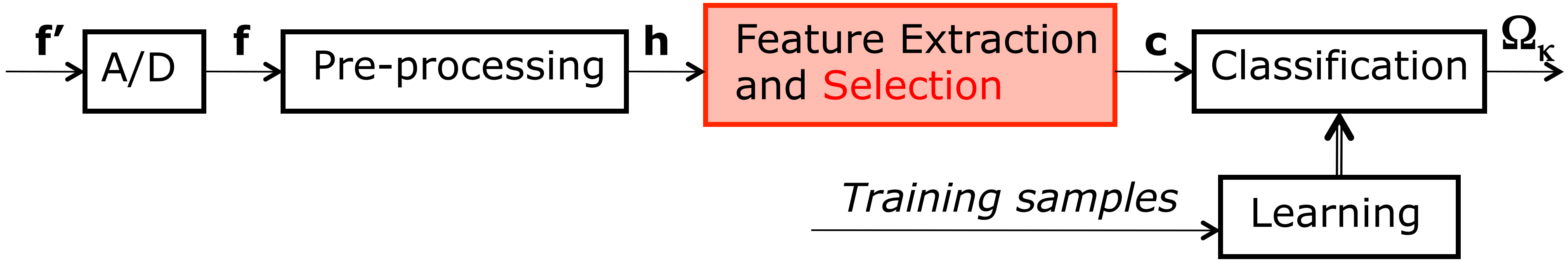
Feature Selection

Dr. Elli Angelopoulou

Lehrstuhl für Mustererkennung (Informatik 5)

Friedrich-Alexander-Universität Erlangen-Nürnberg

Pattern Recognition Pipeline



- Feature Extraction
  - Heuristic feature extraction methods
  - Analytic feature extraction methods
- Feature Selection

Why Feature Selection?

- There are many methods for feature extraction (e.g. Walsh/Hadamard Transform, moments, PCA, etc.).
- We often apply more than one set of feature extraction methods, e.g. Fourier Transform and moments on color data.
- More data is better data.
- Curse of dimensionality.
- So what should one do?
- Use fewer but highly discriminating features. In other words lower the dimensionality of the (combined) feature vector $\vec{c}$.

New Feature Vector

- However, we need a systematic way of selecting which features, (i.e. which elements of the feature vector) will be used in the next step of the pattern recognition pipeline, i.e. in the classification.
- We started with a signal vector of dimensionality N.

$$\vec{f} \in R^N$$

- Through feature extraction we created a new vector $\vec{c}$ that emphasizes the information within the input signal that is characteristic for the pattern recognition problem at hand.

$$\vec{c} \in R^M \quad \text{, where } M < N$$

- Through feature selection we want to create a more compact, yet **at least equally discriminating** vector

$$\vec{c}\,' \in R^{M'} \quad \text{, where } M' < M$$

$$\{c'_1, c'_2, \ldots c'_{M'}\} \subset \{c_1, c_2, \ldots c_M\}$$

Feature Selection

- Feature selection: Given M features, find the optimal subset with M', $M' < M$, features which minimizes the probability for misclassification.
- The best subset of M' features has the property, that there exists no subset with $K,\ K \leq M'$ features which yields a smaller probability of misclassification.
- Generally, it is impossible to find the best subset of features without examining all possible subsets.
- For a lower computational cost, we need to compromise and search for a suboptimal set.

Feature Selection Algorithms

- There are many different feature selection methods.
- Algorithms for feature selection are characterized by:

1. The **objective function** (a.k.a. criterion function) used in evaluating the “goodness” of a subset.
2. The **optimization method** used in searching the space of possible subsets for the best subset.

- Since the search space is composed by the different subsets, we have a discrete search space, which means that we have a *discrete optimization problem*.

Objective Function Properties

- The objective function used in feature extraction should satisfy the following requirements:

1. It should be simple, in terms of computational efficiency, to evaluate.
2. It should properly approximate the misclassification error.
3. It should avoid complete testing of all possible subsets.

   Example: Consider the case where M=300, M'=30.

   An exhaustive search would involve the evaluation of:

$$\binom{M}{M'} = \binom{300}{30} = \frac{300!}{30!270!} \approx 10^{41} \quad \text{subsets}$$

Some Objective Functions

- There are four widely-used good objective functions which closely approximate the misclassification error:

1. Error-rate:

$$p_f = \frac{\#\ \text{of misclassifications}}{\#\ \text{of classified samples}}$$

The goal is to minimize the error rate p_f.

