Data Mining
Feature Selection

Data & Feature Reduction

- **Data reduction**: Obtain a reduced representation of the data set that is much smaller in volume but yet produces the same (or almost the same) analytical results
- Why data reduction? — A database/data warehouse may store terabytes of data. Complex data analysis may take a very long time to run on the complete data set.

- Data reduction strategies
  - Dimensionality reduction, e.g., remove unimportant attributes
    - Filter Feature Selection
    - Wrapper Feature Selection
    - Feature Creation
  - Numerosity reduction (Data Reduction)
    - Clustering, sampling

  - Data compression

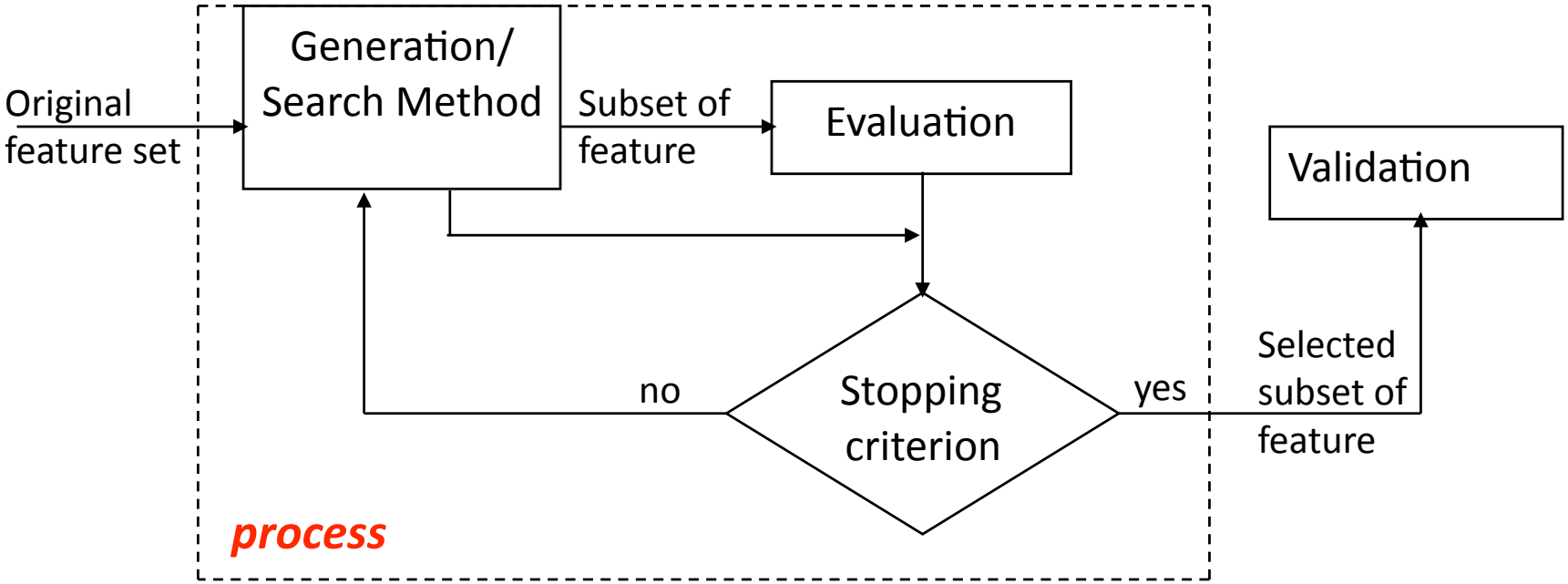
Feature Selection or Dimensionality Reduction

- **Curse of dimensionality**
  - When dimensionality increases, data becomes increasingly sparse
  - Density and distance between points, which is critical to clustering, outlier analysis, becomes less meaningful
  - The possible combinations of subspaces will grow exponentially

- **Dimensionality reduction**
  - Avoid the curse of dimensionality
  - Help eliminate irrelevant features and reduce noise
  - Reduce time and space required in data mining
  - Allow easier visualization

