


## Seleção de Atributos FSS



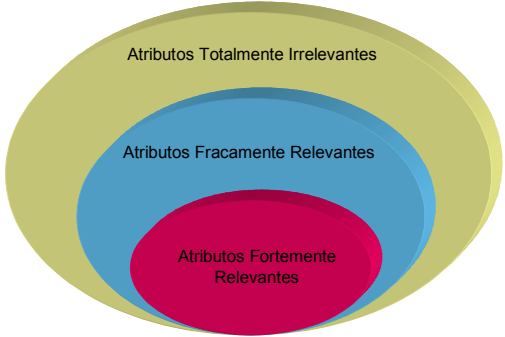
- ❑ Alguns indutores geralmente degradam seu desempenho quando são fornecidos muitos atributos irrelevantes para o conceito a ser aprendido
- ❑ *Feature Subset Selection* (FSS) é o processo de selecionar um subconjunto de atributos para posterior utilização pelo indutor

José Augusto Baranauskas      agosto@ffcirp.usp.br  
 Departamento de Física e Matemática – FFLRP-USP      http://dfm.ffcirp.usp.br/~augusto

## Relevância de Atributos

- ❑ A relevância de um atributo é dependente do conceito a ser aprendido e do indutor a ser utilizado
- ❑ Vários indutores assumem que os atributos originais que descrevem os exemplos são relevantes o suficiente para aprender a tarefa em questão
- ❑ Entretanto, alguns atributos podem não ser diretamente relevantes e outros até irrelevantes
- ❑ Um atributo é **irrelevante** se existe uma hipótese completa e consistente que não usa aquele atributo

## Relevância de Atributos



## Seleção de Atributos - FSS

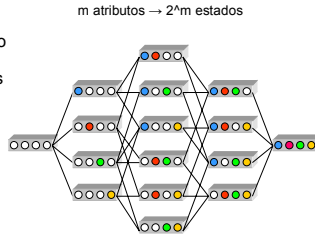
- ❑ **Objetivo:** selecionar um subconjunto de atributos para fornecer ao indutor
- ❑ **Motivação:**
  - Alguns indutores não trabalham bem com muitos atributos irrelevantes
  - Melhoria da precisão
  - Melhoria da compreensibilidade
- ❑ **Abordagens:**
  - Embutida
  - Filtro
  - *Wrapper*

## FSS como Busca no Espaço de Estados

- ❑ Uma forma conveniente para representar as abordagens para a seleção de atributos é a busca heurística, na qual cada estado no espaço de busca é composto por um subconjunto de possíveis atributos
- ❑ Desse modo, qualquer método de seleção de atributos pode ser caracterizado por sua instanciação em relação a quatro questões básicas, as quais determinam a natureza do processo de busca heurística:
  - Ponto de partida
  - Organização da busca
  - Estratégia de busca
  - Critério de parada

## Ponto de Partida

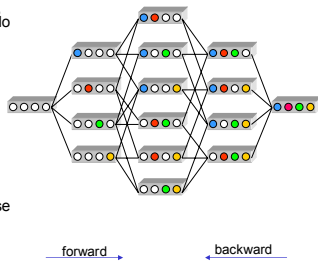
- ❑ A figura mostra subconjuntos possíveis para 4 atributos: círculos brancos indicam que o atributo em questão não foi selecionado; círculos coloridos indicam seleção do atributo para o processo de indução
- ❑ Cada estado no domínio de subconjuntos de atributos especifica quais atributos utilizar durante a indução
- ❑ Nota-se que os estados são parcialmente ordenados onde os estados à direita acrescentam um atributo aos da esquerda



m atributos → 2<sup>m</sup> estados

## Ponto de Partida

- Quando a busca tem seu estado inicial mais à esquerda, o conjunto vazio de atributos, representado pelos quatro círculos em branco, esse sentido da busca é conhecido **forward**
- Já a situação que inicia com o subconjunto contendo todos os atributos e sucessivamente removendo-os, é denominada de eliminação (ou sentido da busca) **backward**
- Também podem ser empregadas variações de ambas abordagens, selecionando-se um estado inicial em algum ponto do espaço de busca e movendo-se a partir desse ponto – busca **outward**