2. Bayesian Distance:

$$B = \int_{R_{\vec{c}}} \sum_{\kappa=1}^{K} p(\vec{c})\, p\left(\Omega_\kappa \middle| \vec{c}\right)^2 d\vec{c}$$

where $R_{\vec{c}}$ is the space of all possible feature vectors. Large *B* values denote that on average one can safely classify the sample. So we try to find the new feature vector that maximizes *B*.

A Review of Entropy

3. Conditional Entropy:

- The entropy, *H*(*X*), of a discrete random variable *X* is a measure of the amount of uncertainty associated with the value of *X*.

$$H(X) = -\int_{-\infty}^{\infty} p(x)\log(p(x))dx$$

- The conditional entropy, *H*(*X*|*Y*), of a discrete random variable *X*, given another random variable *Y*, is a measure of the amount of uncertainty regarding the value of *X* that remains after we already know the value of a 2nd random variable *Y*.

A Review of Entropy - continued

- For a particular value *y* of *Y*, the conditional entropy is defined as:

$$H(X|y) = -\int_{-\infty}^{\infty} p(x|y)\log\big(p(x|y)\big)dx$$

- For *any* possible value of *Y*, we just "average" over all possible *Y* values:

$$H(X|Y) = -\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p(y)p(x|y)\log\big(p(x|y)\big)dxdy$$

- In a pattern recognition system of K classes, the conditional entropy of class Ω_κ given a feature vector $\vec{c}$ is:

$$H(\Omega_\kappa|\vec{c}) = -\int_{R_{\vec{c}}}\sum_{\kappa=1}^{K} p(\vec{c})p(\Omega_\kappa|\vec{c})\log\big(p(\Omega_\kappa|\vec{c})\big)d\vec{c}$$

Entropy-Based Objective Function

- This conditional entropy:

$$H\left(\Omega_\kappa \middle| \vec{c}\right) = -\int_{R_{\vec{c}}} \sum_{\kappa=1}^{K} p(\vec{c})\, p\left(\Omega_\kappa \middle| \vec{c}\right) \log\left(p\left(\Omega_\kappa \middle| \vec{c}\right)\right) d\vec{c}$$

  can be used as an objective function for feature selection.

- The conditional entropy is a measure of the amount of uncertainty associated with belonging to class Ω_κ, given that we have already observed the feature vector $\vec{c}$.
- We want to choose a feature vector that minimizes the uncertainty in classification.
- We want to minimize this uncertainty, minimize *H*().

A Review of Mutual Information

4. Mutual Information:

- Mutual information, $I(X,Y)$, is a measure of “information similarity" between 2 random variables X and Y.
- It measures the amount of information that can be obtained about one random variable, X, by observing another random variable, Y.
- It measures the amount of information about X that is shared with Y.
- It is closely related to entropy $H()$.

A Review of Mutual Information - continued

- Typically mutual information *I*(*X*,*Y*) is defined as:

$$MI = I(X,Y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p(x,y)\log\left(\frac{p(x,y)}{p_1(x)p_2(y)}\right)dxdy$$

- It is also often defined in terms of entropy:

$$MI = I(X,Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$$

- Lastly, it can also be expressed in terms of the Kullback-Leibler divergence. The Kullback-Leibler-Divergence (KL-divergence) is a measurement of the similarity of two probability distributions, *p*(X) and *q*(X):

$$MI = KL(p,q) = \int_{-\infty}^{\infty} p(x)\log\left(\frac{p(x)}{q(x)}\right)dx$$

KL Divergence – Based Objective Function

- Consider the probability distribution function of a particular feature vector $\vec{c}$ belonging to a specific class Ω_κ.
- If the feature vector $\vec{c}$ and the class Ω_κ were completely unrelated, i.e. they were statistically independent variables, then their joint probability distribution $p(\vec{c},\Omega_\kappa)$ would be:

$$p(\vec{c},\Omega_\kappa) = p(\vec{c})\,p(\Omega_\kappa)$$

- But that means that we get no information about Ω_κ by observing $\vec{c}$.