Feature Selection for Classification: General Schema

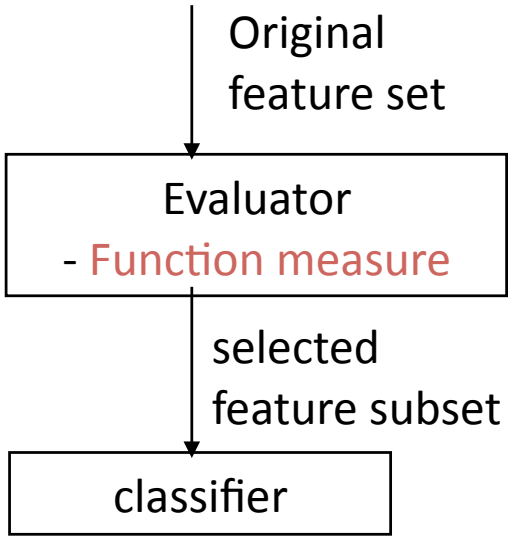
(6) Four main steps in a feature selection method.



Generation = select feature subset candidate.
Evaluation = compute relevancy value of the subset.
Stopping criterion = determine whether subset is relevant.
Validation = verify subset validity.

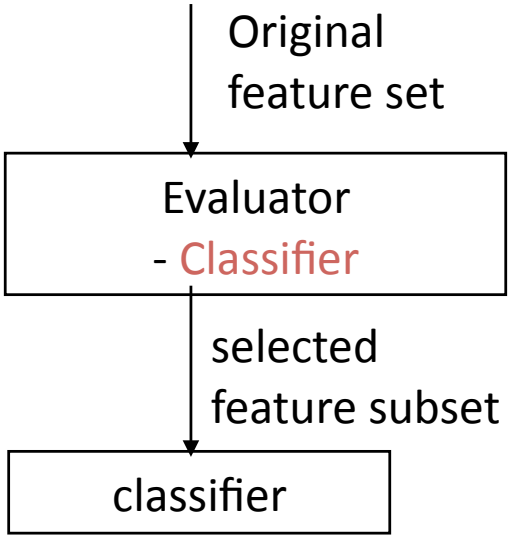
General Approach for Supervised Feature Selection

Filter approach



- evaluation fn <> classifier
- ignored effect of selected subset on the performance of classifier.

Wrapper approach



- evaluation fn = classifier
- take classifier into account.
- loss generality.
- high degree of accuracy.

Filter and Wrapper apprach: Search Method

Generation/Search Method

- select candidate subset of feature for evaluation.
- Start = no feature, all feature, random feature subset.
- Subsequent = add, remove, add/remove.
- categorise feature selection = ways to generate feature subset candidate.
- 5 ways in how the feature space is examined.

  Complete

  Heuristic

  Random

  Rank

  Genetic

Filter and Wrapper apprach: Search Method

Complete/exhaustive

- examine all combinations of feature subset.
  {f1,f2,f3} => { {f1},{f2},{f3},{f1,f2},{f1,f3},{f2,f3},{f1,f2,f3} }
- order of the search space $O(2^p)$, p - # feature.
- optimal subset is achievable.
- too expensive if feature space is large.

Heuristic

- selection is directed under certain guideline
  - selected feature taken out, no combination of feature.
  - candidate = { {f1,f2,f3}, {f2,f3}, {f3} }
- incremental generation of subsets.
- Forward selection or Backward Elimination
- search space is smaller and faster in producing result.
- miss out features of high order relations (parity problem).
  - Some relevant feature subset may be omitted {f1,f2}.

Filter and Wrapper apprach: Search Method

Random

- no predefined way to select feature candidate.
- pick feature at random (ie. probabilistic approach).
- optimal subset depend on the number of try
  - which then rely on the available resource.
- require more user-defined input parameters.
  - result optimality will depend on how these parameters are defined.
  - eg. number of try

Rank (specific for Filter)

- Rank the feature w.r.t. the class using a measure
- Set a threshold to cut the rank
- Select as features, all those features in the upper part of the rank

Filter and Wrapper apprach: Search Method

Genetic

- Use genetic algorithm to navigate the search space
- Genetic algorithm are based on the evolutionary principle
- Inspired by the Darwinian theory (cross-over, mutation)

Filter approach: Evaluator

Evaluator

- determine the relevancy of the generated feature subset candidate towards the classification task.

