Data Mining

- **How to estimate a classifier performance** (e.g. accuracy)**?**
  - Holdout method
  - Cross-validation
  - Bootstrap method

Sections 4.5 and 4.6 of course book.

- **How to compare two classifiers performance?**
- **Introduction to association rule mining**
  - Frequent itemsets
  - Association rules
  - Support and confidence

    Limitations of support and confidence

Sections 6.1-2, 6.7 of course book

Evaluation of a Classifier

- Error on the training data is *not* a good indicator of performance on future data
  - *Resubstitution error:* error rate obtained from training data
  - Resubstitution error is (hopelessly) optimistic!
- Simple solution that can be used, if lots of (labeled) data is available:
  - Split data into **training** and **test set**
  - **Assumption**: both training data and test data are representative samples of the underlying problem
  - ***Ex***: *build a classifier using customer data from a town A and test it on data from a town B*

Making the Most of the Data

- **Ideal situation**: a large representative sample to train the model and another independent large representative sample to test the model
  - Generally, the larger the training data the better the classifier (but returns diminish)
  - The larger the test data the more accurate performance measures estimate (e.g. accuracy)
- **Problem:** What to do if the amount of (labeled) data is limited?

Holdout Method

- The ***holdout* method** reserves a certain amount for testing and uses the remainder for training
  - Usually: one third for testing, the rest for training
- **Problem**: the samples might not be representative
  - Example: a class might be missing in the test data
- Advanced version uses *stratification*
  - Ensures that each class is represented with approximately equal proportions in both subsets
  - **It doesn't prevent bias in the training and test sets**

Estimate Error Rate with Holdout Method

- The holdout method provides only **an estimate** of the true error rate (accuracy) of a classifier
  - The true error rate is *unkown*
- Why is the true error rate so difficult to obtain?
  - We only have a (small) data sample of the whole data (population)
  - Sampling is used to divide the data in test set and training set
    - Bias might be introduced
  - Less data available for building the model
    - Classifier may not be as good as if the whole available data is used

Repeated Holdout Method

- Holdout estimate can be made more reliable by repeating the process with different subsamples
  - In each iteration, a certain proportion is randomly selected for training (possibly with stratification)
  - The error rates (or some other performance measure) on the different iterations are averaged to yield an overall error rate
- This is called the ***repeated holdout* method**
- Still not optimum: the different test sets may overlap
  - Some examples may never appear in training sets
  - Can we prevent overlapping?

Cross-validation

- ***Cross-validation*** avoids overlapping test sets
  1. Split data into $k > 0$ disjoint subsets of equal size
  2. Use each subset in turn for testing, the remainder for training

K-fold cross-validation

- Often the subsets are stratified before the cross-validation is performed
- The error estimates are averaged to yield an overall error estimate

More on Cross-validation

- Standard method for evaluation: *stratified ten-fold cross-validation*
  - Standard error estimation technique when limited data is available
- Why ten?
  - Extensive experiments have shown that this is the best choice to get an accurate estimate
  - There is also some theoretical evidence for this
- Even better: repeated stratified cross-validation
  - E.g. ten-fold cross-validation is repeated ten times and results are averaged (reduces the variance)

Leave-One-Out

- **Leave-One-Out**: a particular form of cross-validation (n-fold CV)
  - Set number of folds to number of training instances
    - i.e., for *n* training instances, build classifier *n* times
  - Mainly used for rather small datasets
- Positive aspects:
  - Makes best use of the data for training
    - Increases the chance of building more accurate classifiers
  - Involves no random subsampling
- Drawbacks:
  - Very computationally expensive
  - Stratification is not possible

The Bootstrap

- ***CV*** uses sampling *without replacement*
  - The same instance, once selected, can not be selected again for a sample
  - There are no duplicate records in the training and test sets
- The *bootstrap uses sampling with replacement* to form the training set
  - Sample a dataset of *n* instances *n* times *with replacement* to form a new dataset of *n* instances
  - Use this data as the training set
    - An instance may occur more than once
  - Use the instances from the original dataset that don't occur in the new training set for testing

The Bootstrap

- The error estimate on the test data will be very pessimistic
  - Trained on just ~63% of the instances
- Therefore, combine it with the resubstitution error:

$$err = 0.632 \cdot e_{\text{test instances}} + 0.368 \cdot e_{\text{training instances}}$$

- The resubstitution error gets less weight than the error on the test data
- Repeat process several times with different replacement samples and average the results
- Probably the best way of estimating performance for very small data sets

How to Compare Classifiers?

- **Discussed so far**: how to obtain reliable estimation of a performance measure (e.g. error rate) of a classifier
  - Open questions:
    - But, how much **confidence** can we place on the estimated error rate?
    - Is the difference in the error rate of two classifiers **statistically significant**?
  - **Ex**: Consider two classifiers A and B

$\texttt{error\_rate}_A$ = 25% $\texttt{error\_rate}_B$ = 24%

Question: Which one is the best classifier?

