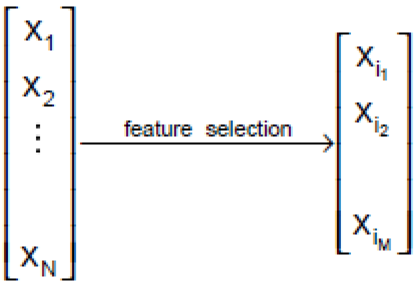
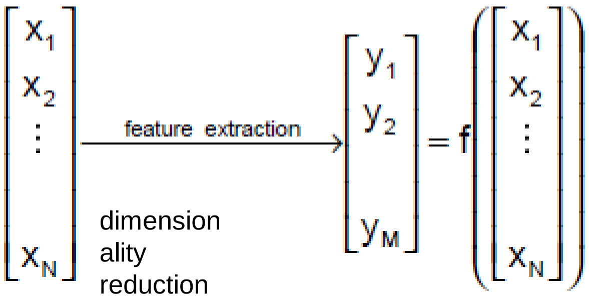
Feature selection

Feature Selection

- Given a set of **n** features, the goal of **feature selection** is to select a subset of **d** features (**d** < **n**) in order to minimize the classification error.





dimension
ality
reduction

- Why perform feature selection?
  - Data interpretation\knowledge discovery (insights into which factors which are most **representative** of your problem)
  - Curse of dimensionality (amount of data grows **exponentially** with # of features O(2^n))

- Fundamentally different from dimensionality reduction (we will discuss next time) based on feature combinations (i.e., **feature extraction**).

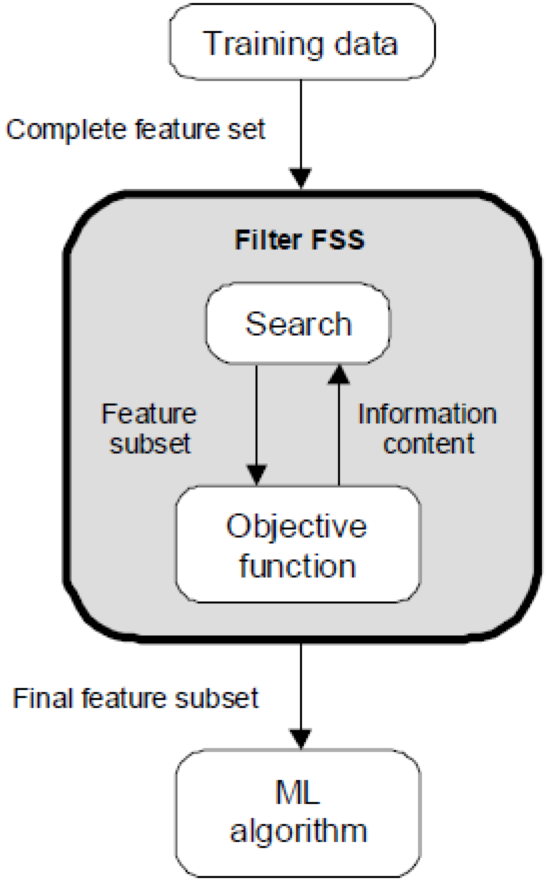
Feature Selection vs. Feature extraction

- Feature Selection
  - When classifying novel patterns, only a small number of features need to be computed (i.e., faster classification).
  - The measurement units (length, weight, etc.) of the features are preserved.

- Feature extraction (next time)
  - When classifying novel patterns, all features need to be computed.

Feature Selection Steps

- Feature selection is an **optimization** problem.
  - Step 1: Search the space of possible feature subsets.
  - Step 2: Pick the subset that is optimal or near-optimal with respect to some objective function.



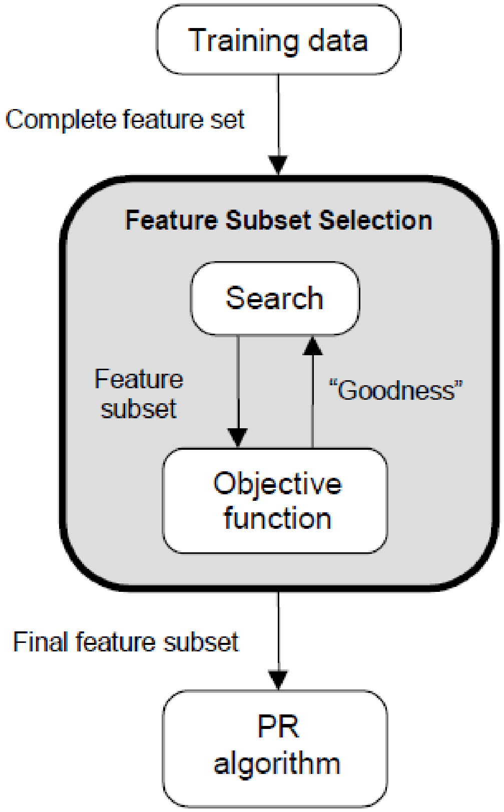
Feature Selection Steps (cont'd)

Search strategies

- Optimal
- Heuristic

Evaluation strategies

- Filter methods
- Wrapper methods



Search Strategies

- Assuming n features, an exhaustive search would require:
  - Examining all $\binom{n}{d}$ possible subsets of size d.
  - Selecting the subset that performs the best according to the criterion function.

•The number of subsets grows combinatorially, making exhaustive search impractical.

•In practice, heuristics are used to speed-up search but they **cannot** guarantee optimality.

Naïve Search

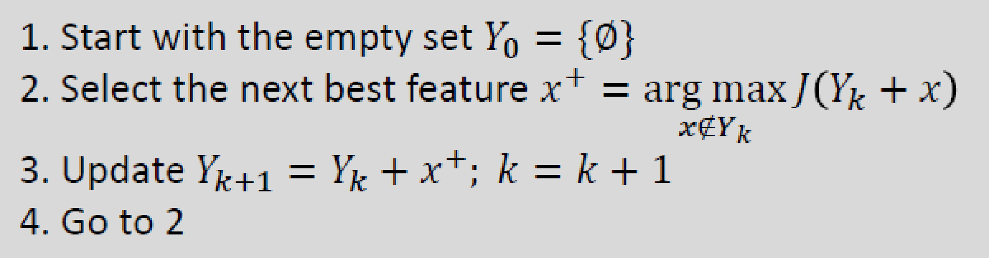
- Sort the given n features in order of their probability of correct recognition.
- Select the top d features from this sorted list.
- Disadvantage
  - Correlation among features is not considered.
  - The best pair of features may not even contain the best individual feature.

Sequential forward selection (SFS)

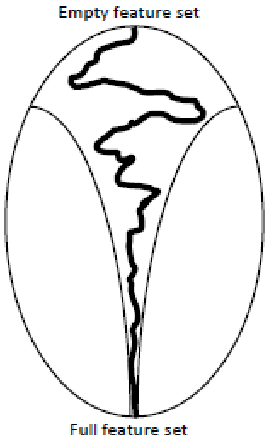
(heuristic search)

- First, the best **single** feature is selected (i.e., using some criterion function).
- Then, **pairs** of features are formed using one of the remaining features and this best feature, and the best pair is selected.
- Next, **triplets** of features are formed using one of the remaining features and these two best features, and the best triplet is selected.
- This procedure continues until a predefined number of features are selected.

SFS performs best when the optimal subset is small.

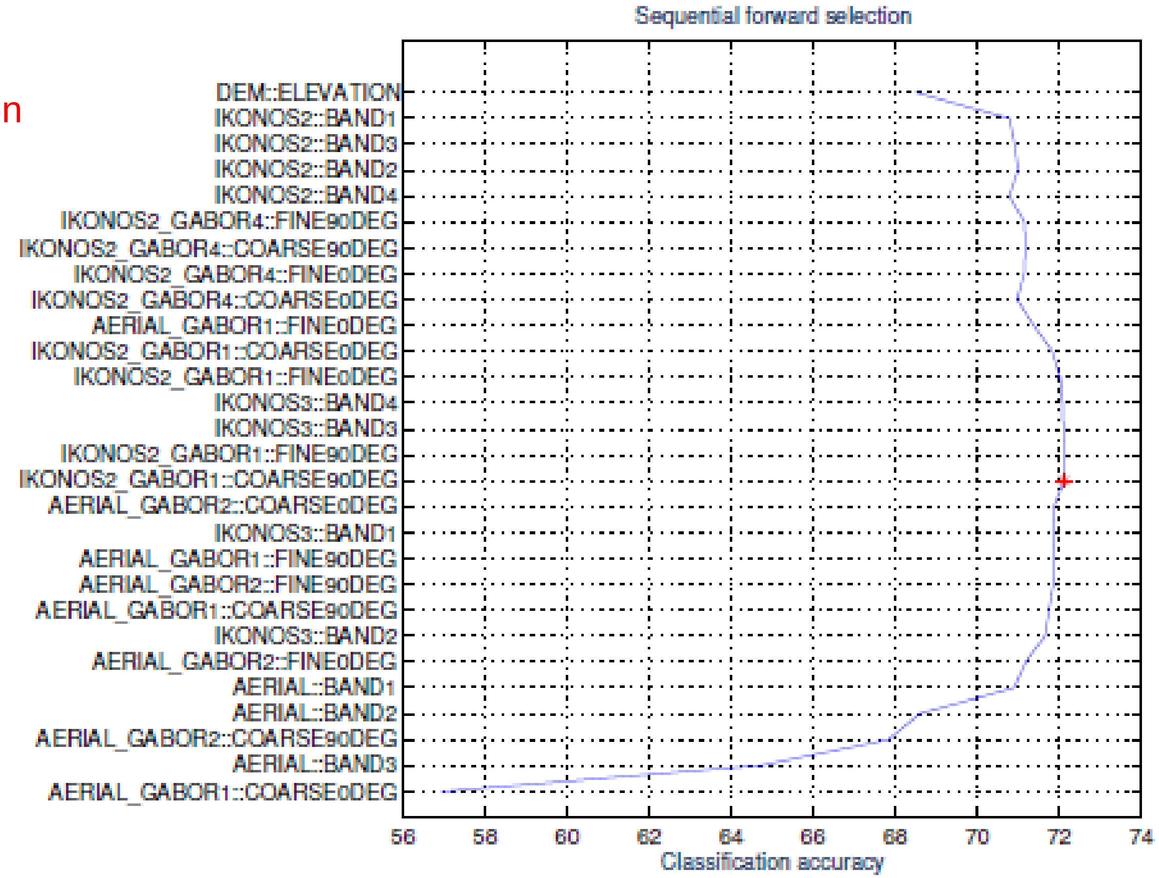


1. Start with the empty set $Y_0 = \{\emptyset\}$
2. Select the next best feature $x^+ = \underset{x \notin Y_k}{\arg\max}\, J(Y_k + x)$
3. Update $Y_{k+1} = Y_k + x^+;\ k = k + 1$
4. Go to 2



Example

features added at each iteration

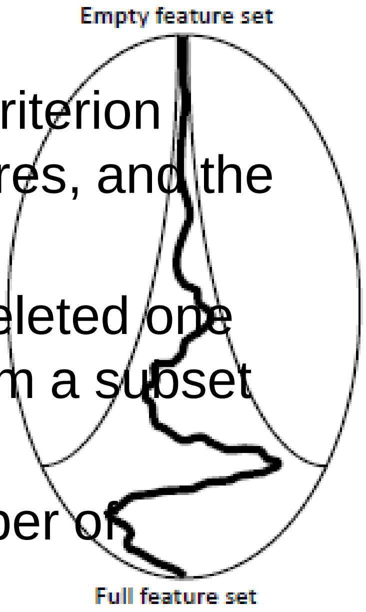


Results of sequential forward feature selection for classification of a satellite image using 28 features. x-axis shows the classification accuracy (%) and y-axis shows the features added at each iteration (the first iteration is at the bottom). The highest accuracy value is shown with a star.