7

## Organização da Busca

- A cada ponto na busca, modificações locais no conjunto de atributos são consideradas; uma dessas é selecionada e uma nova iteração é realizada
- Claramente, uma busca exaustiva em todo o espaço de estados é impraticável, já que para um número  $m$  de atributos existem  $2^m$  possíveis estados
- Uma abordagem mais prática é a utilização de um método *greedy* para a travessia do espaço de busca: em cada ponto da busca, consideram-se alterações locais sobre o conjunto corrente de atributos, seleciona-se um novo atributo e realiza-se uma nova iteração
- Por exemplo, **hill-climbing**, considera tanto a adição quanto a remoção de atributos em cada ponto de decisão, permitindo ainda que a recuperação de uma decisão anterior possa ser realizada
- Dentre essas opções, pode-se considerar todos os estados gerados e selecionar o melhor, ou simplesmente escolher o primeiro estado que melhora a precisão sobre o conjunto corrente
- Pode-se, também, substituir o método *greedy* por métodos mais sofisticados, como, por exemplo, o método **best-first**, o qual, embora sendo mais custoso, é ainda computacionalmente viável para alguns domínios

8

## Estratégia de Busca

- A terceira questão a ser tratada considera a estratégia utilizada na avaliação dos subconjuntos alternativos de atributos
- Uma métrica normalmente empregada envolve a habilidade de um atributo discriminar as classes de um conjunto de treinamento
- Diversos algoritmos de indução incorporam um critério baseado na Teoria da Informação, enquanto outros medem diretamente a precisão do conjunto de treinamento ou de um conjunto separado de avaliação
- Outro aspecto importante é como a estratégia de FSS interage com o algoritmo básico de indução
- Essa interação pode ser subdividida em três abordagens:
  - (i) embutida, na qual FSS é embutida no algoritmo básico de indução;
  - (ii) filtro, na qual FSS é utilizada para filtrar atributos de forma independente do algoritmo de indução a ser utilizado e
  - (iii) *wrapper*, a qual emprega o próprio algoritmo de indução como uma caixa-preta

9

## Crítérios de Parada

- Alguns critérios mais utilizados são:
  - Parar de remover ou adicionar atributos quando nenhuma das alternativas melhora o desempenho do classificador
  - Continuar gerando subconjuntos de atributos até que um extremo do espaço de busca seja alcançado e escolher o melhor desses subconjuntos
  - Ordenar os atributos segundo algum critério e utilizar um parâmetro para determinar o ponto de parada, por exemplo, o número de atributos desejado no subconjunto

10

## FSS: Embutida

- Alguns indutores são capazes de realizar sua própria seleção de atributos de forma dinâmica, enquanto procuram por uma hipótese
- Em geral, a maioria dos algoritmos *eager* possuem uma abordagem embutida para a seleção de atributos
- Por exemplo, métodos de particionamento recursivo, tais como árvores de decisão, efetuam uma busca *greedy* através do espaço de árvores
  - A cada passo, eles usam uma função de avaliação para selecionar o atributo que tem a melhor capacidade de discriminar entre as classes
  - Eles particionam o conjunto de treinamento baseados nesse atributo e repetem o processo para cada subconjunto, estendendo a árvore até que nenhuma discriminação adicional seja possível
  - Este método é usado pelo indutor C4.5

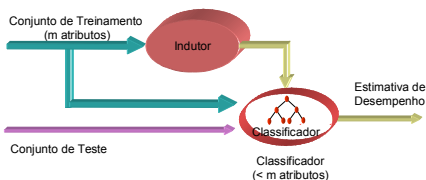
11

## FSS: Embutida

- Métodos de indução de regras também possuem seleção embutida de atributos
- Estes métodos usam uma função de avaliação para selecionar o atributo que ajuda a distinguir uma classe **C** das outras; então eles adicionam o teste resultante em uma única regra conjuntiva para essa classe **C**
- Eles repetem esse processo até que a regra exclua todos os exemplos de outras classes e então removem os exemplos da classe **C** que a regra cobre, repetindo esse processo nos exemplos de treinamento remanescentes
- Este método é empregado pelo indutor CN2

