Combining Near-Optimal Feature Selection with *gSpan*

Marisa Thoma[1]

joint work with

Karsten Borgwardt[2], Xifeng Yan[3], Hong Cheng[6], Arthur Gretton[4], Le Song[5], Alex Smola[5], Jiawei Han[6], Philip Yu[2], Hans-Peter Kriegel[1]

[1] LMU Munich, [2] University of Cambridge, [3] IBM Watson Research Center New York, [4] MPI Tübingen, [5] NICTA Canberra, [6] UIUC

Overview

- Why feature selection?
- Submodular selection criteria
- CORK, a novel feature selection criterion
- Inclusion of CORK into *gSpan*
- Experimental validation
- Summary and outlook

Used setting

- Two-class problem:
  - Collection of graphs
  - Predict class label from graph topology
- Current solutions:
  - Pattern-based learning (subgraphs, paths, circles)
  - Graph kernel approaches (random walk, pattern based approaches)
  - Nested approaches (feature generation adapted to the dataset)

Feature Properties

- Frequent Subgraphs
  - Can be efficiently enumerated (*gSpan*, FSG, MoSS)
- They also contain
  - Insignificant features
  - Redundant features, already covered by other substructures
- Potentially exponentially many

Feature Selection

- Identify the most discriminative subgraphs

- Set $\mathcal{T}_{opt} \subseteq \mathcal{F}$ (total feature set), s.t.

$$\mathcal{T}_{opt} = \underset{\mathcal{S} \subseteq \mathcal{F}}{\text{argmax}}\ f(\mathcal{S})$$

  according to information criterion *f*.

Feature Selection

Approaches

- Complete subset enumeration and test (Wrapper)
- Ranker selection (assuming feature independence)
- Greedy approaches
  - Backward Elimination vs. Forward Selection
- Nested approaches
  - Feature selection performed in combination with feature generation

Submodularity

Proof by Krause and Guestrin

A note on the Budgeted Maximization of Submodular Functions. CMU 2005

Submodular decision function

Greedy Forward Selection

Guarantee of

$$f(\widehat{\mathcal{F}}) \geq \left(1 - \frac{1}{e}\right) f(\mathcal{F}_{opt})$$

Submodularity

Proof by Krause and Guestrin

A note on the Budgeted Maximization of Submodular Functions. CMU 2005

Submodular decision function

Greedy Forward Selection

Guarantee of

$$f(\widehat{\mathcal{F}}) \geq \left(1 - \frac{1}{e}\right) f(\mathcal{F}_{opt})$$

set function $f: \mathcal{F} \rightarrow \mathbb{R}$

f is *submodular* $\Leftrightarrow$

$\mathcal{S} \subset \mathcal{T} \subseteq \mathcal{F}$, $s \in \mathcal{F}$:

$$f(\mathcal{S} \cup \{s\}) - f(\mathcal{S}) \geq f(\mathcal{T} \cup \{s\}) - f(\mathcal{T})$$

Submodularity

Proof by Krause and Guestrin

A note on the Budgeted Maximization of Submodular Functions. CMU 2005

Submodular decision function

Greedy Forward Selection

Guarantee of

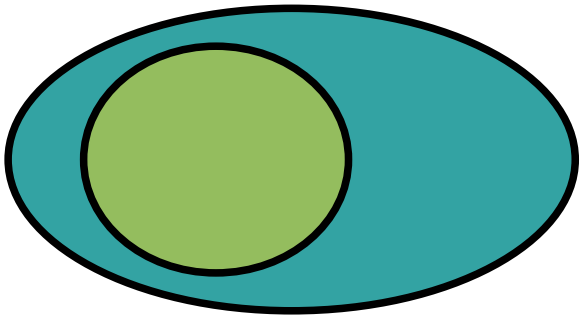
$$f(\widehat{\mathcal{T}}) \geq \left(1 - \frac{1}{e}\right) f(\mathcal{T}_{opt})$$

set function $f: \mathcal{F} \rightarrow \mathbb{R}$

f is *submodular* $\Leftrightarrow$

$$\mathcal{S} \subset \mathcal{T} \subseteq \mathcal{F}, s \in \mathcal{F} :$$

$$f(\mathcal{S} \cup \{s\}) - f(\mathcal{S}) \geq f(\mathcal{T} \cup \{s\}) - f(\mathcal{T})$$