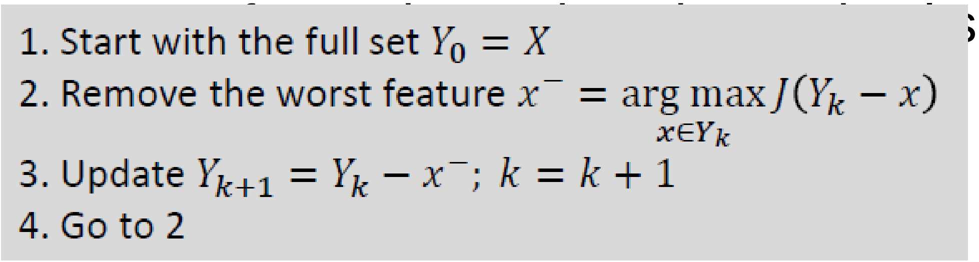
Sequential backward selection (SBS)

(heuristic search)

- First, the criterion function is computed for all **n** features.
- Then, each feature is deleted one at a time, the criterion function is computed for all subsets with **n-1** features, and the worst feature is discarded.
- Next, each feature among the remaining **n-1** is deleted one at a time, and the worst feature is discarded to form a subset with **n-2** features.
- This procedure continues until a predefined number of features are left.



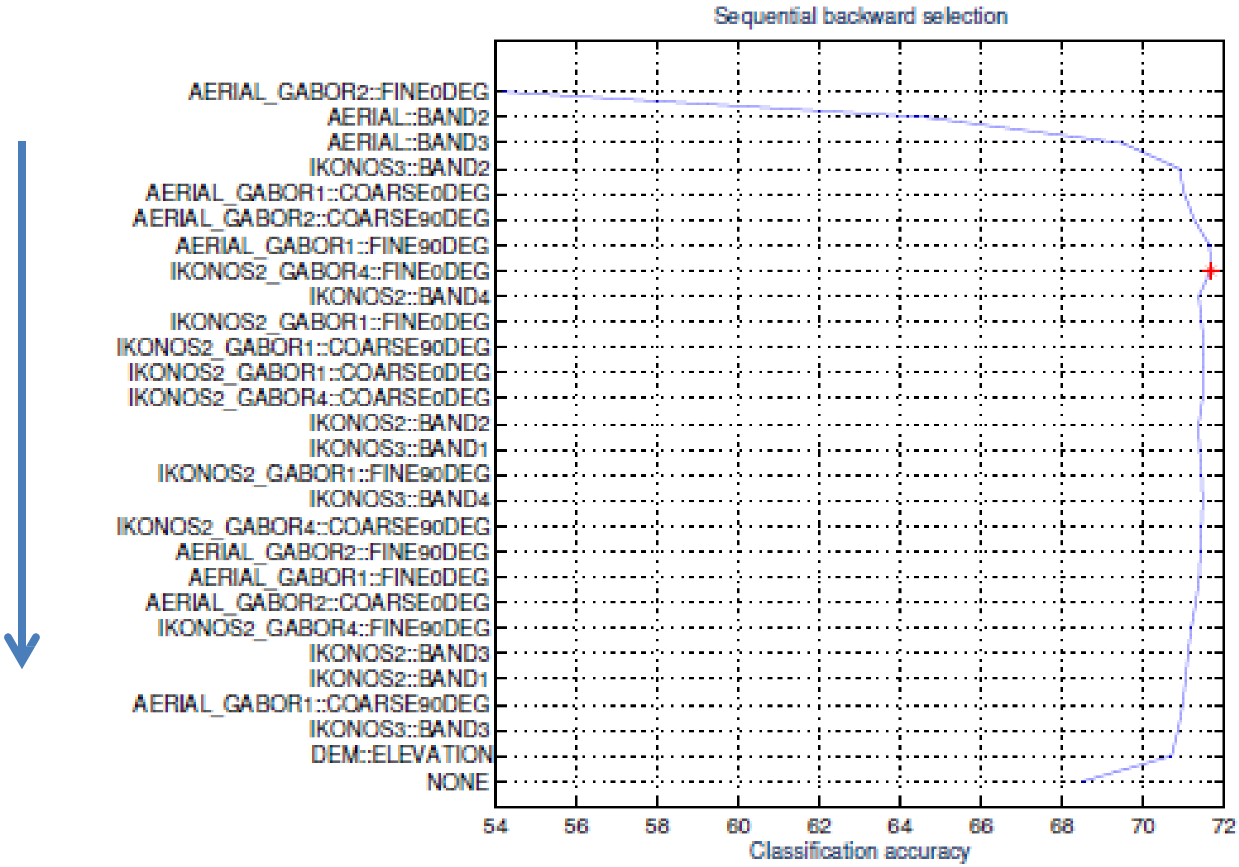
subset is large.



1. Start with the full set $Y_0 = X$
2. Remove the worst feature $x^- = \arg\max_{x \in Y_k} J(Y_k - x)$
3. Update $Y_{k+1} = Y_k - x^-;\ k = k + 1$
4. Go to 2

Example

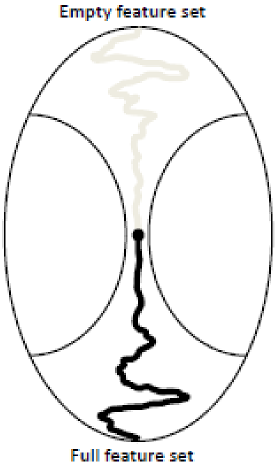
features removed at each iteration

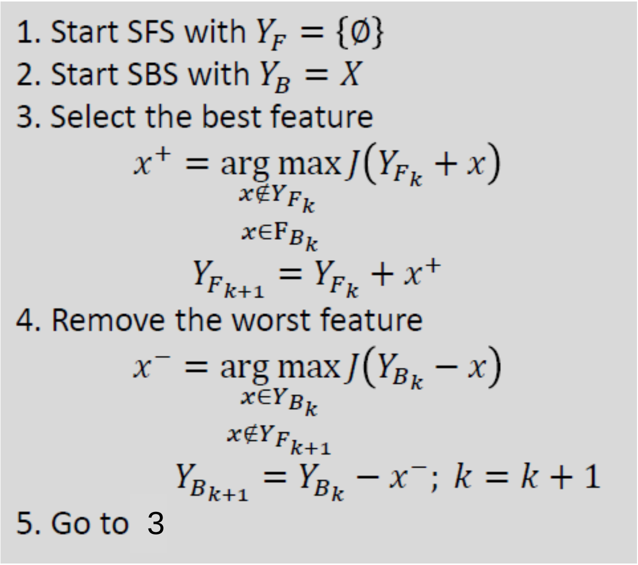


Results of sequential backward feature selection for classification of a satellite image using 28 features. x-axis shows the classification accuracy (%) and y-axis shows the features removed at each iteration (the first iteration is at the top). The highest accuracy value is shown with a star.

Bidirectional Search (BDS)

- BDS applies SFS and SBS simultaneously:
  - SFS is performed from the empty set.
  - SBS is performed from the full set.
- To guarantee that SFS and SBS converge to the same solution:
  - Features already selected by SFS are not removed by SBS.
  - Features already removed by SBS are not added by SFS.





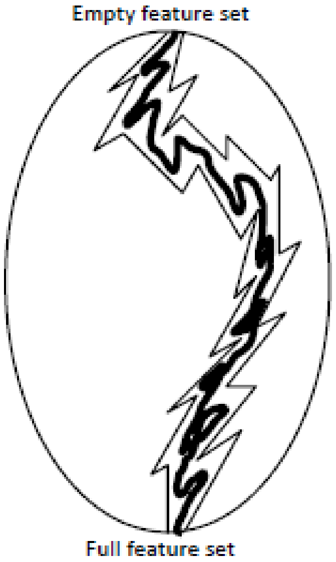
```
1. Start SFS with $Y_F = \{\emptyset\}$
2. Start SBS with $Y_B = X$
3. Select the best feature
        $x^+ = \underset{\substack{x \notin Y_{F_k} \\ x \in F_{B_k}}}{\arg\max} J(Y_{F_k} + x)$
        $Y_{F_{k+1}} = Y_{F_k} + x^+$
4. Remove the worst feature
        $x^- = \underset{\substack{x \in Y_{B_k} \\ x \notin Y_{F_{k+1}}}}{\arg\max} J(Y_{B_k} - x)$
        $Y_{B_{k+1}} = Y_{B_k} - x^-;\ k = k + 1$
5. Go to 3
```

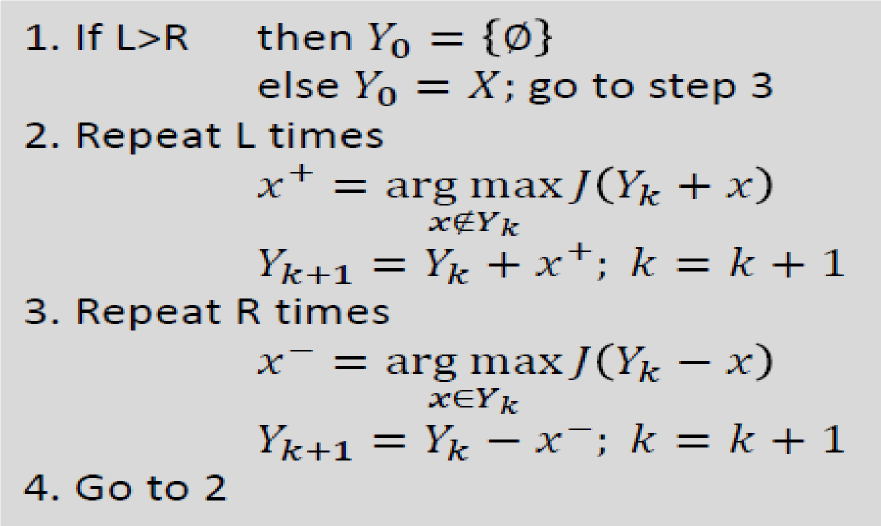
Limitations of SFS and SBS

- The main limitation of SFS is that it is unable to remove features that become non useful after the addition of other features.
- The main limitation of SBS is its inability to reevaluate the usefulness of a feature after it has been discarded.
- We will examine some generalizations of SFS and SBS:
  - Plus-L, minus-R” selection (LRS)
  - Sequential floating forward/backward selection (SFFS and SFBS)

"Plus-L, minus-R" selection (LRS)

- A generalization of SFS and SBS
  - If L>R, LRS starts from the empty set and:
    - Repeatedly add L features
    - Repeatedly remove R features
  - If L<R, LRS starts from the full set and:
    - Repeatedly removes R features
    - Repeatedly add L features



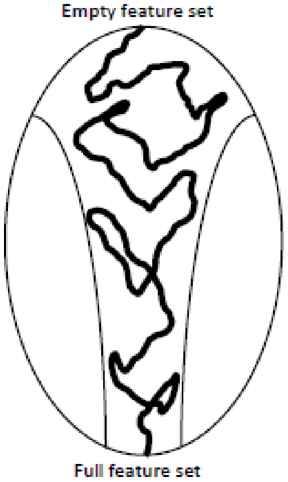


```
1. If L>R    then $Y_0 = \{\emptyset\}$
             else $Y_0 = X$; go to step 3
2. Repeat L times
             $x^+ = \arg\max_{x \notin Y_k} J(Y_k + x)$
             $Y_{k+1} = Y_k + x^+;\ k = k + 1$
3. Repeat R times
             $x^- = \arg\max_{x \in Y_k} J(Y_k - x)$
             $Y_{k+1} = Y_k - x^-;\ k = k + 1$
4. Go to 2
```

Its main limitation is the lack of a theory to help choose the optimal values of L and R.