12

## FSS: Embutida



13

## FSS: Filtro

- ❑ Essa abordagem de seleção de atributos introduz um processo separado, o qual ocorre antes da aplicação do algoritmo de indução propriamente dito
- ❑ A idéia é filtrar atributos irrelevantes, segundo algum critério, antes de iniciar a indução
- ❑ Esse passo de pré-processamento considera características gerais do conjunto de exemplos para selecionar alguns atributos e excluir outros
- ❑ Sendo assim, métodos de filtros são independentes do algoritmo de indução que, simplesmente, receberá como entrada o conjunto de exemplos contendo apenas os atributos selecionados pelo filtro

14

## FSS: Filtro

- ❑ Qualquer algoritmo que efetue algum tipo de seleção pode ser usado para filtrar atributos
- ❑ A saída do algoritmo de filtragem é o conjunto de atributos por ele selecionados
- ❑ Os atributos restantes são removidos do conjunto de exemplos, reduzindo assim sua dimensão
- ❑ Após isso, o conjunto de exemplos reduzido pode ser usado por qualquer indutor
- ❑ Entretanto, atributos considerados como bons por um filtro não são, necessariamente, úteis para outras famílias de algoritmos que podem ter um bias de aprendizado diferente.
- ❑ Por exemplo, um algoritmo de indução de árvores de decisão pode ser usado como um filtro de atributos
  - O conjunto de atributos selecionado pela árvore constitui a saída do processo de filtragem, sendo a árvore descartada
- ❑ Existem dois algoritmos, especificamente desenvolvidos para atuarem com filtros de atributos, **Focus** e **Relief**

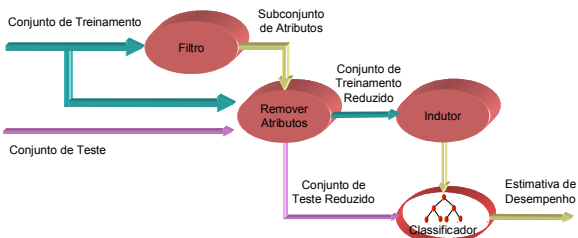
15

## FSS: Filtro

- ❑ Focus
  - O algoritmo Focus envolve um grau maior de busca no espaço de atributos
  - Esse algoritmo, inicialmente proposto para domínios booleanos sem ruído, procura exaustivamente pela combinação mínima de atributos que seja suficiente para descrever a classe de todos os exemplos de treinamento
  - Desse modo, esse método inicia a busca examinando cada atributo em separado, depois examina pares de atributos, triplas e assim por diante, até encontrar uma combinação que gera as melhores partições do conjunto de treinamento
- ❑ Relief
  - O algoritmo Relief incorpora uma função de avaliação de atributos mais complexa que o algoritmo Focus
  - Relief atribui uma relevância ponderada para cada atributo, que é definida para denotar a relevância do atributo em relação às classes
  - Relief é um algoritmo que usa amostras aleatórias dos exemplos e atualiza os valores de relevância baseado na diferença entre o exemplo selecionado e os dois exemplos mais próximos da mesma classe e de outra classe
- ❑ A principal diferença entre Relief e Focus é que, enquanto Focus procura por um conjunto mínimo de atributos, o algoritmo Relief procura por todos os atributos relevantes