$f(S) = area(S)$

Submodularity

Proof by Krause and Guestrin

A note on the Budgeted Maximization of Submodular Functions. CMU 2005

Submodular decision function

Greedy Forward Selection

Guarantee of

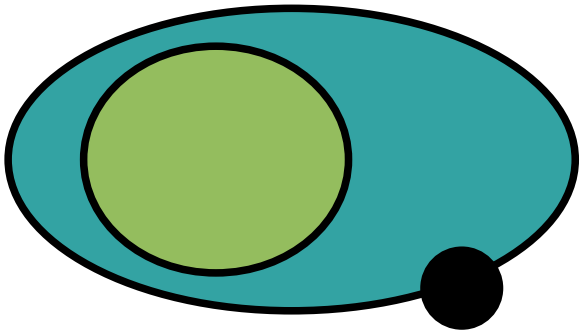
$$f(\widehat{\mathcal{T}}) \geq \left(1 - \frac{1}{e}\right) f(\mathcal{T}_{opt})$$

set function $f: \mathcal{F} \rightarrow \mathbb{R}$

f is *submodular* $\Leftrightarrow$

$$\mathcal{S} \subset \mathcal{T} \subseteq \mathcal{F}, s \in \mathcal{F} :$$

$$f(\mathcal{S} \cup \{s\}) - f(\mathcal{S}) \geq f(\mathcal{T} \cup \{s\}) - f(\mathcal{T})$$



$f(S) = area(S)$

CORK

Correspondence:

A pair of instances $i, j \in$ Dataset D with class(i) $\neq$ class(j) is called a correspondence in a feature set $\mathcal{S}$, if i and j have the same values for $\mathcal{S}$.

Correspondence-based Quality Criterion (*CORK*):

$$q(\mathcal{S}) = (-1) * \text{“}\textit{number of correspondences in } \mathcal{S}\text{”}$$

From now:

use CORK for binary feature values {0 , 1}.

CORK for One Feature *S*

Correspondence-based Quality Criterion (*CORK*):

$$q(S) = (-1) * \text{“}number\ of\ correspondences\ in\ S\text{”}$$

A_{S_0}: # “feature $S = 0$ in class A“ (# mis-matches)

A_{S_1}: # “feature $S = 1$ in class A“ (# matches)

$$q(\{S\}) = -\left(A_{S_0} \cdot B_{S_0} + A_{S_1} \cdot B_{S_1}\right)$$

2-class dataset

Features		A			B		
		a_1	a_2	a_3	b_1	b_2	b_3
	s_1	1	1	0	1	0	0
	s_2	1	0	0	1	0	0

CORK for One Feature *S*

Correspondence-based Quality Criterion (*CORK*):

$$q(S) = (-1) * \text{“}number\ of\ correspondences\ in\ S\text{”}$$

A_{S_0}: # “feature $S = 0$ in class A“ (# mis-matches)

A_{S_1}: # “feature $S = 1$ in class A“ (# matches)

$$q(\{S\}) = -\left(A_{S_0} \cdot B_{S_0} + A_{S_1} \cdot B_{S_1}\right)$$

$$q(\{s_1\}) = -(1 \cdot 2 + 2 \cdot 1) = -4$$

2-class dataset

Features		A			B		
		a_1	a_2	a_3	b_1	b_2	b_3
	s_1	1	1	0	1	0	0
	s_2	1	0	0	1	0	0

CORK for One Feature *S*

Correspondence-based Quality Criterion (*CORK*):

$$q(S) = (-1) * \text{“}number\ of\ correspondences\ in\ S\text{”}$$

A_{S_0}: # “feature $S = 0$ in class A“ (# mis-matches)