KL Divergence – Based Objective Function

- Thus, we want to make $p(\vec{c},\Omega_\kappa)$ and $p(\vec{c})p(\Omega_\kappa)$ as different as possible.
- We want to have the probability distribution of $p(\vec{c},\Omega_\kappa)$ and of $p(\vec{c})p(\Omega_\kappa)$ as dissimilar a possible.
- In other words, their Kullback-Leibler divergence

$$KL = \int_{R_{\vec{c}}} \sum_{\kappa=1}^{K} p(\vec{c},\Omega_\kappa) \log\left(\frac{p(\vec{c},\Omega_\kappa)}{p(\vec{c})p(\Omega_\kappa)}\right) d\vec{c}$$

  should be maximized.
- Note: The Kullback-Leibler divergence is not a true metric since it is not symmetric and does not satisfy the triangle inequality.

Summary of the Four Objective Functions

- There are four widely used objective functions:
  1. Error-rate (minimize)
  2. Bayesian distance (maximize)
  3. Conditional entropy (minimize)
  4. Mutual information (maximize)
- All of these evaluation functions have the following advantages:
  1. Very good approximations for the probability of misclassification,
  2. Can be used for K-class problems,
  3. Have good theoretical foundations
- Common limitation: they are typically quite hard to evaluate.

Comparison of the Objective Functions

- Let p_{opt} be the error probability of the *optimal classifier*.
- It is a theoretical performance measure.
- It has been proven that:

$$p_{opt} \leq (1 - B) \leq \frac{1}{2} H\left(\Omega_{\kappa} \middle| \vec{c}\right)$$

- So we are getting closer to the optimal classifier if we use the Bayesian distance than if we use the conditional entropy, but the conditional entropy is somewhat more efficient to compute.

Specialization in Feature Selection

- When we have prior knowledge, it is quite common to do some kind of specialization, which typically means that we can assume some type of parametric distribution of features.
- For instance, we can assume that features are normally distributed (this is , for example, the case for features created via PCA).
- Then the statistics of the feature vector are captured by the mean and the covariance matrix.
- If we know that features are normally distributed we can take advantage of special properties of normal distributions in doing our feature selection.

Specialization in Feature Selection - cont

- Consider for example a vector $\vec{u}^T = (x,y,z)$ which is normally distributed

$$\vec{u} \approx \mathcal{N}(\vec{u}, \vec{\mu}_u, \Sigma_u)$$

  where $\vec{\mu}_u$ is its mean vector and Σ_u its covariance.
- Assume that out of this vector we only want to select elements *x* and *y*.
- What is the probability distribution of the resulting 2-elt vector (*x*,*y*) if we know that $\vec{u}$ is normally distributed? What is $p(x,y)$?
- We know that $p(x,y,z)$ is a normal distribution. How does it change when we remove z?

Specialization in Feature Selection - cont

- We can select only feature elements x and y as follows:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix}$$

$$A \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix}$$

- Then the marginal density of (*x*, *y*) is:

$$p(x,y) = \int_{-\infty}^{\infty} \mathscr{N}(\vec{u},\vec{\mu}_u,\Sigma_u)dz = \mathscr{N}\left(\begin{bmatrix} x \\ y \end{bmatrix}, A\vec{\mu}_u, A\Sigma_u A^T\right)$$

(special property of normal distributions is that their marginals also follow the normal distribution)

Feature Selection Methods

- As previously stated, algorithms for feature selection are characterized by:

1. The **objective function** (a.k.a. criterion function) used in evaluating the "goodness" of a subset.
2. The **optimization method** used in searching the space of possible subsets for the best subset.

- Once an objective function is chosen, one has to systematically examine the different subsets of a feature vector.
- Each subset is evaluated, using the chosen objective function.

Different Feature Selection Methods

1. Random Selection.

   Use a random selection of features to include in $\vec{c}'$.

   On average, this does not result in good feature vectors.

2. Exhaustive Search.

   Select the M' features that give the best objective function values.

   May be difficult to compute.

   Choosing the features that give the best objective function values may not, after all, be the features that give the best classification results.

Different Feature Selection Methods - cont

3. Greedy `(gierig).`

   Choose one feature at at time. Select the feature that best fits to the already existing ones, i.e. leads to the highest increase (or decrease) of the objective function value.

4. Hardest Pair.

   Add the feature that contributes the most to the separation of the hardest class pair.

   Or equivalently eliminate the feature with the smallest contribution to the separation of the hardest pair.

Different Feature Selection Methods - cont

5. (l,r)-search.

   Add the l strongest features while eliminating the r weakest ones.

   Different variations: fix r and l , or keep one of them or both of them dynamic.

6. Branch and Bound.

   Key idea: Use a monotonic objective function (e.g. mutual information) so that instead of rejecting a single feature c_i due to its low contribution, one can eliminate a set of features that all give lower contributions than c_i .