16

## FSS: Filtro



17

## FSS: Wrapper

- ❑ Em contraste com filtros, a abordagem *wrapper* gera um subconjunto de atributos como candidato, executa o indutor com apenas esses atributos no conjunto de treinamento e usa a precisão do classificador extraído para avaliar o subconjunto de atributos em questão
- ❑ Este processo é repetido para cada subconjunto candidato, até que o critério de parada seja satisfeito
- ❑ A idéia geral por trás da abordagem é que o algoritmo de FSS existe como um *wrapper* ao redor do indutor e é responsável por conduzir a busca por um bom subconjunto de atributos
- ❑ A qualidade de um subconjunto candidato é avaliada utilizando o próprio indutor como uma caixa-preta
- ❑ O objetivo da busca é encontrar o subconjunto (nó) com a melhor qualidade, utilizando uma função heurística para guiá-la

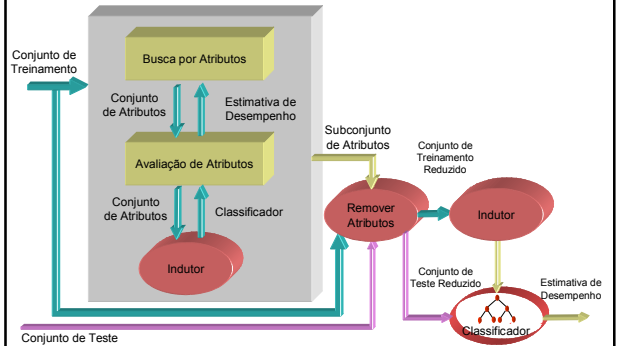
18

## FSS: Wrapper

- Em geral, a busca é conduzida no espaço do subconjunto de atributos, com os operadores adicionar ou remover, utilizando como busca o método hill-climbing ou best-first assim com direções *forward* e *backward* como direção da busca
- A precisão pode ser estimada por *cross-validation*
- Um argumento a favor da abordagem *wrapper* é que o mesmo algoritmo de indução que vai utilizar o subconjunto de atributos selecionado deve prover uma estimativa melhor de precisão que um outro algoritmo, o qual pode possuir um *bias* de indução totalmente diferente
- Por outro lado, essa abordagem pode ser computacionalmente dispendiosa, uma vez que o indutor deve ser executado para cada subconjunto de atributos considerado

19

## FSS: Wrapper



20

## FSS: Experimentos

- 11 conjuntos de exemplos
- 4 indutores aplicados através do *wrapper* (C4.5, CN2, NB, IB)
- 2 filtros: ID3 e CI (Mineset)
- 108 experimentos
- Mais de 32 dias de processamento ininterrupto para executar *wrapper*
- Maiores detalhes:
  - <http://dfm.ffclrp.usp.br/~augusto/publication.htm>

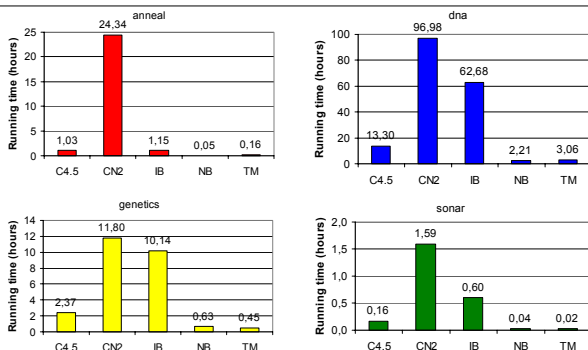
21

## Datasets Descriptions

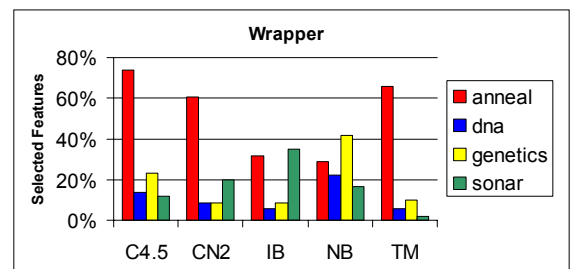
Dataset	# Instances	#Features (cont.disc)	Class	Class %	Majority Error	Missing Values
anneal	898	38 (6,32)	1	0.89%	23.83% on value 3	Yes
			2	11.02%		
			3	76.17%		
			5	7.46%		
			U	4.45%		
dna	3186	180 (0,180)	1	24.07%	48.09% on value 3	No
			2	24.01%		
			3	51.91%		
genetics	3190	60 (0,60)	N	51.88%	48.12% on value N	No
			EI	24.04%		
			IE	24.08%		
sonar	208	60 (60,0)	M	53.37%	46.63% on value M	No
			R	46.63%		