Sequential floating forward/backward selection (SFFS and SFBS)

- An extension to LRS:
  - Rather than fixing the values of L and R, floating methods determine these values from the data.
  - The dimensionality of the subset during the search can be thought to be "floating" up and down

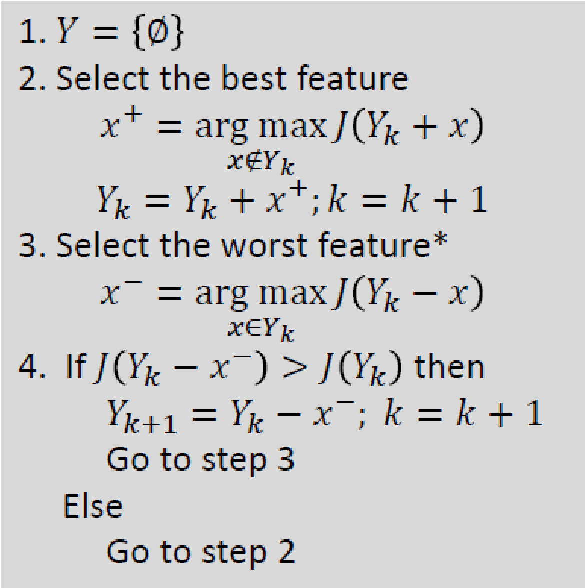


- Two floating methods:
  - Sequential floating forward selection (SFFS)
  - Sequential floating backward selection (SFBS)

P. Pudil, J. Novovicova, J. Kittler, Floating search methods in feature selection, Pattern Recognition Lett. 15 (1994) 1119–1125.

Sequential floating forward selection (SFFS)

- Sequential floating forward selection (SFFS) starts from the empty set.
- After each forward step, SFFS performs backward steps as long as the objective function increases.



```
1. $Y = \{\emptyset\}$
2. Select the best feature
        $x^{+} = \underset{x \notin Y_k}{\arg\max}\, J(Y_k + x)$
        $Y_k = Y_k + x^{+}; k = k + 1$
3. Select the worst feature*
        $x^{-} = \underset{x \in Y_k}{\arg\max}\, J(Y_k - x)$
4. If $J(Y_k - x^{-}) > J(Y_k)$ then
        $Y_{k+1} = Y_k - x^{-};\ k = k + 1$
        Go to step 3
   Else
        Go to step 2
```

*Notice that you'll need to do book-keeping to avoid infinite loops

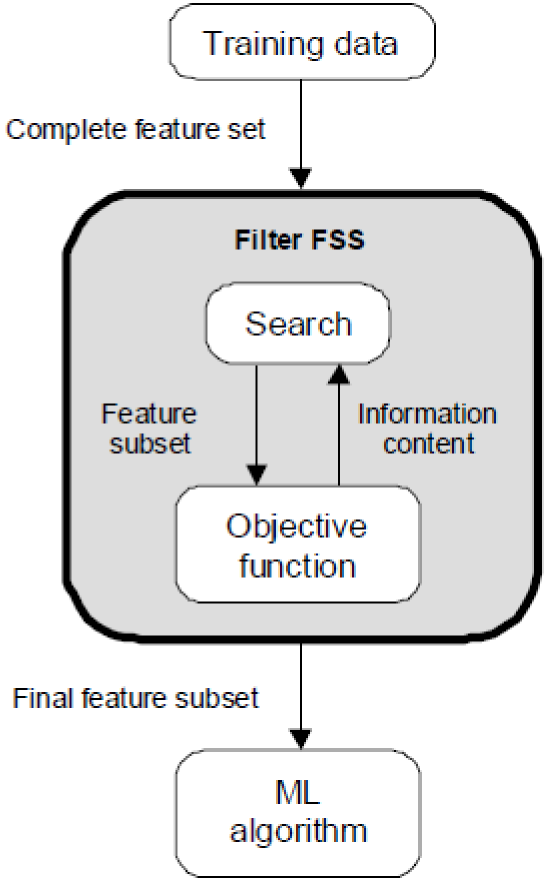
Sequential floating backward selection (SFBS)

- Sequential floating backward selection (SFBS) starts from the full set.

- After each backward step, SFBS performs forward steps as long as the objective function increases.

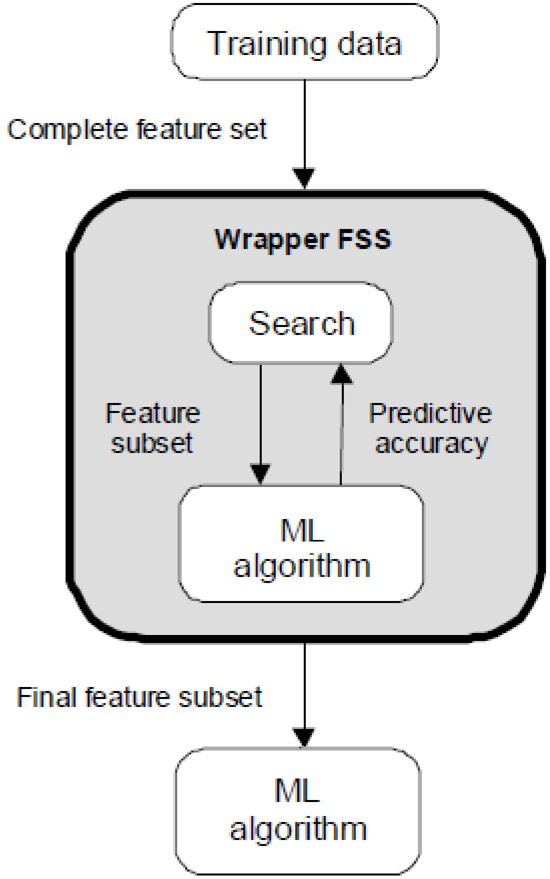
Evaluation Strategies

- Filter Methods
  - Evaluation is **independent** of the classification algorithm.
  - The objective function evaluates feature subsets by their information content, typically interclass distance, statistical dependence or information-theoretic measures.

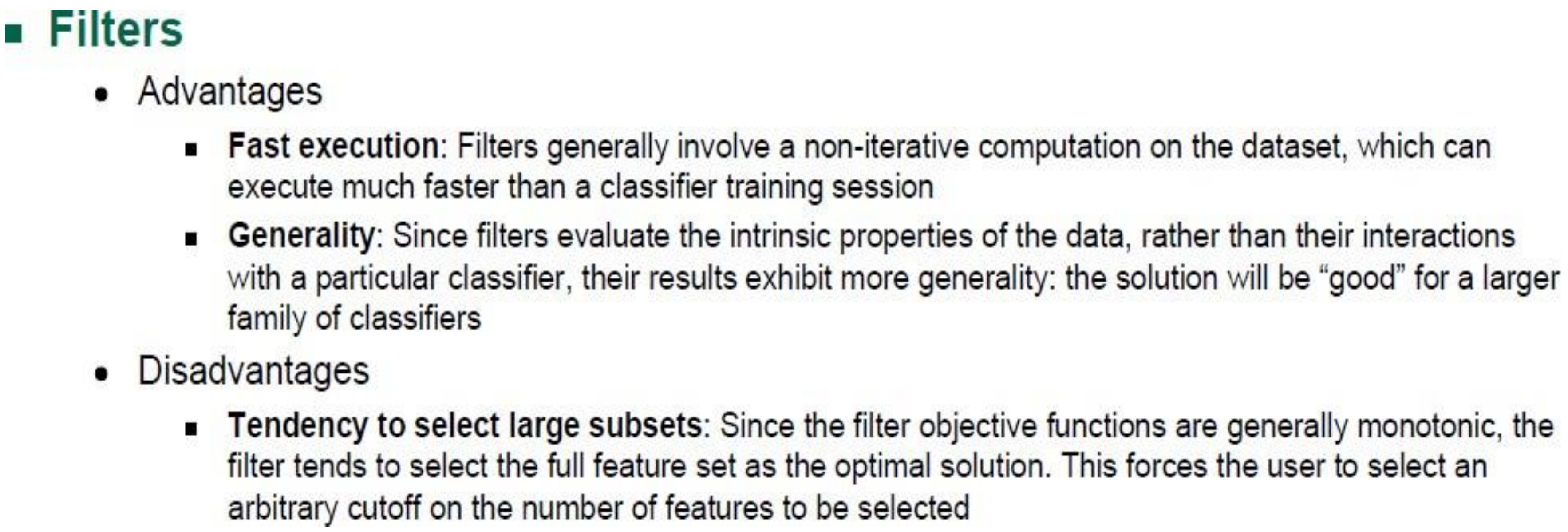


Evaluation Strategies

- Wrapper Methods
  - Evaluation uses criteria **related** to the classification algorithm.
  - The objective function is a pattern classifier, which evaluates feature subsets by their predictive accuracy (recognition rate on test data) by statistical resampling or cross-validation.

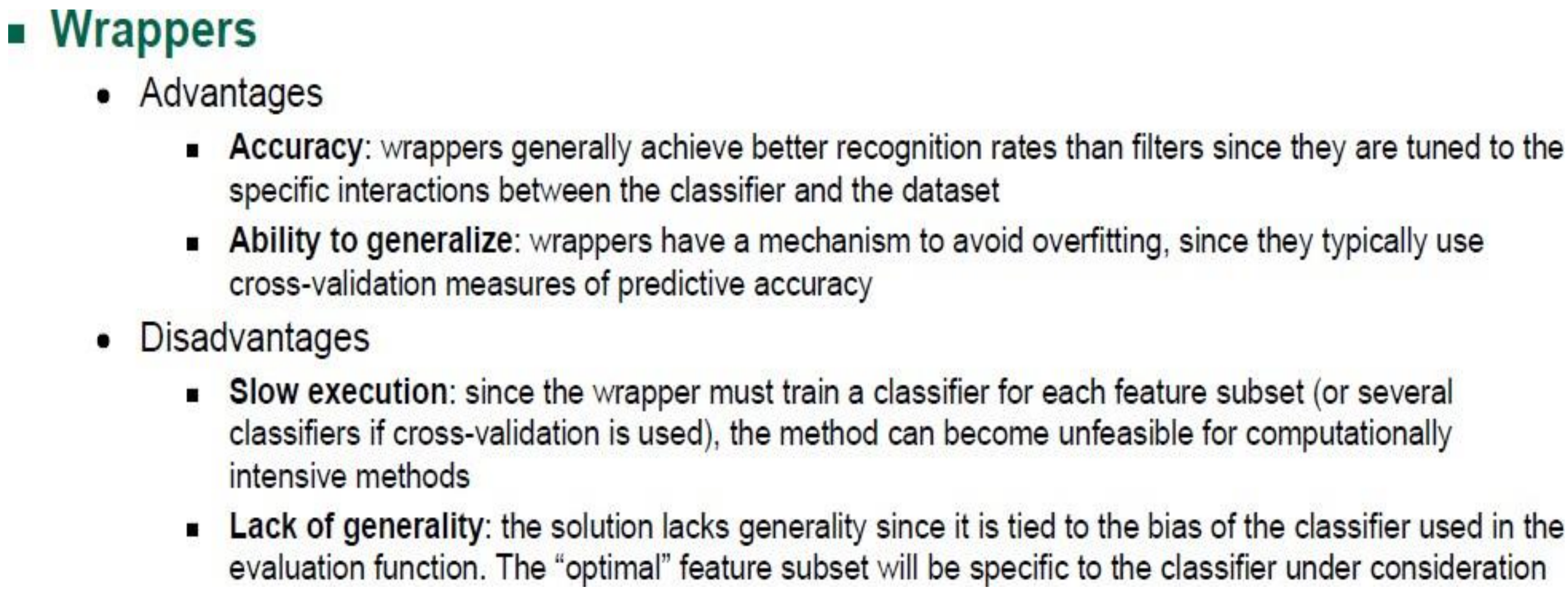


Filter vs. Wrapper Approaches



- **Filters**
  - Advantages
    - **Fast execution**: Filters generally involve a non-iterative computation on the dataset, which can execute much faster than a classifier training session
    - **Generality**: Since filters evaluate the intrinsic properties of the data, rather than their interactions with a particular classifier, their results exhibit more generality: the solution will be "good" for a larger family of classifiers
  - Disadvantages
    - **Tendency to select large subsets**: Since the filter objective functions are generally monotonic, the filter tends to select the full feature set as the optimal solution. This forces the user to select an arbitrary cutoff on the number of features to be selected

Filter vs. Wrapper Approaches



- **Wrappers**
  - Advantages
    - **Accuracy**: wrappers generally achieve better recognition rates than filters since they are tuned to the specific interactions between the classifier and the dataset
    - **Ability to generalize**: wrappers have a mechanism to avoid overfitting, since they typically use cross-validation measures of predictive accuracy
  - Disadvantages
    - **Slow execution**: since the wrapper must train a classifier for each feature subset (or several classifiers if cross-validation is used), the method can become unfeasible for computationally intensive methods
    - **Lack of generality**: the solution lacks generality since it is tied to the bias of the classifier used in the evaluation function. The "optimal" feature subset will be specific to the classifier under consideration

Feature Selection

- Has two-fold advantage of providing some interpretation of the data and making the learning problem easier
- Finding global optimum impractical in most situations, rely on heuristics instead (greedy\random search)
- Filtering is fast and general but can pick a large # of features
- Wrapping considers model bias but is MUCH slower due to training multiple models

What is feature selection?