A_{S_1}: # “feature $S = 1$ in class A“ (# matches)

$$q(\{S\}) = -\left(A_{S_0} \cdot B_{S_0} + A_{S_1} \cdot B_{S_1}\right)$$

$$q(\{s_1\}) = -(1 \cdot 2 + 2 \cdot 1) = -4$$

$$q(\{s_2\}) = -(2 \cdot 2 + 1 \cdot 1) = -5$$

2-class dataset

Features	A			B		
	a_1	a_2	a_3	b_1	b_2	b_3
s_1	1	1	0	1	0	0
s_2	1	0	0	1	0	0

CORK on Feature Sets

$$q(\{S\}) = -\left(A_{S_0} \cdot B_{S_0} + A_{S_1} \cdot B_{S_1}\right)$$

$$q(\{S, T\}) = -\left(\sum_{i,j=0}^{1} A_{S_i,T_j} \cdot B_{S_i,T_j}\right)$$

	A			B		
	a_1	a_2	a_3	b_1	b_2	b_3
s_1	1	1	0	1	0	0
s_2	1	0	0	1	0	0

Histogram over *equivalence classes* for the possible feature combinations

For M features: 2^M possible equivalence classes

$$q(\{s_1, s_2\}) = -\left(\overset{0,0}{1 \cdot 2} + \overset{0,1}{0 \cdot 0} + \overset{1,0}{1 \cdot 0} + \overset{1,1}{1 \cdot 1}\right) = -3$$

CORK is Submodular

$\mathcal{S} \subset \mathcal{T} \subseteq \mathcal{F}$, $s \in \mathcal{F}$:

Improvement of $\mathcal{T}$ ⇒ must also improve $\mathcal{S}$

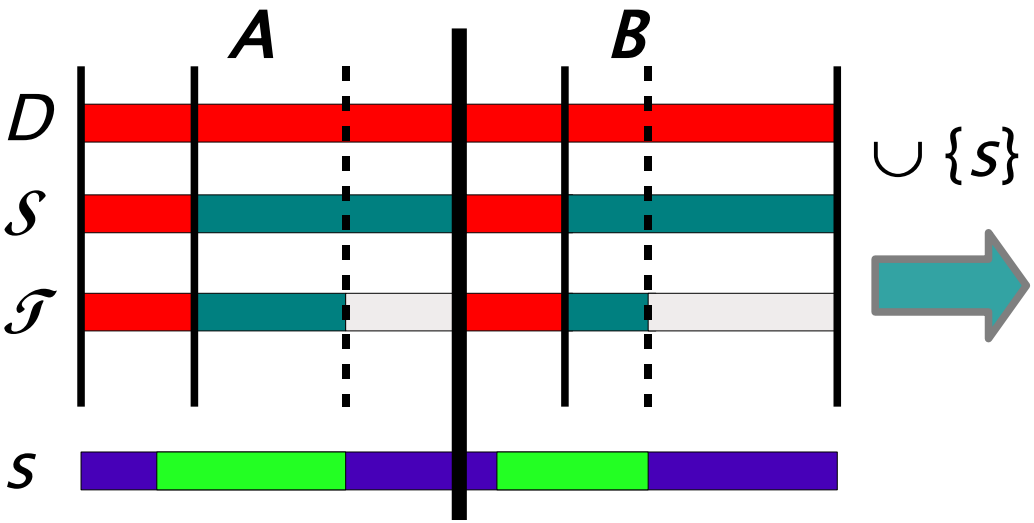
Improvement $\triangleq$ remove a correspondence

CORK is Submodular

$\mathcal{S} \subset \mathcal{T} \subseteq \mathcal{F}$, $s \in \mathcal{F}$:

Improvement of $\mathcal{T}$ ⇒ must also improve $\mathcal{S}$

Improvement $\triangleq$ remove a correspondence



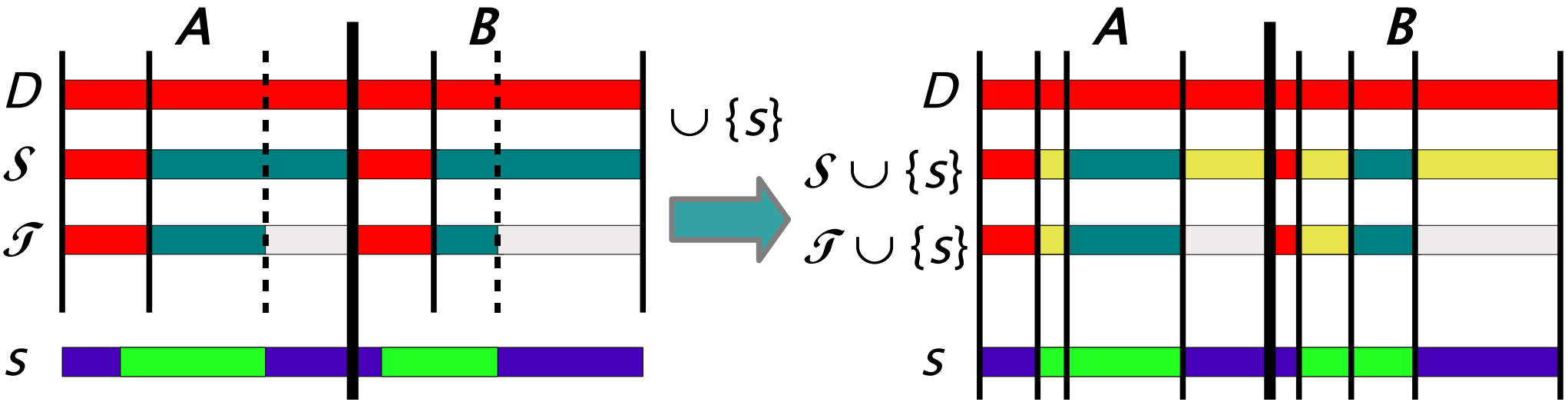
$$f(\mathcal{S} \cup \{s\}) - f(\mathcal{S}) \geq f(\mathcal{T} \cup \{s\}) - f(\mathcal{T})$$

CORK is Submodular

$\mathcal{S} \subset \mathcal{T} \subseteq \mathcal{F}$, $s \in \mathcal{F}$:

Improvement of $\mathcal{T}$ ⇒ must also improve $\mathcal{S}$

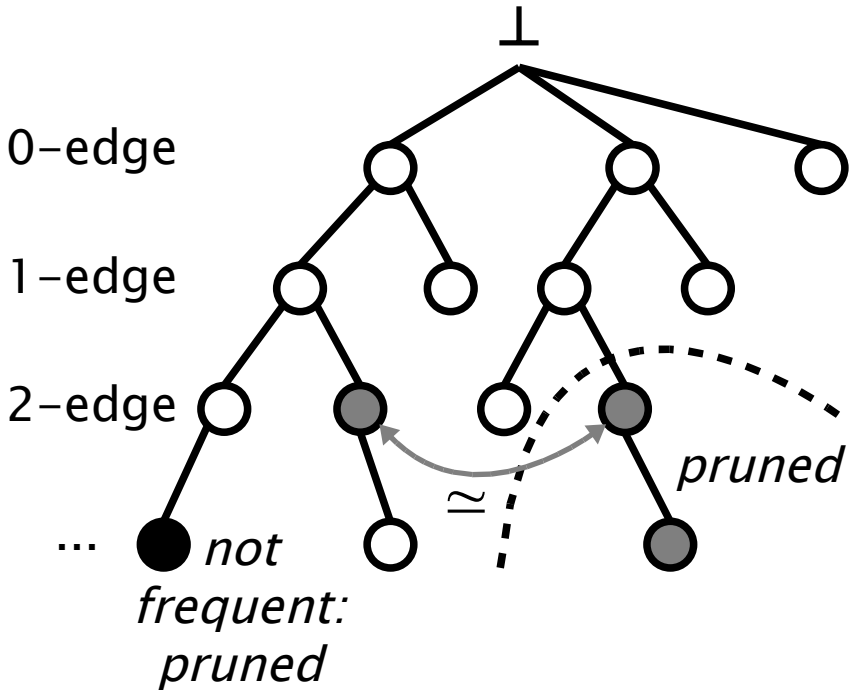
Improvement $\triangleq$ remove a correspondence



