDE'08): Direct mining of discriminative patterns: Efficient



DPClass (Shang et al., SDM'16): A better solution (see the next page)—efficient, effective & generating a very limited number of (such as only 20 or so) patterns

DPClass: Discriminative Pattern-Based Classification



Input: A feature table for training data

- Adopt every prefix path in an (extremely) random forest as a candidate pattern
  - The split points of continuous variables are automatically chosen by random forest → No discretization!
- Run top-k (e.g., top-20) pattern selection based on training data
- Train a generalized linear model (e.g., logistic regression) based on "bag-of-patterns" representations of training data

Explanatory Discriminative Patterns: Generation

- Example: For each patient, we have several uniformly sampled features as follows
  - Age (A): Positive integers no more than 60
  - Gender (G): Male or female
  - Lab Test 1 (LT1): Categorical values from (A, B, O, AB)
  - Lab Test 2 (LT2): Continuous values in [0..1]
- The positive label of the hypothetical disease will be given when at least one of the following rules holds
  - (age > 18) and (gender = Male) and (LT1 = AB) and (LT2 ≥ 0.6)
  - (age > 18) and (gender = Female) and (LT1 = O) and (LT2 ≥ 0.5)
  - (age ≤ 18) and (LT2 ≥ 0.9)
- Training: 10^5 random patients + add 0.1% noise
  - Flip the binary labels with 0.1% probability
- Testing: 5×10^4 random patients in test

Explanatory Discriminative Patterns: Evaluation

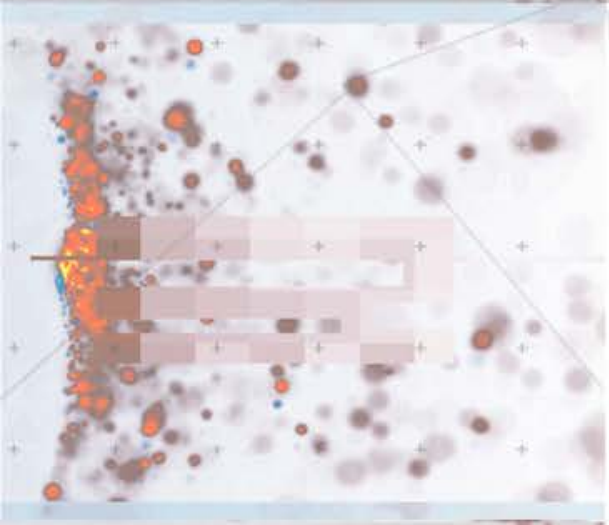
- Accuracy:
  - DPClass 99.99% (perfect)
  - DDPMine 95.64% (reasonable)
- Top-3 Discriminative Patterns:
  - DPClass generates a high quality model here:
    - (age > 18) and (gender = Female) and (LT1 = O) and (LT2 ≥ 0.496)
    - (age ≤ 18) and (LT2 ≥ 0.900)
    - (age > 18) and (gender = Male) and (LT1 = AB) and (LT2 ≥ 0.601)
  - DDPMine generates a relatively poor quality model here:
    - (LT2 > 0.8)
    - (gender = Male) and (LT1 = AB) and (LT2 ≥ 0.6) and (LT2 < 0.8)
    - (gender = Female) and (LT1 = O) and (LT2 ≥ 0.6) and (LT2 < 0.8)

A Comparison on Classification Accuracy

- DPClass: Discriminative & frequent at the same time, then select top-k
- Only top-20 patterns are used in DPClass
  - Two methods on pattern selection
    - Forward vs. LASSO
- In comparison with DDPMine and Random Forest, DPClass maintains high accuracy

	Dataset	DPClass (Forward)	DPClass (LASSO)	DDPMine	Random Forest
Low-Dimensional Data	adult	**85.66%**	84.33%	83.42%	85.45%
	hypo	**99.58%**	99.28%	92.69%	97.22%
	sick	98.35%	**98.87%**	93.82%	94.03%
	crx	**89.35%**	87.96%	87.96%	**89.35%**
	sonar	**85.29%**	83.82%	73.53%	83.82%
	chess	92.25%	92.05%	90.04%	**94.22%**
High-Dimensional Data	namao	97.17%	96.94%	96.83%	**97.86%**
	musk	95.92%	95.71%	93.29%	**96.60%**
	madelon	74.50%	**76.00%**	59.84%	56.50%

- An extension of DPClass has been applied to health study
  - Cheng, Q. et al. (2016). Mining discriminative patterns to predict health status for cardiopulmonary patients. *ACM-BCB'16*