- Consider our training data as a matrix where each row is a vector and each column is a dimension.
- For example consider the matrix for the data x_1=(1, 10, 2), x_2=(2, 8, 0), and x_3=(1, 9, 1)
- We call each dimension a feature or a column in our matrix.

Feature selection

- Useful for high dimensional data such as genomic DNA and text documents.
- Methods
  - Univariate (looks at each feature independently of others)
    - Pearson correlation coefficient
    - F-score
    - Chi-square
    - Signal to noise ratio
    - And more such as mutual information, relief
  - Multivariate (considers all features simultaneously)
    - Dimensionality reduction algorithms (PCA)
    - Linear classifiers such as support vector machine
    - Recursive feature elimination

Feature selection

- Methods are used to rank features by rank/importance
- Ranking cut-off is determined by user
- Univariate methods measure some type of correlation between two random variables. We apply them to machine learning by setting one variable to be the label (y_i) and the other to be a fixed feature (x_{ij} for fixed j)

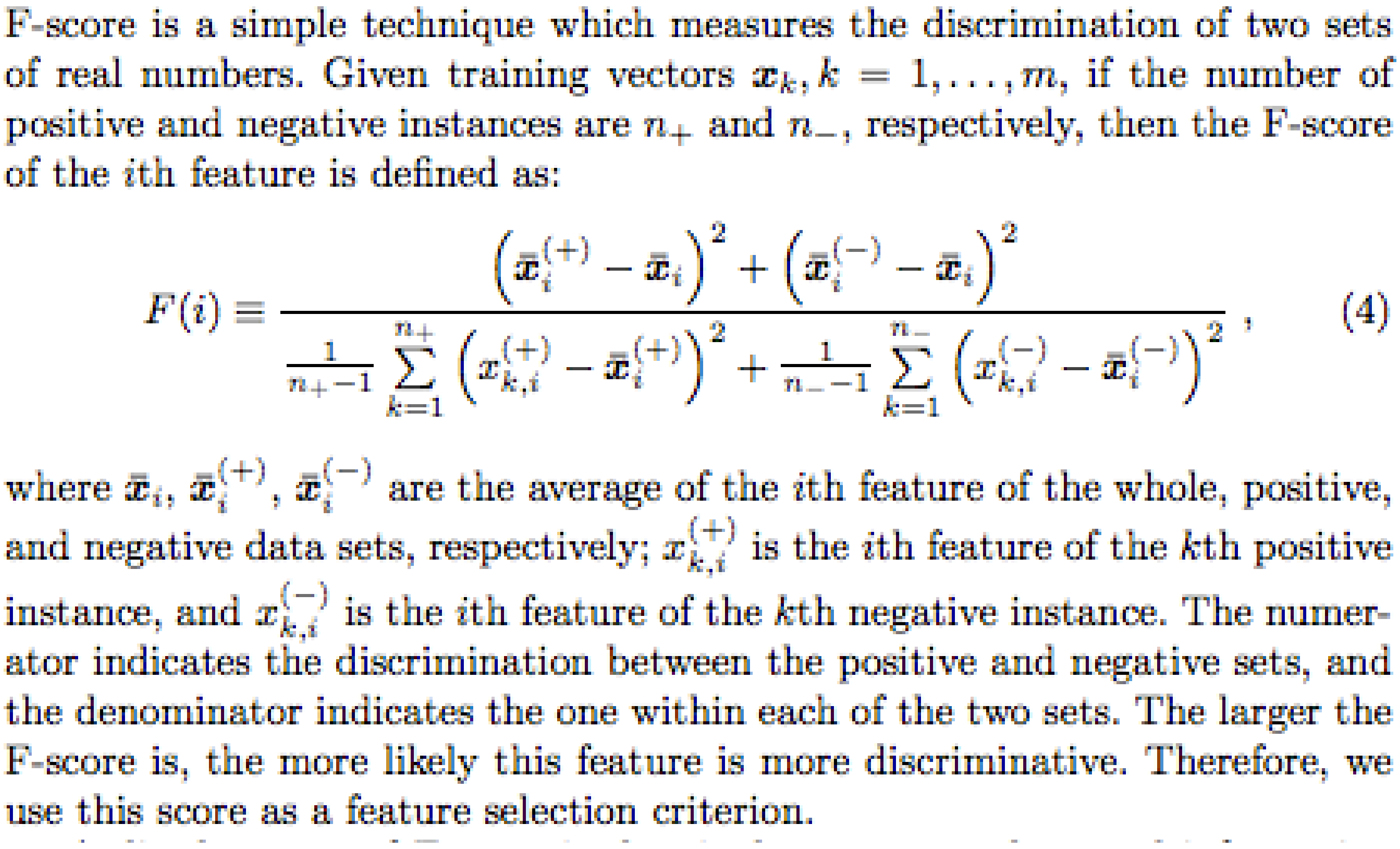
Pearson correlation coefficient

- Measures the correlation between two variables
- Formulas:
  - Covariance(X,Y) = E((X-μ_X)(Y-μ_Y))
  - Correlation(X,Y)= Covariance(X,Y)/$\sigma_X\sigma_Y$
  - Pearson correlation =

$$r = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i - \bar{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i - \bar{Y})^2}}$$

- The correlation r is between -1 and 1. A value of 1 means perfect positive correlation and -1 in the other direction

F-score



F-score is a simple technique which measures the discrimination of two sets of real numbers. Given training vectors $\boldsymbol{x}_k, k = 1, \ldots, m$, if the number of positive and negative instances are n_+ and n_-, respectively, then the F-score of the ith feature is defined as:

$$F(i) \equiv \frac{\left(\bar{\boldsymbol{x}}_i^{(+)} - \bar{\boldsymbol{x}}_i\right)^2 + \left(\bar{\boldsymbol{x}}_i^{(-)} - \bar{\boldsymbol{x}}_i\right)^2}{\frac{1}{n_+-1}\sum_{k=1}^{n_+}\left(x_{k,i}^{(+)} - \bar{\boldsymbol{x}}_i^{(+)}\right)^2 + \frac{1}{n_--1}\sum_{k=1}^{n_-}\left(x_{k,i}^{(-)} - \bar{\boldsymbol{x}}_i^{(-)}\right)^2}, \qquad (4)$$

where $\bar{\boldsymbol{x}}_i$, $\bar{\boldsymbol{x}}_i^{(+)}$, $\bar{\boldsymbol{x}}_i^{(-)}$ are the average of the ith feature of the whole, positive, and negative data sets, respectively; $x_{k,i}^{(+)}$ is the ith feature of the kth positive instance, and $x_{k,i}^{(-)}$ is the ith feature of the kth negative instance. The numerator indicates the discrimination between the positive and negative sets, and the denominator indicates the one within each of the two sets. The larger the F-score is, the more likely this feature is more discriminative. Therefore, we use this score as a feature selection criterion.

From Lin and Chen, 2006

Signal to noise ratio

- Difference in means divided by difference in standard deviation between the two classes
- S2N(X,Y) = $(\mu_X - \mu_Y)/(\sigma_X + \sigma_Y)$
- Large values indicate a strong correlation

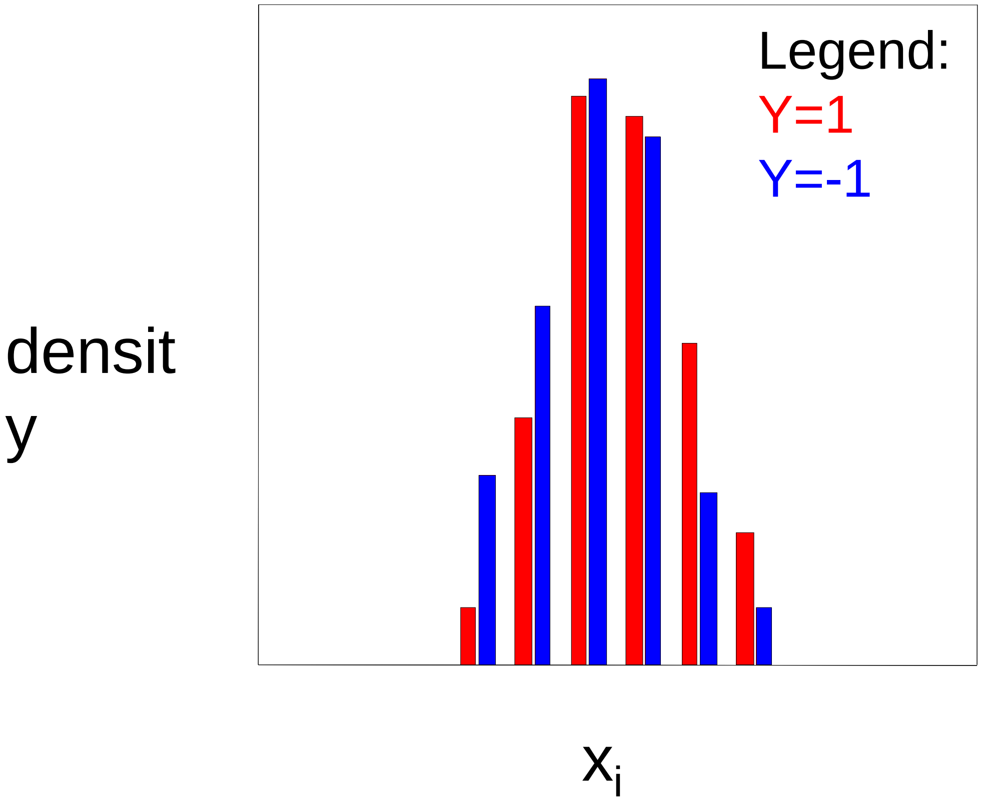
Limitations

- Unclear how to tell in advance if feature selection will work
  - Only known way is to check but for very high dimensional data (at least half a million features) it helps most of the time
- How many features to select?
  - Perform cross-validation

Filtering

- Basic idea: assign score to each feature x indicating how "related" x and the class y are
  - Intuition: if $x=y$ for all instances, then x is great no matter what our model is; x contains all information needed to predict y
- Pick the $\boldsymbol{n}$ highest scoring features to keep