- How much confidence can we place in the accuracy of the classifiers?
- Is it possible that the differences in the performances are a result of variations in the composition of training/test sets?

Confidence Interval Estimation

- Assume the estimated error rate is $p_e = 25\%$
  - How close is this to the true error rate p on the target population?
    - Within 5%? 10%?
    - **Answer**: e.g. with 95% confidence $p \in [0.25-\delta, 0.25+\delta]$
    - i.e., in 1 out of 20 cases $p \notin [p_e-\delta, p_e+\delta]$, when considering new data sets
    - Higher confidence levels with small δ are preferred
  - It depends on the amount of test data
- **Interval estimation** is a statistical technique
  - 100% confidence is only possible with the data of the whole population
  - There is a remarkable reduction in the amount of data needed, if only a 99% confidence is selected

Comparing Data Mining Techniques

- **Frequent question**: which of two learning schemes performs better?
  - Obvious way: compare 10-fold ***CV*** estimates
  - Problem: variance in estimate
    - Any particular cross-validation experiment yields only an approximation of the true error rate
- **Question**: do the two means of the 10 ***CV*** estimates differ significantly?
  - **Statistical technique**: significance tests tell us how confident we can be that there really is a difference
    - *Student's t-test* tells whether the means of two (small) samples are significantly different

PART II

- Association Rule Mining

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

Market-Basket transactions

TID	*Items*
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

Example of Association Rules

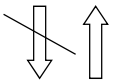
{Diaper} → {Beer},
{Milk, Bread} → {Eggs,Coke},
{Beer, Bread} → {Milk},

Implication means co-occurrence, not causality!

Interpreting Association Rules

1. {Milk, Bread} → Eggs,
2. {Milk, Bread} → Coke

Is not the same as



3. {Milk, Bread} → {Eggs, Coke}

However, 3. means that the following holds:

Is not the same

4. {Milk, Bread, Eggs} → Coke
5. {Milk, Bread, Coke} → Eggs

Frequent Itemset

- **Itemset**
  - A collection of one or more items
    - Ex: {Milk, Bread, Diaper}
  - k-itemset
    - An itemset that contains **k** items
- **Support count (σ)**
  - Frequency of occurrence of an itemset
  - Ex: σ({Milk, Bread,Diaper}) = 2
- **Support (s)**
  - Fraction of transactions that contain an itemset
  - Ex: s({Milk, Bread, Diaper}) = 2/5 = 40%
- **Frequent Itemset**
  - An itemset whose support is greater than or equal to a *minsup* threshold

TID	*Items*
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

Association Rule

- **Association Rule**
  - An expression of the form X → Y, where X and Y are itemsets
  - Example: {Milk, Diaper} → {Beer}

TID	Items
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

- **Rule Evaluation Metrics**
  - **Support (s)**
    - Fraction of transactions that contain both X and Y
  - **Confidence (c)**
    - Measures how often items in Y appear in transactions that contain X
    - Like a conditional probability P(Y|X)

Example:

$\{Milk, Diaper\} \Rightarrow Beer$

$$s = \frac{\sigma(\text{Milk}, \text{Diaper}, \text{Beer})}{|\text{DB}|} = \frac{2}{5} = 0.4$$

$$c = \frac{\sigma(\text{Milk}, \text{Diaper}, \text{Beer})}{\sigma(\text{Milk}, \text{Diaper})} = \frac{2}{3} = 0.67$$

Association Rule Mining Task

- Given a set of transactions DB, the goal of association rule mining is to find all rules having
  - support ≥ *minsup* threshold
  - confidence ≥ *minconf* threshold

- Brute-force approach:
  - List all possible association rules
  - Compute the support and confidence for each rule
  - Prune rules that fail the *minsup* and *minconf* thresholds

  ⇒ Computationally prohibitive!

Mining Association Rules

TID	Items
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

Example of Rules:

{Milk,Diaper} → {Beer} (s=0.4, c=0.67)
{Milk,Beer} → {Diaper} (s=0.4, c=1.0)
{Diaper,Beer} → {Milk} (s=0.4, c=0.67)
{Beer} → {Milk,Diaper} (s=0.4, c=0.67)
{Diaper} → {Milk,Beer} (s=0.4, c=0.5)
{Milk} → {Diaper,Beer} (s=0.4, c=0.5)

Observations:

- All the above rules are binary partitions of the same itemset:
  {Milk, Diaper, Beer}
- Rules originating from the same itemset have identical support but can have different confidence
- Thus, we may decouple the support and confidence requirements