  Rvalue = J(candidate subset)

  if (Rvalue > best_value) best_value = Rvalue

- 4 main type of evaluation functions.

  distance (euclidean distance measure).

  information (entropy, information gain, etc.)

  dependency (correlation coefficient).

  consistency (min-features bias).

Filter Approach: Evaluator

Distance measure

- $z^2 = x^2 + y^2$
- select those features that
  support instances of the same class to
  stay within the same proximity.
- instances of same class should be closer in terms of distance
  than those from different class.

Filter Approach: Evaluator

Information measure

- Entropy of variable *X* $$H(X) = -\sum_i P(x_i)\log_2(P(x_i))$$
- Entropy of *X* after observing *Y* $$H(X|Y) = -\sum_j P(y_j)\sum_i P(x_i|y_j)\log_2(P(x_i|y_j))$$
- Information Gain $$IG(X|Y) = H(X) - H(X|Y)$$
- Symmetrical Uncertainty $$SU(X,Y) = 2\left[\frac{IG(X|Y)}{H(X)+H(Y)}\right]$$

For instance select an attribute A if IG(A) > IG(B).

Filter Approach: Evaluator

Dependency measure

- correlation between a feature and a class label.
- how close is the feature related to the outcome of the class label?
- dependence between features = degree of redundancy.
  - if a feature is heavily dependence on another, than it is redundant.
- to determine correlation, we need some physical value.
  value = distance, information

Filter Approach: Evaluator

Consistency measure

- two instances are *inconsistent*
  if they have *matching feature values*
  but group under *different class label*.

	f_1	f_2	class
instance 1	a	b	c1
instance 2	a	b	c2

- select {f1,f2}
  if in the training data set there exist no instances as above.
- heavily rely on the training data set.
- min-feature = want smallest subset with consistency.
- problem = 1 feature alone guarantee no inconsistency (eg. IC #).

Example of Filter method: FCBF

Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution, Lei Yu and Huan Liu, (ICML-2003)

- Filter approach for feature selection
- Fast method that use a correlation measure from information theory
- Based on the Relevance and Redundancy criteria
- Use a rank method without any threshold setting
- Implemented in Weka (**SearchMethod** : FCBFSearch
  Evaluator: SymmetricalUncertAttributeSetEval)

Fast Correlation-Based Filter (FCBF) Algorithm

- How to decide whether a feature f_i is relevant to the class C or not
  - Find a subset S', such that

$$\forall f_i \in S', 1 \le i \le N, SU_{i,c}, \quad \delta$$

- How to decide whether such a relevant feature is redundant
  - Use the correlation of features and class as a reference

Definitions

- Relevance Step
  - Rank all the features w.r.t. their correlation with the class
- Redundancy Step
  - Start to scan the feature rank from f_i, if a f_j(with $f_{jc} < f_{ic}$) has a correlation with f_i greater than the correlation with the class ($f_{ji} > f_{jc}$), erase feature f_j

FCBF Algorithm

input: $S(f_1, f_2, ..., f_N, C)$ // a training data set
δ // a predefined threshold
output: S_{best} // an optimal subset

```
1   begin
2     for i = 1 to N do begin
3       calculate SU_{i,c} for f_i;
4       if (SU_{i,c} ≥ δ)
5         append f_i to S'_{list};
6     end;
7     order S'_{list} in descending SU_{i,c} value;
```

Relevance Step

FCBF Algorithm (cont.)

```
8     f_p = getFirstElement(S'_list);
9     do begin
10       f_q = getNextElement(S'_list, f_p);
11       if (f_q <> NULL)
12          do begin
13             f'_q = f_q;
14             if (SU_p,q ≥ SU_q,c)
15                remove f_q from S'_list;
16                f_q = getNextElement(S'_list, f'_q);
17             else f_q = getNextElement(S'_list, f_q);
18          end until (f_q == NULL);
19       f_p = getNextElement(S'_list, f_p);
20    end until (f_p == NULL);
21 S_best = S'_list;
22 end;
```

Redundancy Step

Wrapper apprach: Evaluator

(8.5) Evaluator.