Individual Feature Irrelevance

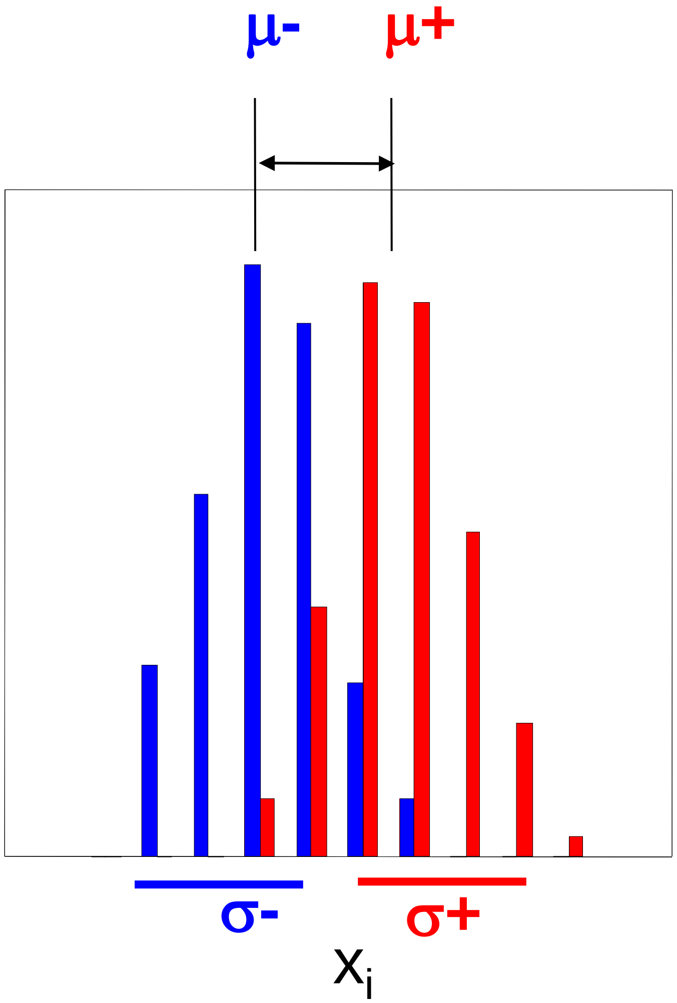


$P(X_i, Y) = P(X_i)\, P(Y)$

$P(X_i | Y) = P(X_i)$

$P(X_i | Y=1) = P(X_i | Y=-1)$

Individual Feature Relevance



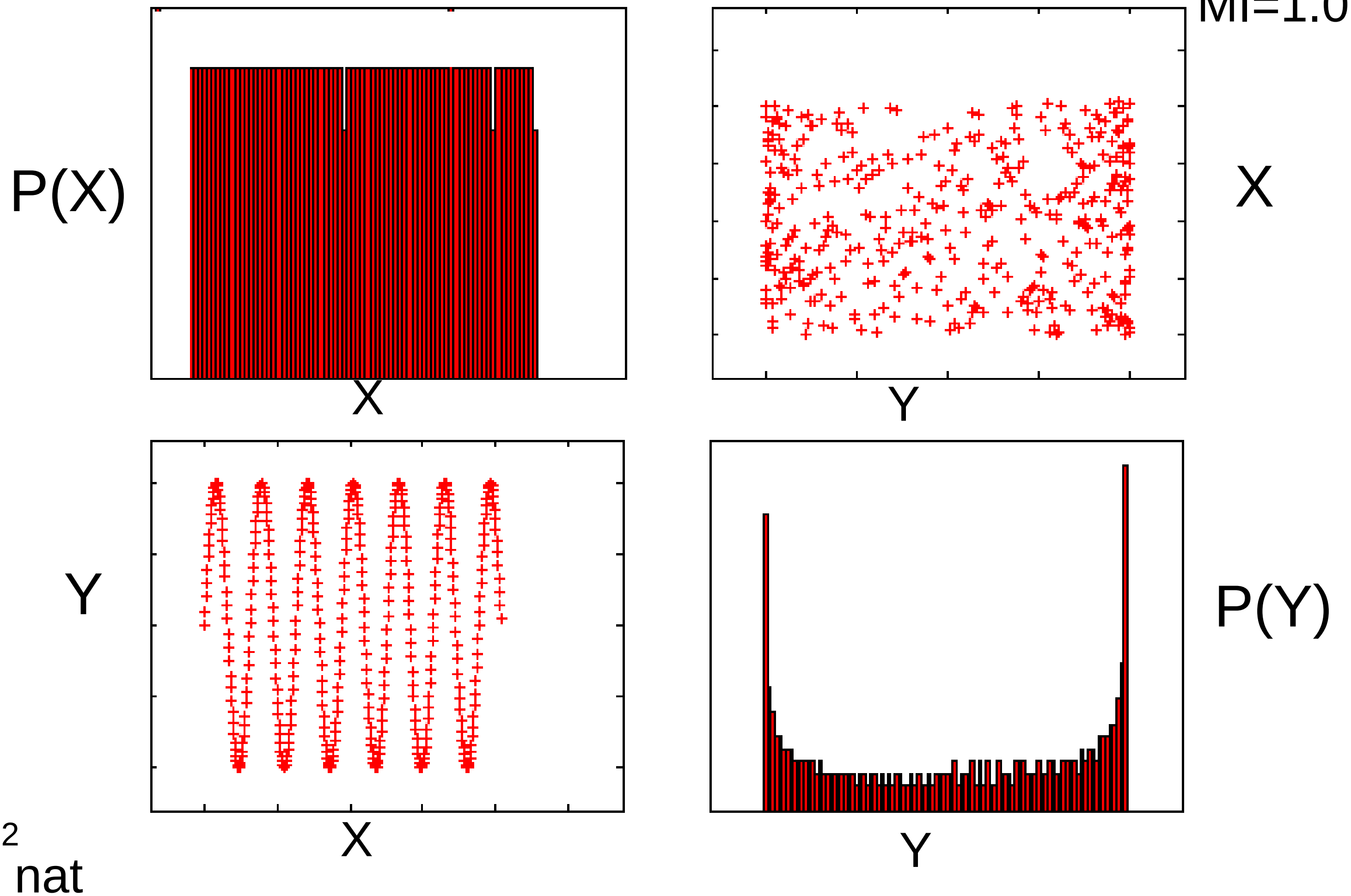
Univariate Dependence

- Independence:

  $$P(X, Y) = P(X)\,P(Y)$$

- Measures of dependence:
  - Mutual Information (see notes from board)
  - Correlation (see notes from board)

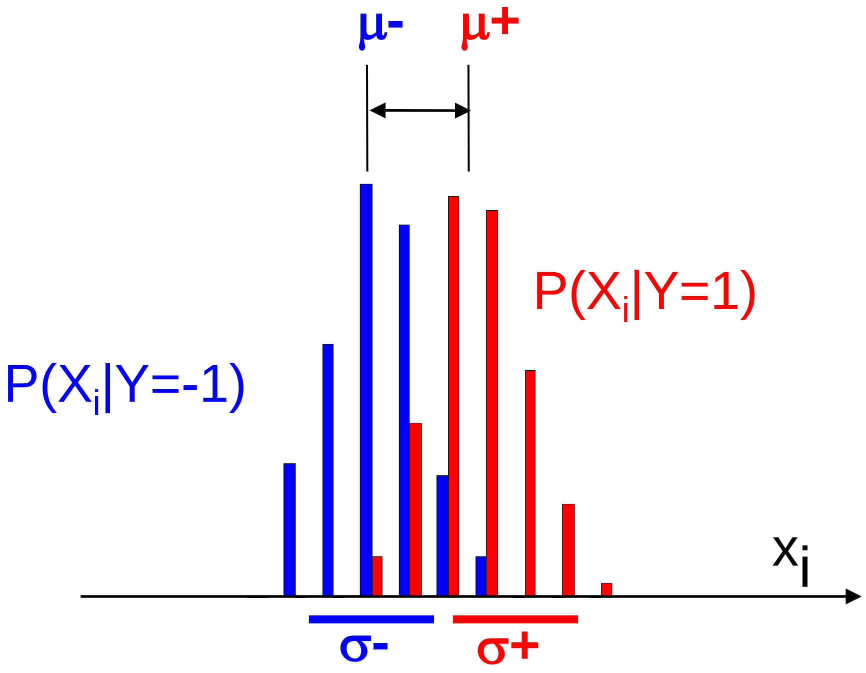
Correlation and MI



R=0.02
MI=1.03 nat

R=0.00$_{02}$
MI=1.65 nat

T-test



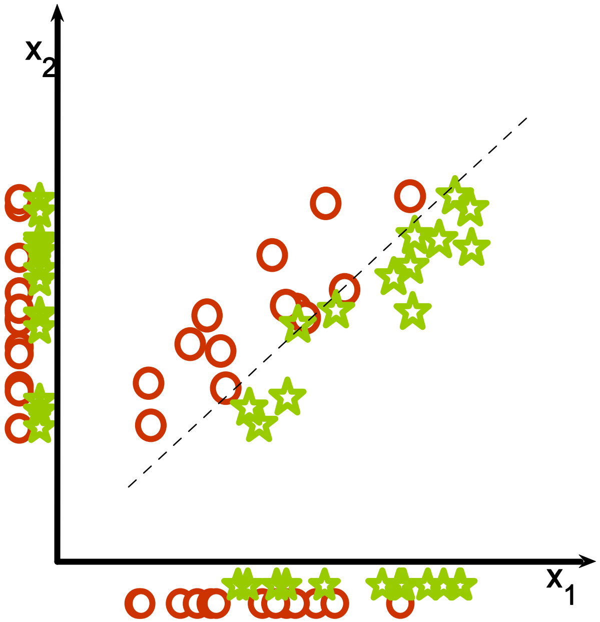
- Normally distributed classes, equal variance σ^2 unknown; estimated from data as σ^2_{within}.
- Null hypothesis H_0: $\mu+ = \mu-$
- T statistic: If H_0 is true,

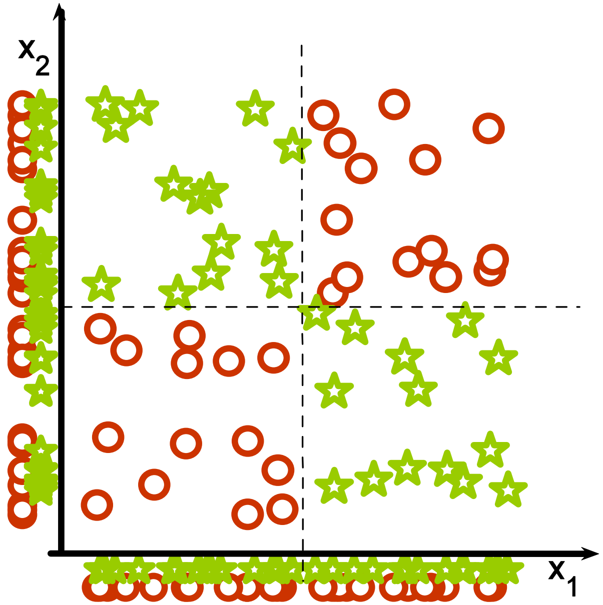
$$t = (\mu+ - \mu-)/(\sigma_{within}\sqrt{1/m^+ + 1/m^-}) \sim \text{Student}(m^+ + m^- - 2 \text{ d.f.})$$

T-test

Other ideas for Univariate Feature Selection?

Considering each feature alone may fail





Guyon-Elisseeff, JMLR 2004; Springer 2006

Multivariate Filter Methods?