22

## Wrapper Execution Time

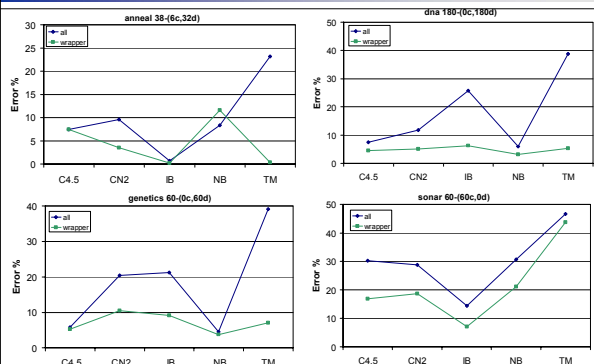


## Wrapper Selected Features

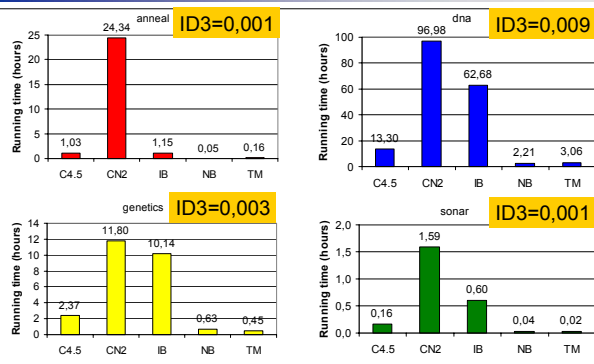


23

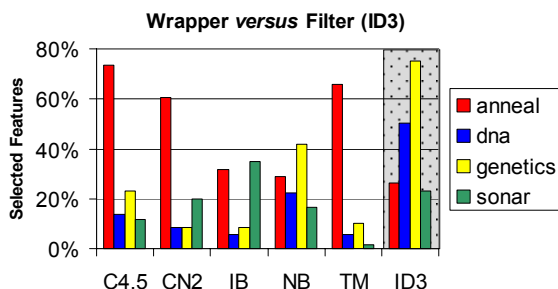
## Error Comparison



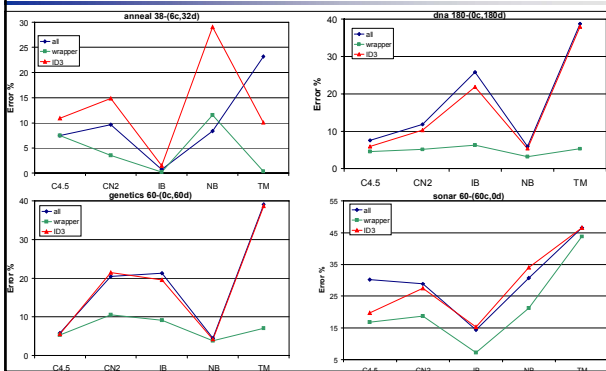
## Wrapper versus Filter Execution Time



## Wrapper versus Filter Selected Features



## Error Comparison



## FSS: Principais Resultados

- ❑ Filtros devem ser considerados antes de se cogitar a utilização de *wrappers*
- ❑ Filtros foram, em média, 1.000-10.000 vezes mais rápidos, com pequena perda de precisão
- ❑ FSS com o indutor CN2 causou um aumento no número de regras
- ❑ Se o classificador final for avaliado por um ser humano, FSS deve ser efetuada de forma cuidadosa

## Resumo

- ❑ O processo de FSS pode ser descrito como uma busca pelo espaço de estados
- ❑ Cada estado representa um subconjunto de atributos
- ❑ O valor de um nó é a precisão estimada
- ❑ Os operadores são geralmente adicionar/remover um atributo