- wrapper approach: Classifier error rate.
  error_rate = classifier(feature subset candidate)
  if (error_rate < predefined threshold) select the feature subset
- feature selection loss its generality, but gain accuracy towards the classification task.
- computationally very costly.

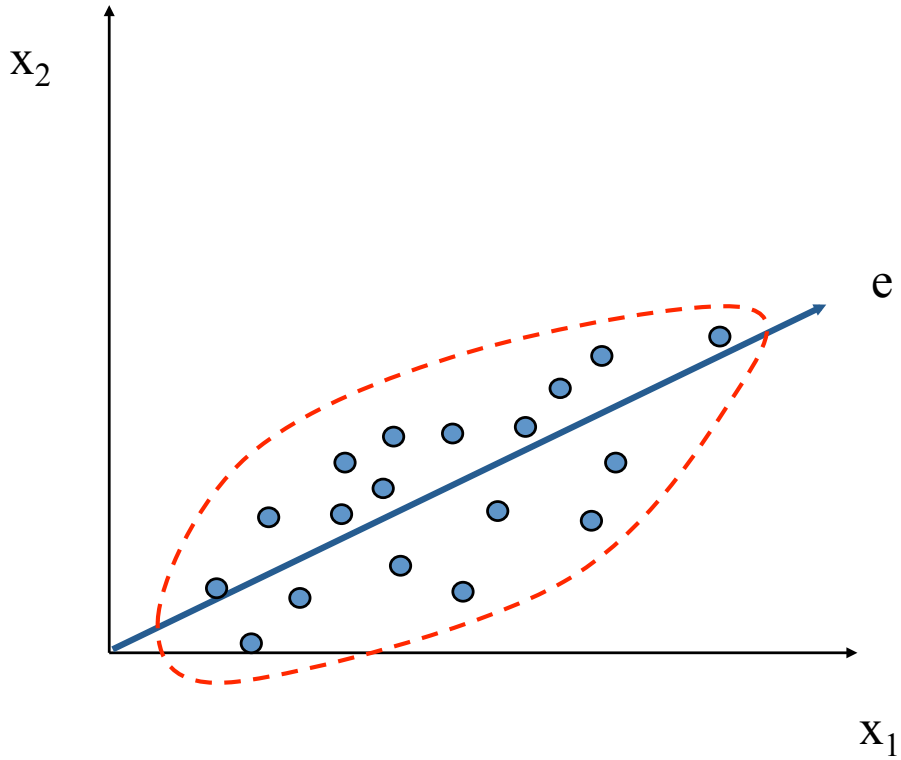
Feature Construction

- **Replacing the feature space**
  - Replacing the old feature with a linear (or non linear) combination of the previous attributes
  - Useful if there are some correlation between the attributes
  - If the attributes are independent the combination will be useless

- **Principal Techniques:**
  - Independent Component Analysis
  - Principal Component Analysis

Principal Component Analysis (PCA)

- Find a projection that captures the largest amount of variation in data
- The original data are projected onto a much smaller space, resulting in dimensionality reduction. We find the eigenvectors of the covariance matrix, and these eigenvectors define the new space



Principal Component Analysis (Steps)

- Given *N* data vectors from *n*-dimensions, find $k \leq n$ orthogonal vectors (*principal components*) that can be best used to represent data
  - Normalize input data: Each attribute falls within the same range
  - Compute *k* orthonormal (unit) vectors, i.e., *principal components*
  - Each input data (vector) is a linear combination of the *k* principal component vectors
  - The principal components are sorted in order of decreasing "significance" or strength
  - Since the components are sorted, the size of the data can be reduced by eliminating the *weak components*, i.e., those with low variance (i.e., using the strongest principal components, it is possible to reconstruct a good approximation of the original data)
- Works for numeric data only

Summary

- Important pre-processing in the Data Mining process
- Different strategies to follow
- First of all, understand the data and select a reasonable approach to redure the dimensionality