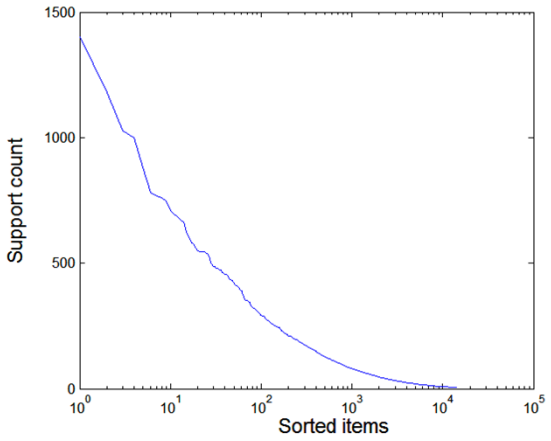
Mining Association Rules

- Two-step approach:
  1. Frequent Itemset Generation
     - Generate all itemsets whose support ≥ *minsup*
  2. Rule Generation
     - Generate high confidence rules from each frequent itemset, where each rule is a binary partitioning of a frequent itemset
     - Support counts for itemsets has been computed in step 1
- Frequent itemset generation is computationally expensive
  - How to compute them efficiently?

Effect of Support Distribution

- Many real data sets have skewed support distribution

Support distribution of a retail data set



Effect of Support Distribution

- How to set the appropriate *minsup* threshold?
  - If *minsup* is set too high, we could miss itemsets involving interesting rare items (e.g., expensive products)
  - If *minsup* is set too low then
    - It is computationally expensive and the number of itemsets is very large
    - Too many patterns might be extracted
    - Spurious patterns might be found (good confidence)

    E.g. Assume caviar is a very low frequency item and milk is a very high frequency item then the confidence of Caviar → Milk can be quite high

Drawback of Confidence

	Coffee	$\overline{\text{Coffee}}$	
Tea	15	5	20
$\overline{\text{Tea}}$	75	5	80
	90	10	100

← **Contigency Table**

Association Rule: Tea → Coffee

Confidence= P(Coffee|Tea) = 0.75

but P(Coffee) = 0.9

⇒ Although confidence is high, rule is misleading

⇒ P(Coffee|$\overline{\text{Tea}}$) = 0.9375

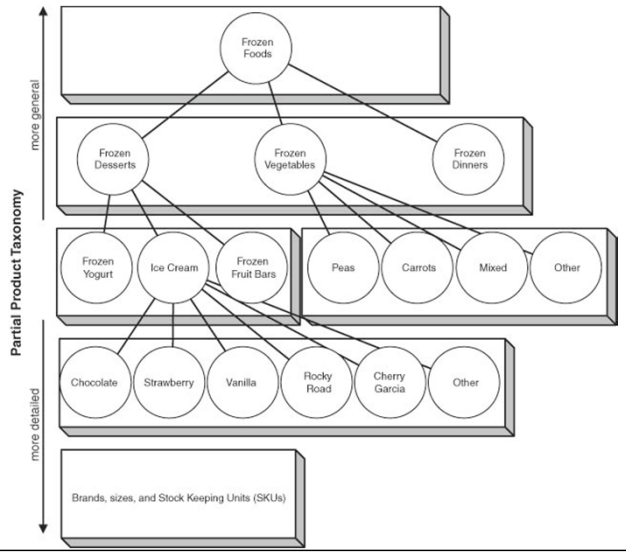
Lift of an Association Rule

- Drawback of confidence: support of the RHS of a rule is ignored
  - High confidence might however hide a negative correlation
- Lift of a rule A → B

$$\text{Lift} = \frac{c(A \rightarrow B)}{s(B)}$$

  - If **Lift > 1** then there is a positive correlation between A and B
  - If **Lift = 1** then A and B are independent
  - If **Lift < 1** then there is a negative correlation between A and B

Association Rules and Hierarchies



Association Rules and Hierarchies

- Using items at higher levels of the hierarchy
  - Less items, less itemsets
  - Less time and memory needed to obtain rules
  - Help in finding rules with sufficient support (rare items)
  - Easier to find interdepartmental relationships
- Using items at lower levels of the hierarchy
  - More interesting rules can be obtained
    - E.g. knowing what sells with a particular brand of vanilla ice cream can help in managing the relationship with the manufacturer
- Not all items need to be generalized to the same level of the hierarchy

Association Rules and Hierarchies

- Initially, use more general items (near to the top of the hiearchy)
  - E.g. *what can be done to boost the sales of frozen desserts?*
    - *{frozen vegetables} --> {frozen desserts}*
- *Repeat the rule generation process in more specific items*
  - *E.g. use only the subset of transactions that contains frozen desserts and frozen desserts*
    - *Roll down frozen vegetables (and frozen desserts)*