$$f(\mathcal{S} \cup \{s\}) - f(\mathcal{S}) \geq f(\mathcal{T} \cup \{s\}) - f(\mathcal{T})$$

Primer on *gSpan*

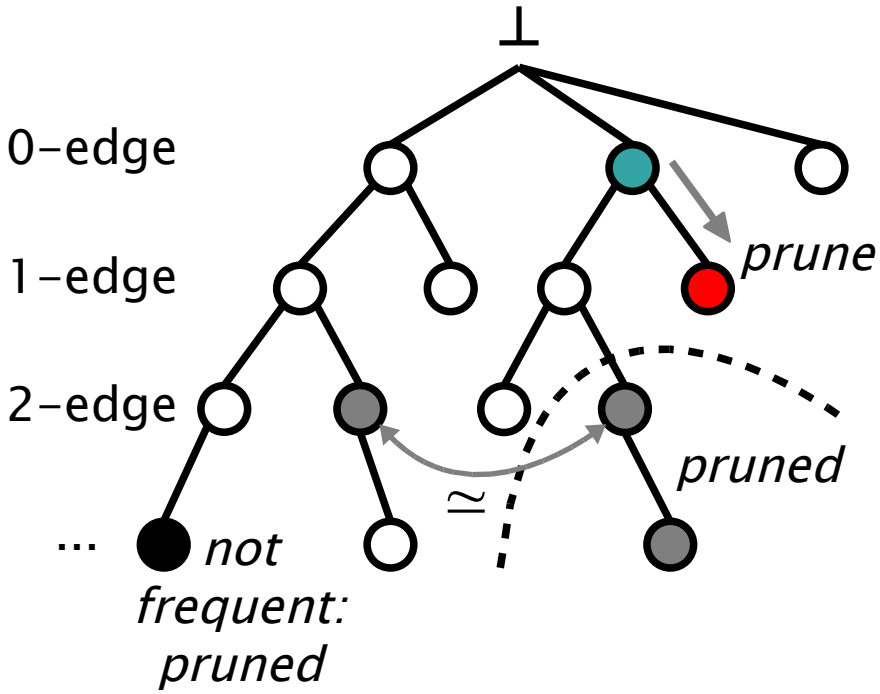
Yan, X. & Han, J.: *gSpan*: Graph-based sub-structure pattern mining. *Proc. ICDM* 2002



- DFS Code Tree
- Efficient branch-and-bound
  - Frequency bound
  - Minimality bound of DFS Code
- Worst-case runtime: exponential

- For any parent-child relationship in the search tree: *parent* $\sqsubseteq$ *child*