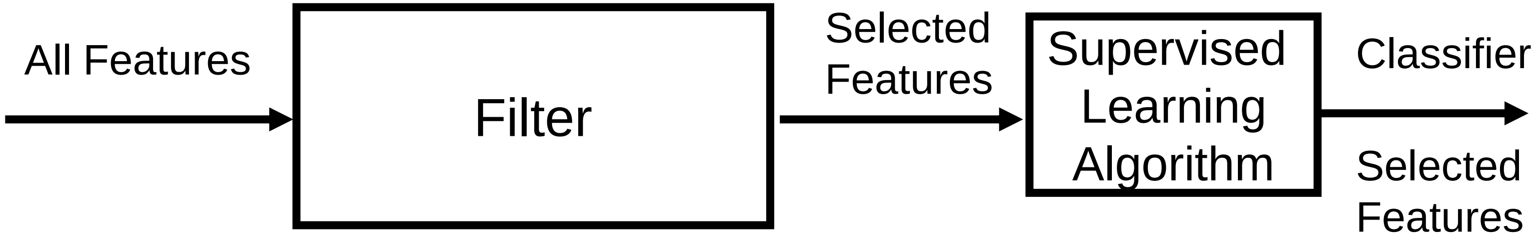
Filtering

- Advantages:
  - Fast, simple to apply

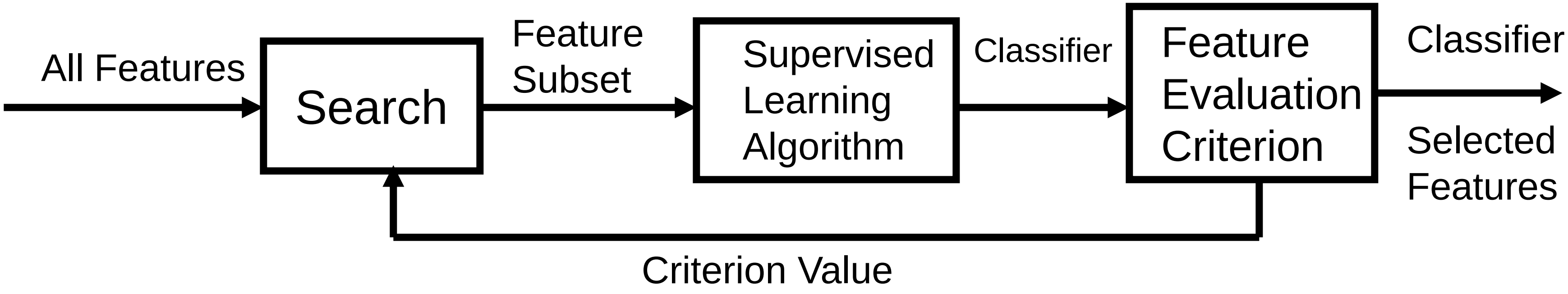
- Disadvantages:
  - Doesn’t take into account which learning algorithm will be used
  - Doesn’t take into account correlations between features, just correlation of each feature to the class label

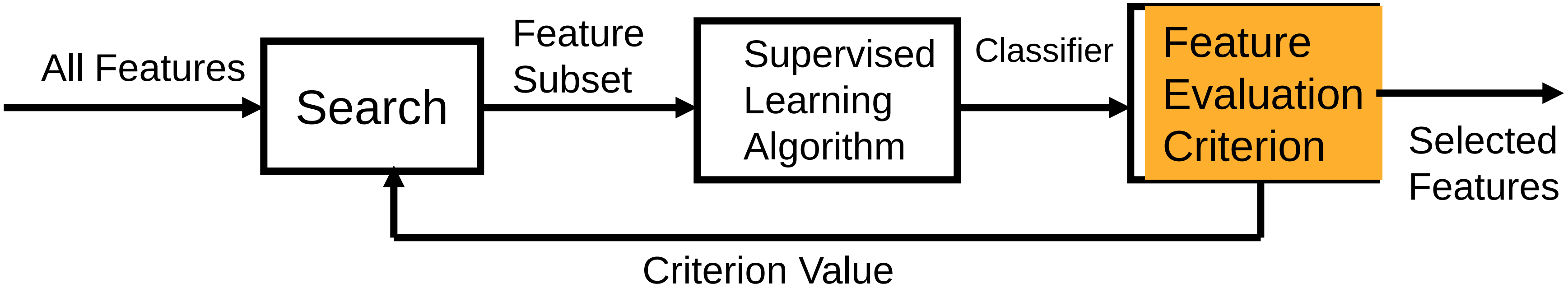
Feature Selection Methods

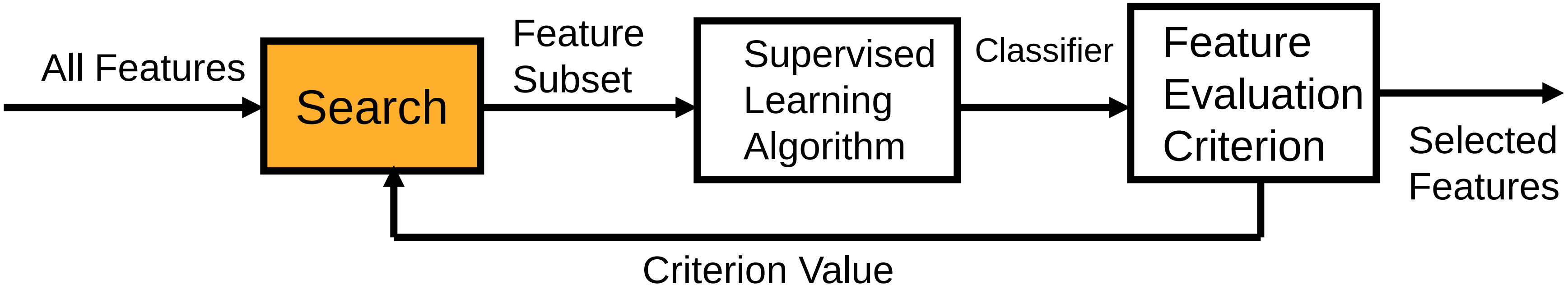
Filter:



Wrapper:




All Features
Search
Feature Subset
Supervised Learning Algorithm
Classifier
Feature Evaluation Criterion
Selected Features
Criterion Value