Branch and Bound

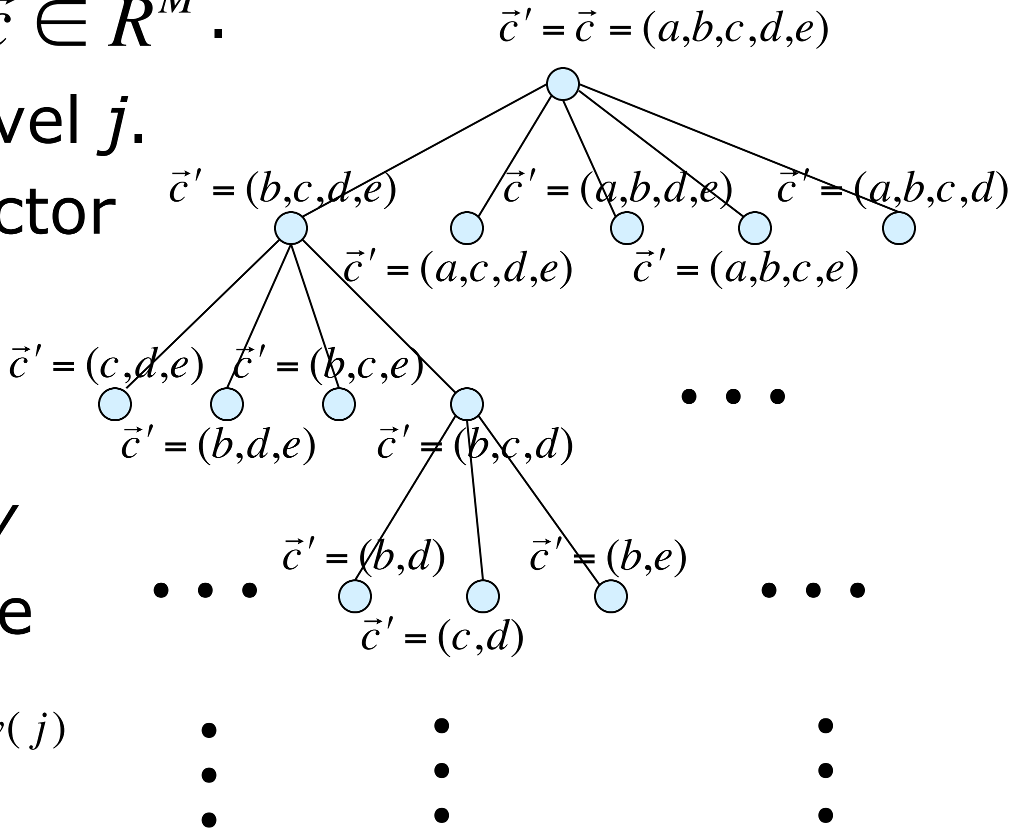
- The Branch and Bound algorithm is guaranteed to find the optimal feature subset under the **monotonicity assumption**.
- The monotonicity assumption states that the addition of features can only increase the value of the objective function (assuming a maximizing objective function):

$$G(c_{i_1}) < G(c_{i_1}, c_{i_2}) < G(c_{i_1}, c_{i_2}, c_{i_3}) < \cdots < G(c_{i_1}, c_{i_2}, \ldots, c_{i_{M'}})$$

- Branch and Bound typically starts from the full set and removes features using a depth-first strategy.

Branch and Bound - continued

- The search space is represented by a tree.
- The root node (at level j=0) corresponds to the entire feature set $\vec{c}\,' = \vec{c} \in R^M$.
- Consider a node *v* at level *j*. It includes a feature vector of dimensionality *M-j*,
$$\vec{c}\,'_{v(j)} \in R^{M-j}$$
- Each child of the node *v* corresponds to a feature vector obtained from $\vec{c}\,'_{v(j)}$ by removing a different element from $\vec{c}\,'_{v(j)}$.