CORK in *gSpan*

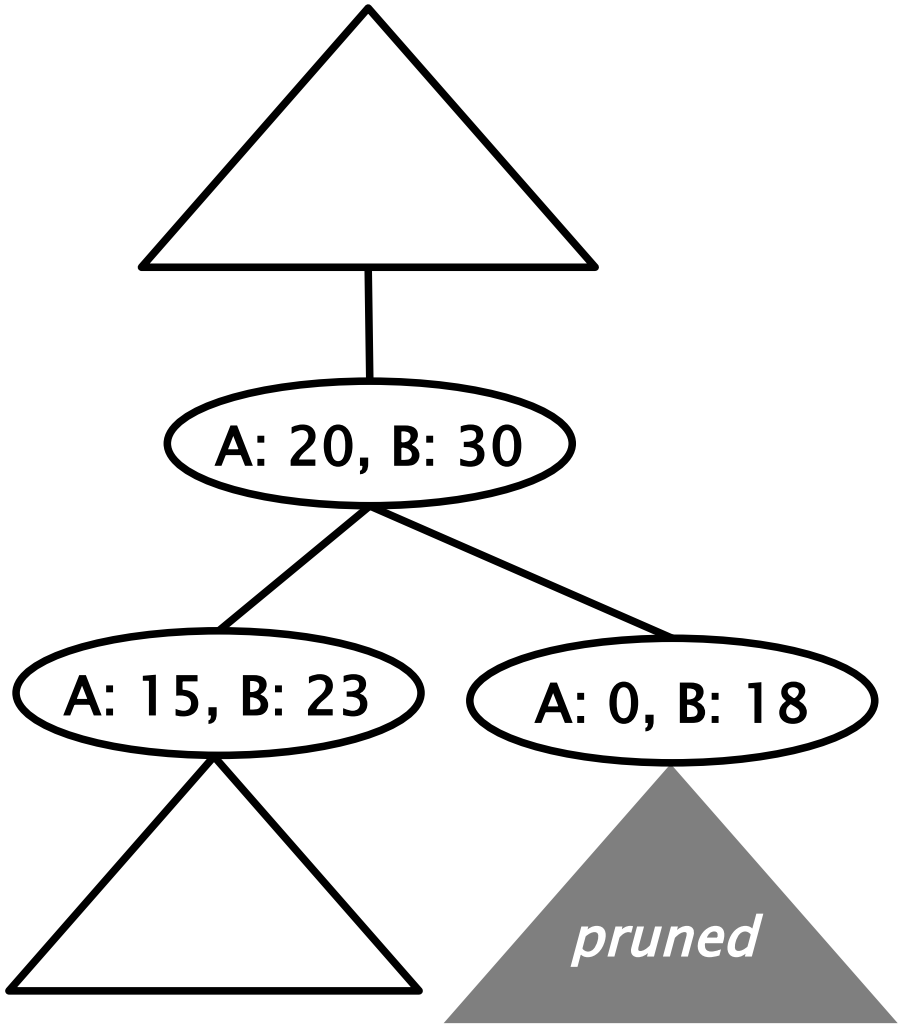


- Let *parent* $S \sqsubseteq$ *child* T be frequent subgraphs
- From S to T, we can only loose matching subgraph embeddings but never gain additional matches. $\left(A_{S_1} \geq A_{T_1} \wedge B_{S_1} \geq B_{T_1}\right)$

CORK: $q(\{S\}) = -\left(A_{S_0} \cdot B_{S_0} + A_{S_1} \cdot B_{S_1}\right)$

- The maximal value of $q(\{T\})$ is reached, when either all matches in A or all matches in B are lost.

CORK in *gSpan*



- Let *parent* $S \sqsubseteq$ *child* T be frequent subgraphs
- From S to T, we can only loose matching subgraph embeddings but never gain additional matches. $\left(A_{S_1} \geq A_{T_1} \wedge B_{S_1} \geq B_{T_1}\right)$

CORK: $q(\{S\}) = -\left(A_{S_0} \cdot B_{S_0} + A_{S_1} \cdot B_{S_1}\right)$

- The maximal value of $q(\{T\})$ is reached, when either all matches in A or all matches in B are lost.