K Nearest Neighbors Algorithm

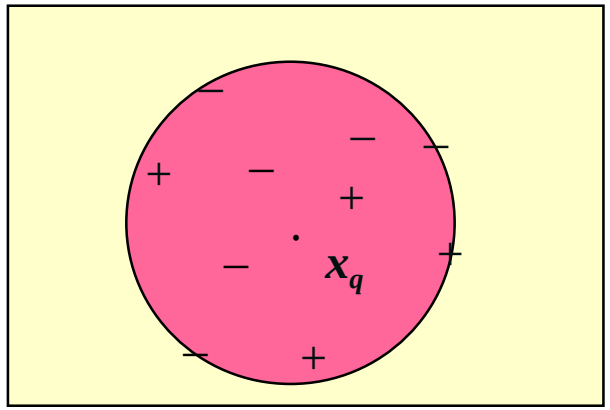
Lazy vs. Eager Learning

- Lazy vs. eager learning
  - **Eager learning** (previously discussed methods): Given a set of training objects, constructs a classification model before receiving new (e.g., test) data to classify
  - **Lazy learning** (e.g., instance-based learning): Simply stores training data (or only minor processing) and waits until it is given a test tuple
- Lazy: Less time in training but more work in predicting
- Accuracy
  - Eager learning: Must commit to a single hypothesis that covers the entire instance space
  - Lazy method may effectively use a richer hypothesis space since it uses many *local linear functions* to form an implicit global approximation to the target function

Lazy Learner: Instance-Based Methods

- Instance-based learning:
  - Training phase: Only storing the feature vectors and class labels of the training samples
  - Delay the processing ("lazy evaluation") until an instance (queried example) must be classified
- Typical approaches
  - *k*-nearest neighbor approach
    - Instances represented as points in a Euclidean space
  - Locally weighted regression
    - Constructs local approximation
  - Case-based reasoning
    - Uses symbolic representations and knowledge-based inference

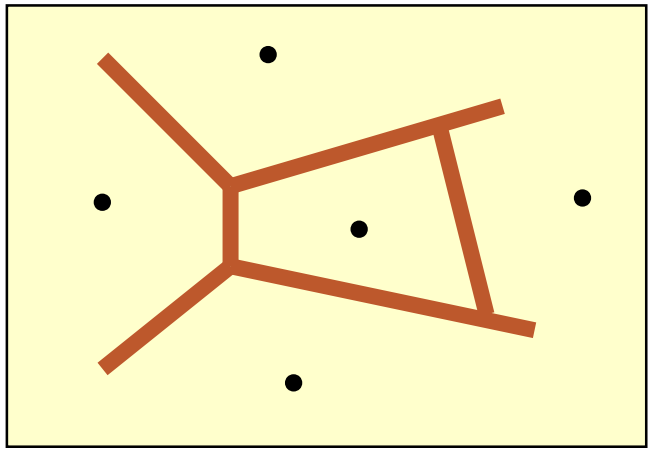
The *k*-Nearest Neighbors Algorithm



- Distance metric for k-NN: How to define the nearest neighbor?
  - All instances correspond to points in the *n*-D space
  - For continuous variables, commonly used metric: Euclidean distance
  - For text classification: Overlap metric (or Hamming distance)
  - For gene expression microarray data:
    - Correlation coefficients such as Pearson
- KNN can be used for classification or regression
  - *k* is a user-defined constant
  - *k-NN classification*: An object is assigned to the class most common among the *k* nearest neighbors
  - *k-NN regression*: Returns the average of the values of its *k* nearest neighbors

Distance-Weighted *k*-NN and Other Discussions

- Distance-weighted k-nearest neighbor algorithm
  - Give greater weight to closer neighbors
    - A common weighting scheme:
      - Giving each neighbor a weight of $1/d$ (d is the distance to the query x_q)
      - or $w \equiv \frac{1}{d(x_q, x_i)^2}$
- For high-dimensional data, first employ dimensionality reduction or feature selection, such as PCA (principal component analysis) or LDA (linear discriminant analysis)
- Efficient pre-computation or indexing
  - Vonoroi diagram: The decision surface induced by 1-NN for a typical set of training examples



Recommended Readings

- Altman, N. S. (1992). An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician, 46*(3), 175–185. Retrieved from https://www.jstor.org/stable/2685209?seq=1#page_scan_tab_contents
- Cover, T. M. & Hart, P. E. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory 13*(1), 21–27. Retrieved from http://ieeexplore.ieee.org/document/1053964/
- Zhang, H., Berg, A. C., Maire, M., & Malik, J. (2006). SVM-KNN: Discriminative nearest neighbor classification for visual category recognition. *CVPR, 2*, 2126-2136. Retrieved from http://ieeexplore.ieee.org/document/1641014/
- Zhang, M. L. & Zhou, Z. H. (2007). ML-KNN: A lazy learning approach to multi-label learning. *Pattern Recognition, 40*(7), 2038-2048. Retrieved from https://www.sciencedirect.com/science/article/pii/S0031320307000027

References

- All other multimedia elements belong to © 2018 University of Illinois Board of Trustees.