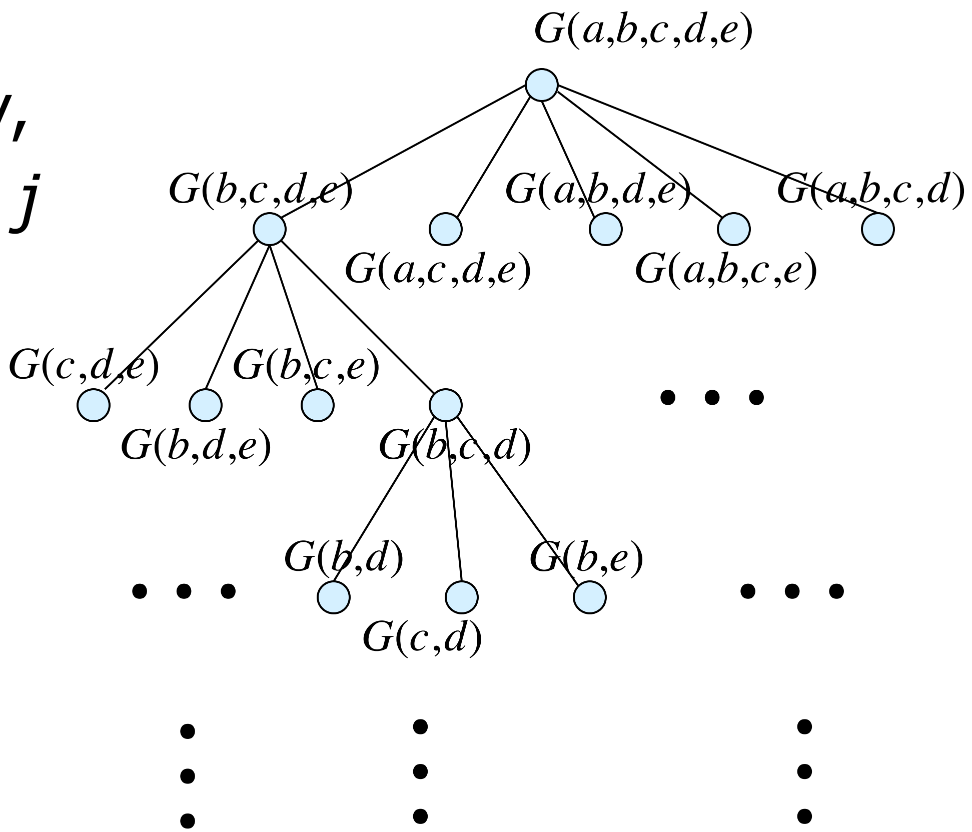
Branch and Bound - continued

- Recall that Branch and Bound assumes that the objective function is monotonic:

$$G(c_{i_1}) < G(c_{i_1}, c_{i_2}) < G(c_{i_1}, c_{i_2}, c_{i_3}) < \cdots < G(c_{i_1}, c_{i_2}, \ldots, c_{i_{M'}})$$

- Due to this monotonicity, the parent node at level *j* has a higher objective function value than its children at level *j+1*.