All Features

Search

Feature Subset

Supervised Learning Algorithm

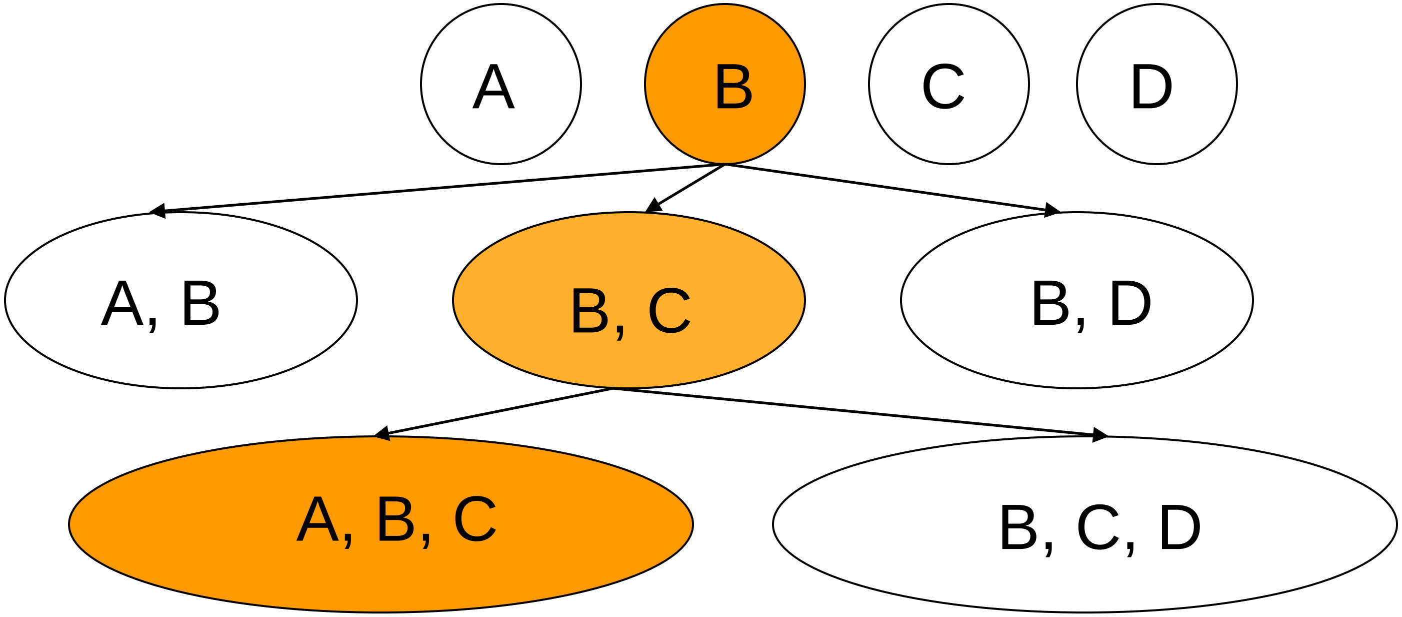
Classifier

Feature Evaluation Criterion

Selected Features

Criterion Value

Search Method: sequential forward search



A

B

C

D

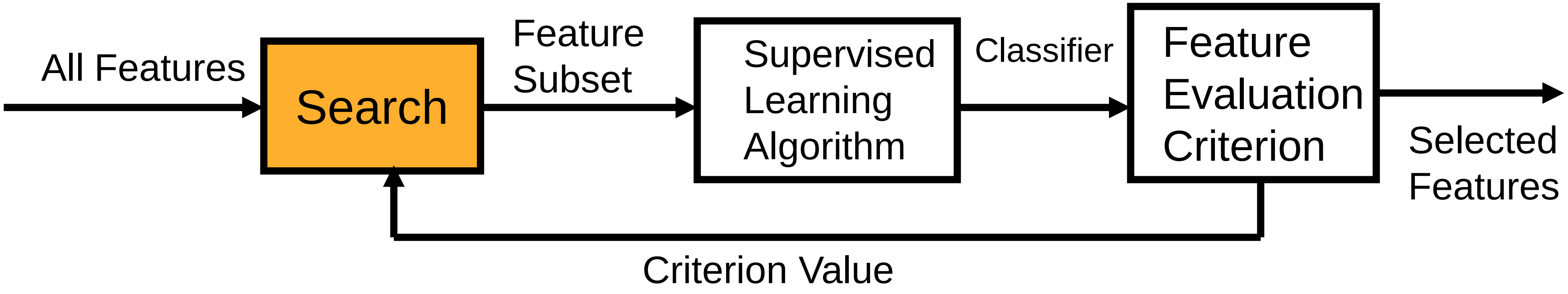
A, B

B, C

B, D

A, B, C

B, C, D



All Features

Search

Feature Subset

Supervised Learning Algorithm

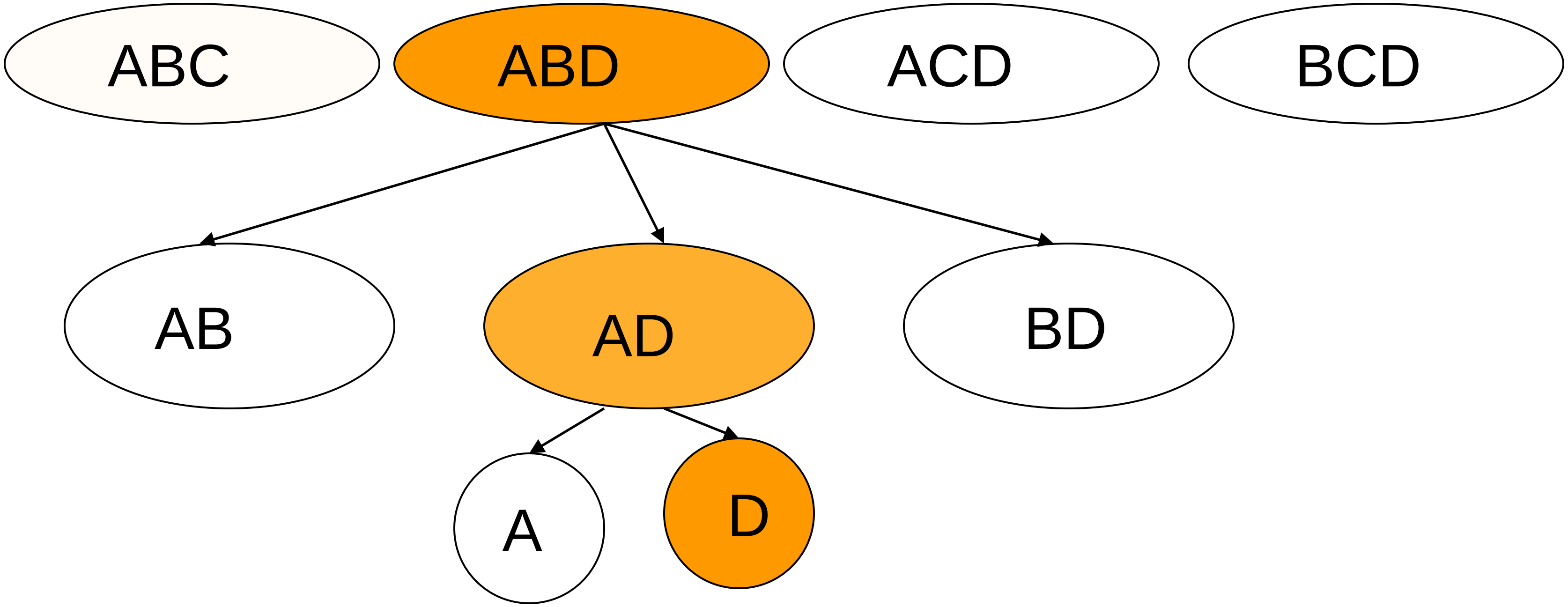
Classifier

Feature Evaluation Criterion

Selected Features

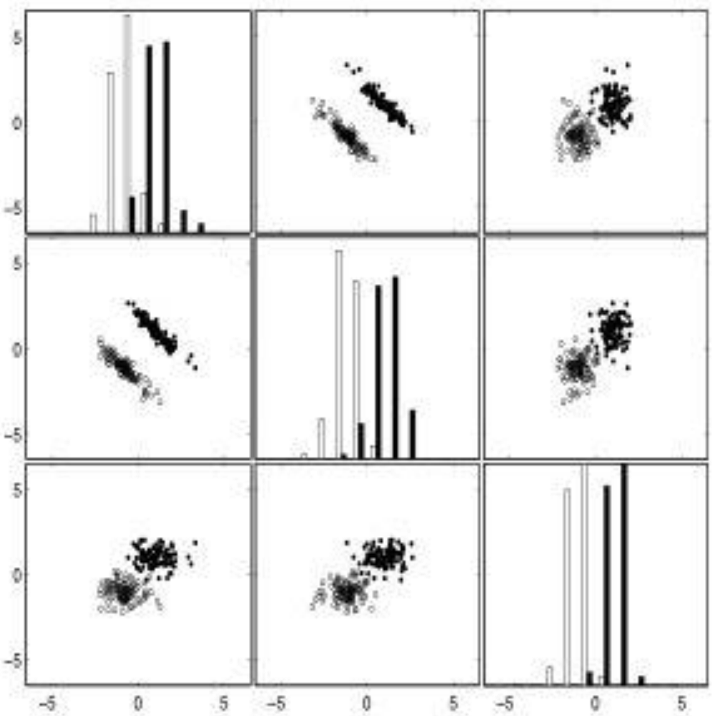
Criterion Value

Search Method: sequential backward elimination



- **Forward or backward selection?** Of the three variables of this example, the third one separates the two classes best by itself (bottom right histogram). It is therefore the best candidate in a forward selection process. Still, the two other variables are better taken together than any subset of two including it. A backward selection method may perform better in this case.

GUYON AND ELISSEEFF



Model search

- More sophisticated search strategies exist
  - Best-first search
  - Stochastic search
  - See "Wrappers for Feature Subset Selection", Kohavi and John 1997
- Other objective functions exist which add a model-complexity penalty to the training error
  - AIC, BIC

Regularization

- In certain cases, we can move model selection *into* the induction algorithm
  - Only have to fit one model; more efficient
- This is sometimes called an **embedded** feature selection algorithm

Regularization

- Regularization: add model complexity penalty to training error.
- $J(\boldsymbol{w}) = L(\boldsymbol{w}) + C\|\boldsymbol{w}\|_p = \sum_{i=1}^{n}(y_i - \boldsymbol{w}^\top \boldsymbol{x}_i)^2 + C\|\boldsymbol{w}\|_p$

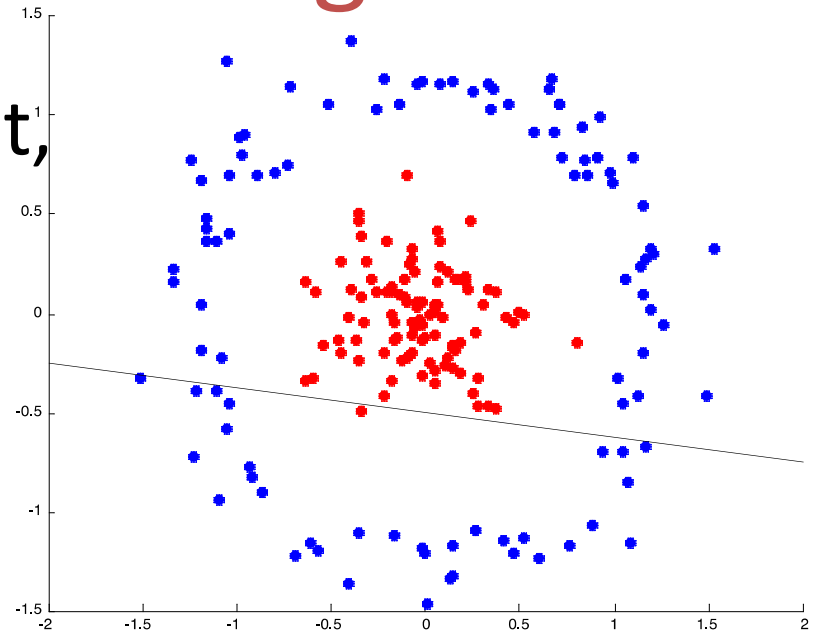
  for some constant C
- Now $\hat{\boldsymbol{w}} = \text{argmin}_w \, J(w)$
- Regularization forces weights to be small, but does it force weights to be exactly *zero*?
  - $w_f = 0$ is equivalent to removing feature f from the model

Kernel Methods (Quick Review)

- Expanding feature space gives us new potentially useful features
- Kernel methods let us work implicitly in a high-dimensional feature space
  - All calculations performed quickly in low-dimensional space

Feature Engineering

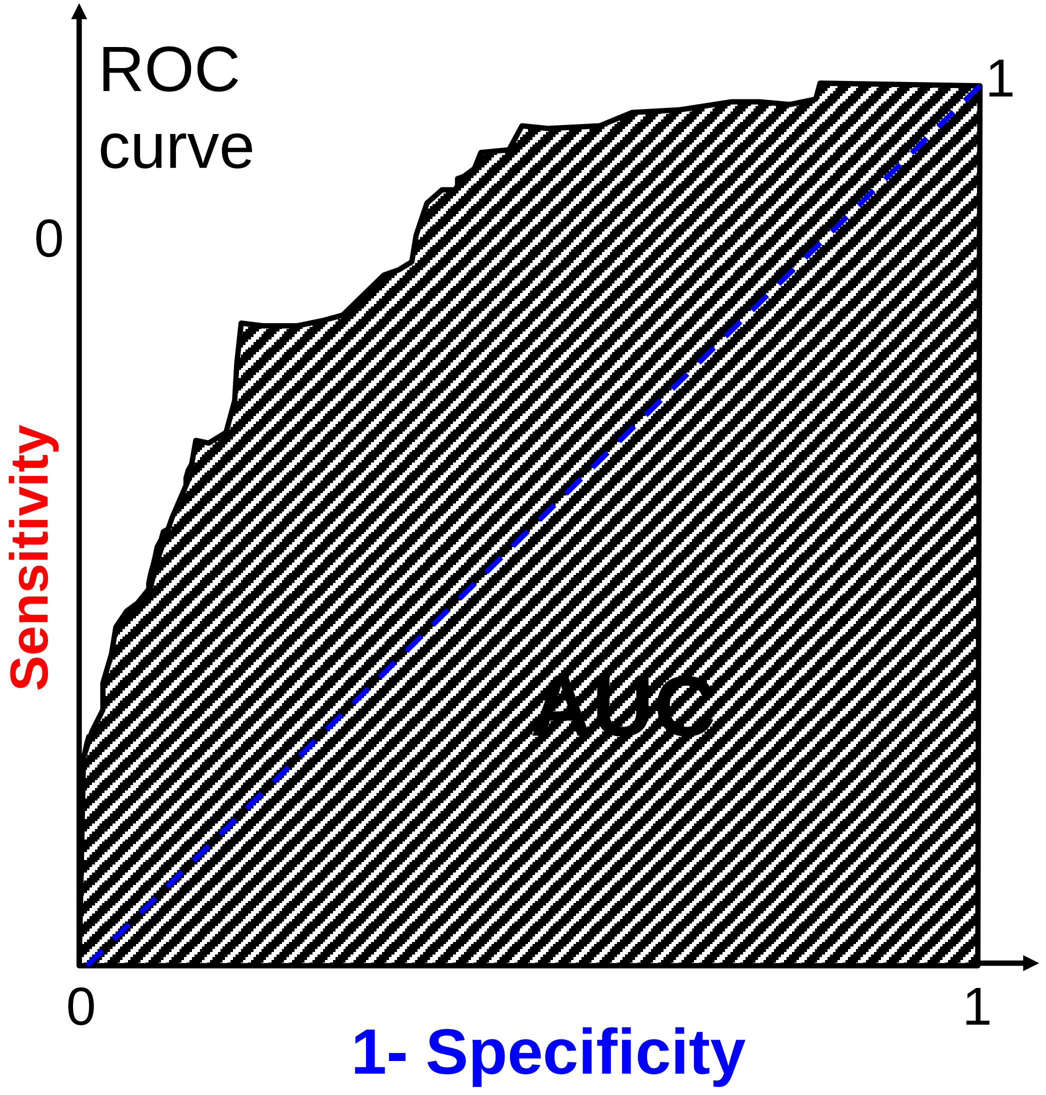
- Linear models: convenient, fairly broad, but limited
- We can increase the expressiveness of linear models by expanding the feature space.

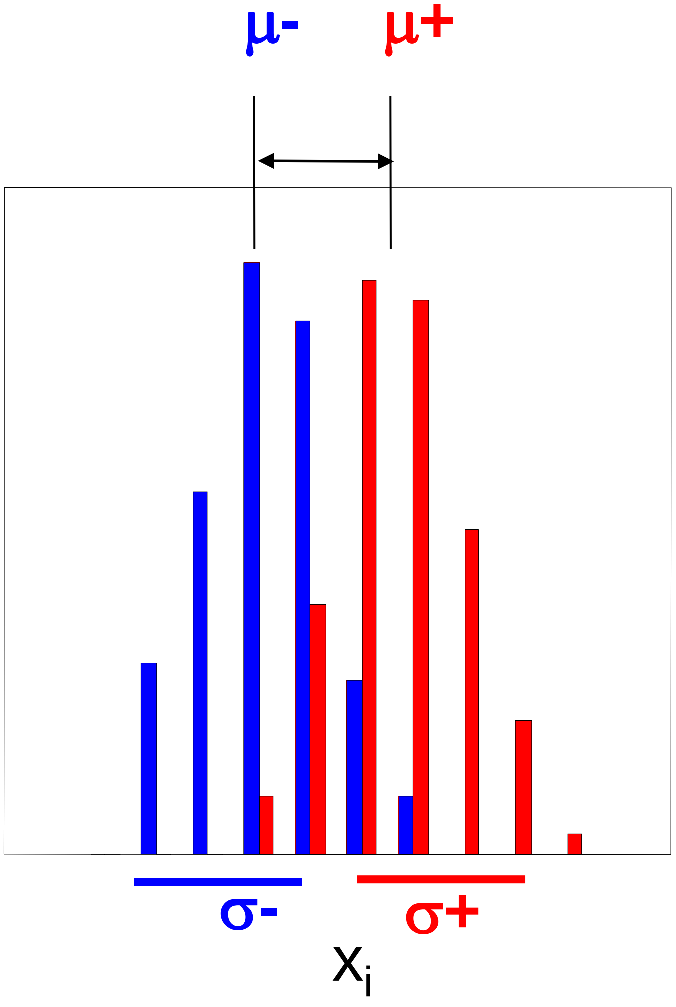


  - E.g. $\Phi([x_1\ x_2]) = [1\ \sqrt{2}x_1\ \sqrt{2}x_2\ \sqrt{2}x_1x_2\ x_1^2\ x_2^2]$
  - Now feature space is R^6 rather than R^2
  - Example *linear* predictor in these features:

$$\mathrm{y} = [1\ 0\ 0\ 0\ \text{-}1\ \text{-}1] \cdot \Phi(\boldsymbol{x}) = 1 - x_1^2 - x_2^2$$