CORK as a Threshold

For $S \sqsubseteq T$ we can derive:

$$q(\{T\}) \leq q(\{S\}) + \max \begin{cases} (A_{S_1} - A_{S_0}) \cdot B_{S_1} \\ A_{S_1} \cdot (B_{S_1} - B_{S_0}) \\ 0 \end{cases}$$

Enables earlier pruning:

- No supergraph of T can exceed this upper bound.
- Thus, if we have seen a better subgraph, this branch can be pruned.

gSpan via CORK

- CORK bound is extendible to Feature Sets
- For the enumeration of significant features:
  1. $\mathcal{F} = \emptyset$
  2. *gSpan* $\Rightarrow$ best subgraph S according to $q(\{S\} \cup \mathcal{F})$
  3. If $q(\{S\} \cup \mathcal{F}) > q(\mathcal{F})$:
  4. $\mathcal{F} = \mathcal{F} \cup S$
  5. GoTo 2.
  6. return $\mathcal{F}$

Experimental Validation

- NCI Dataset:
  - 6 chemical structure sets of variable size
  - Mapped to label “effective against cancer” (Y / N)
- Comparison to 2 feature ranking methods:
  - Sequential Cover via confidence score
  - Pearson Correlation
- Comparison to wrapper approach LAR-LASSO

Tsuda, K. : Entire regularization paths for graph data. *ICML* 2007

Results

- 10 repetitions of 10-fold cross validation
- on *gSpan* enumeration with freq. threshold 10%
- validated via a linear SVM

		Filter						Wrapper	
		SC		PC		CORK		LAR	
Dataset	#Features	Acc.	Std.	Acc.	Std.	Acc.	Std.	Acc.	Std.
NCI1	57	66.98	2.31	65.43	3.82	70.98	2.31	73.08	2.06
NCI33	53	66.50	2.57	64.15	3.46	70.08	2.76	72.81	2.51
NCI41	49	70.20	3.23	65.37	4.27	70.38	2.72	72.39	2.58
NCI47	56	67.04	2.35	67.00	3.45	71.42	2.22	72.62	2.07
NCI81	64	69.04	2.17	64.27	5.01	70.76	2.21	72.58	1.88

Outlook

- Improve runtime
  - Save minimal DFS Codes
  - Sharpen the bound for later iterations
  - Combine with other bounding criteria
  - → Subgraph mining procedure can be applied without giving a frequency threshold
- Exploit tree structure for decision tree learning

Thank you.

Forward Selection

Input: Set of features $\mathcal{F}$, FS criterion f

$\mathcal{T} := \emptyset$
$\mathcal{S} := \mathcal{F}$
$s = \textbf{argmax}\ \{f(\{s^*\}) : s^* \in \mathcal{S}\}$
while $f(\mathcal{T}) < f(\mathcal{T} \cup \{s\})$ **do**
  $\mathcal{T} := \mathcal{T} \cup \{s\}$
  $\mathcal{S} := \mathcal{S} \setminus \{s\}$
  $s = \textbf{argmax}\ \{f(\mathcal{T} \cup \{s^*\}) : s^* \in \mathcal{S}\}$
end

Output: selected feature set $\mathcal{T}$

CORK as a Threshold

$$S \sqsubseteq T, (A_{S_1} \geq A_{T_1} \wedge B_{S_1} \geq B_{T_1}) \qquad q(\{S\}) = -\left(A_{S_0} \cdot B_{S_0} + A_{S_1} \cdot B_{S_1}\right)$$

- T can loose all matches in B:

$$q(\{T\}) \leq -\left(A_{S_0} \cdot (B_{S_0} + B_{S_1}) + A_{S_1} \cdot 0\right) = -A_{S_0} \cdot (B_{S_0} + B_{S_1})$$

- T can loose all matches in A:

$$q(\{T\}) \leq -\left((A_{S_0} + A_{S_1}) \cdot B_{S_0} + 0 \cdot B_{S_1}\right) = -(A_{S_0} + A_{S_1}) \cdot B_{S_0}$$

- The maximal CORK value of T is thus

$$q(\{T\}) \leq \max\begin{Bmatrix} -A_{S_0} \cdot |B| \\ -|A| \cdot B_{S_0} \\ q(\{S\}) \end{Bmatrix} = q(\{S\}) + \max\begin{Bmatrix} (A_{S_1} - A_{S_0}) \cdot B_{S_1} \\ A_{S_1} \cdot (B_{S_1} - B_{S_0}) \\ 0 \end{Bmatrix}$$