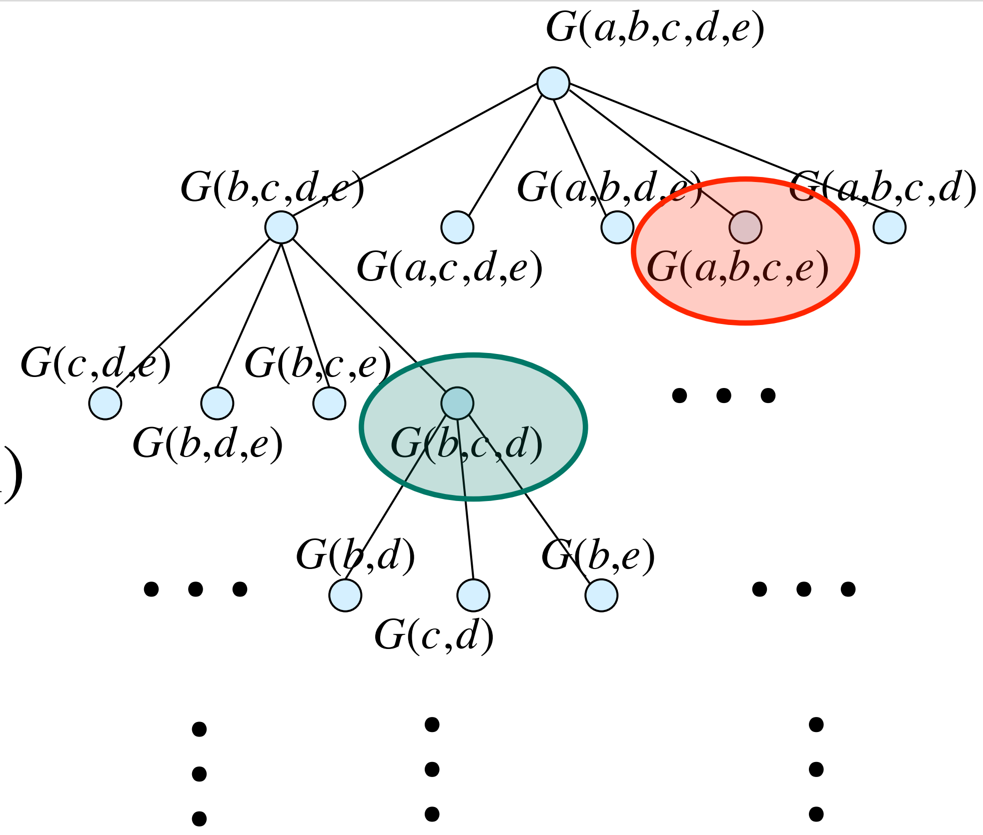
$$G(v) > G(\text{child}(v))$$



Branch and Bound - continued

- Keep in mind that:
  - $G(v) > G(\text{child}(v))$
  - Goal: maximize $G()$
- Assume that at level j+1 there exist a node $v_k(j+1)$ whose objective function value is higher than that of a node $v_m(j)$ at the previous level j:

$$G(v_k(j+1)) > G(v_m(j))$$



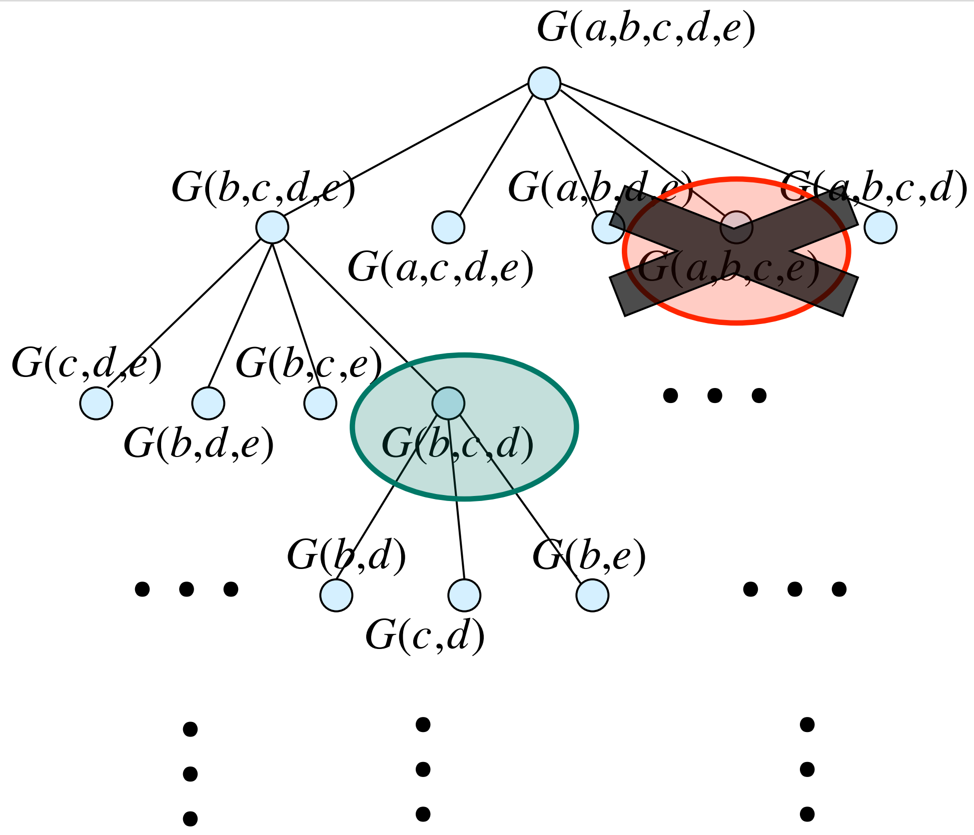
Branch and Bound - continued

- Facts:

$$G(v) > G(\text{child}(v))$$

$$G(v_k(j+1)) > G(v_m(j))$$

- Due to the monotonicity, of the objective function all descendents of $v_m(j)$ will have objective function values lower than $v_k(j+1)$.



$$G(v_k(j+1)) > G(v_m(j)) > G(\text{descendant}(v_m(j)))$$

- The subtree rooted at the node $v_m(j)$ can be pruned.