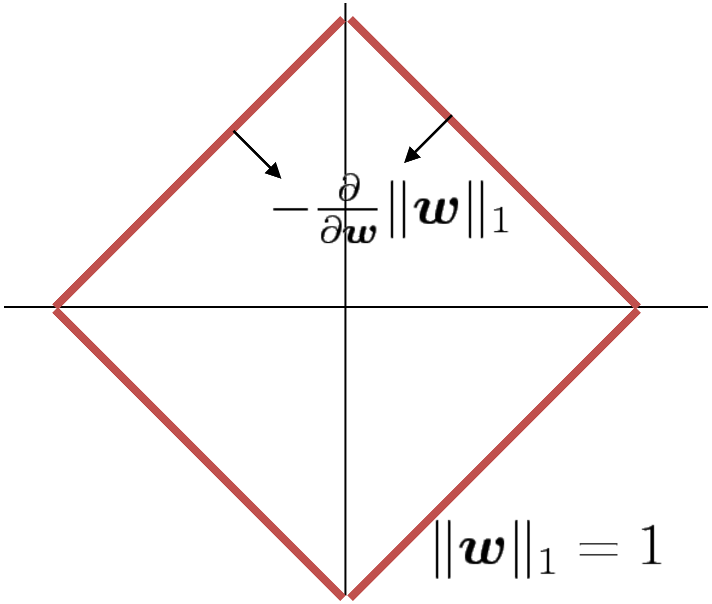
Individual Feature Relevance

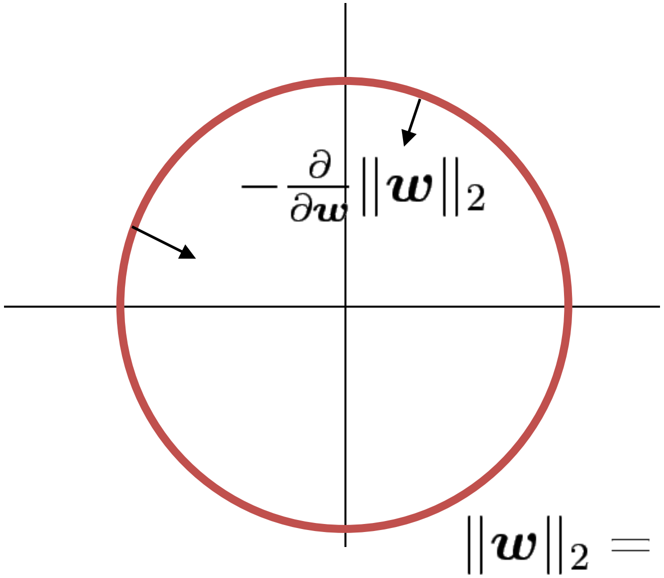




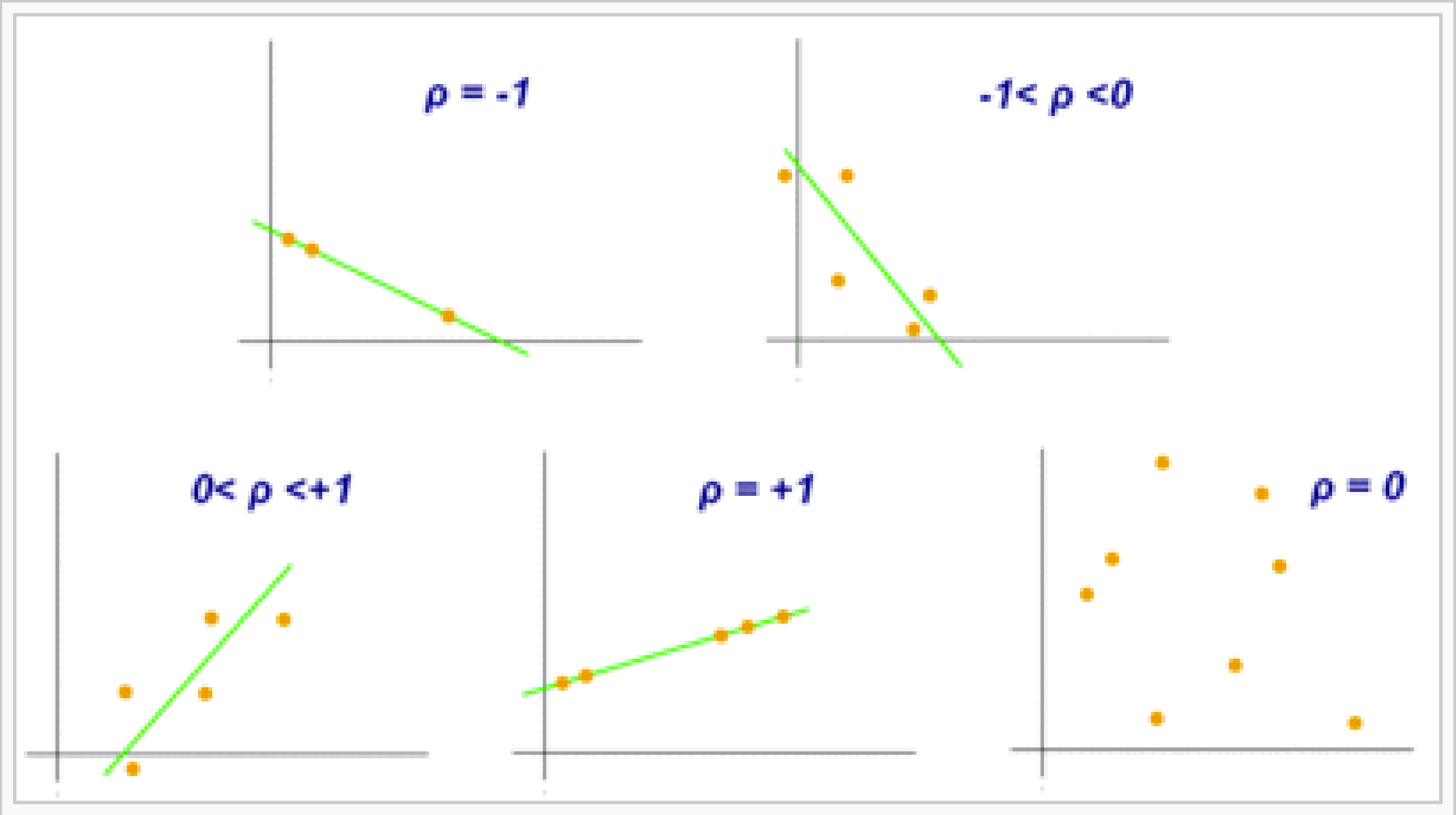
L_1 versus L_2 Regularization

$$\|\boldsymbol{w}\|_1 = \sum_{f=0}^{d} |w_f| \qquad \|\boldsymbol{w}\|_2 = \sqrt{\sum_{f=0}^{d} w_f^2}$$


$-\frac{\partial}{\partial \boldsymbol{w}}\|\boldsymbol{w}\|_1$
$\|\boldsymbol{w}\|_1 = 1$


$-\frac{\partial}{\partial \boldsymbol{w}}\|\boldsymbol{w}\|_2$
$\|\boldsymbol{w}\|_2 = 1$

Pearson correlation coefficient



From Wikipedia

Chi-square test

- We have two random variables:
  - Label (L): 0 or 1
  - Feature (F): Categorical
- Null hypothesis: the two variables are independent of each other (unrelated)
- Under independence
  - P(L=0,F=A)= P(L=0)P(F=A)
  - P(L=0) = (c1+c2)/n
  - P(F=A) = (c1+c3)/n
- Expected values
  - E(X1) = P(L=0)P(F=A)n
- We can calculate the chi-square statistic for a given feature and the probability that it is independent of the label (using the p-value).
- Features with very small probabilities deviate significantly from the independence assumption and therefore considered important.

Contingency table

	Feature=A	Feature=B

Label=0	Observed=c1
	Expected=X1
		Observed=c2
		Expected=X2

Label=1	Observed=c3
	Expected=X3
		Observed=c4
		Expected=X4

$$c^2 = \underset{i}{\text{å}} \frac{(e_i - c_i)^2}{e_i}$$

Multivariate feature selection

- Consider the vector w for any linear classifier.
- Classification of a point x is given by wTx+w0.
- Small entries of w will have little effect on the dot product and therefore those features are less relevant.
- For example if w = (10, .01, -9) then features 0 and 2 are contributing more to the dot product than feature 1. A ranking of features given by this w is 0, 2, 1.

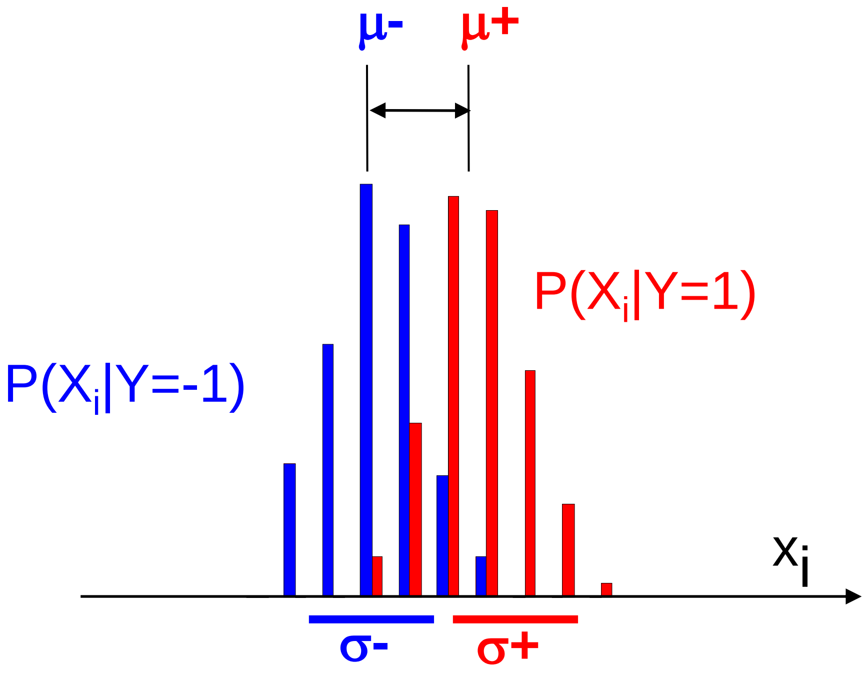
Multivariate feature selection

- The w can be obtained by any of linear classifiers we have see in class so far
- A variant of this approach is called recursive feature elimination:
  - Compute w on all features
  - Remove feature with smallest w_i
  - Recompute w on reduced data
  - If stopping criterion not met then go to step 2

Feature selection in practice

- NIPS 2003 feature selection contest
  - Contest results
  - Reproduced results with feature selection plus SVM
- Effect of feature selection on SVM
- Comprehensive gene selection study comparing feature selection methods
- Ranking genomic causal variants with SVM and chi-square

T-test



- Normally distributed classes, equal variance σ^2 unknown; estimated from data as σ^2_{within}.
- Null hypothesis H_0: $\mu+ = \mu-$
- T statistic: If H_0 is true,

$$t = (\mu+ - \mu-)/(\sigma_{within}\sqrt{1/m^+ + 1/m^-}) \sim \text{Student}(m^+ + m^- - 2 \text{ d.f.})$$