SELEÇÃO DE ATRIBUTOS

Estagiário PAE: Pablo Andretta Jaskowiak
Professor: Ricardo J. G. B. Campello

SCC0173 | Mineração de Dados Biológicos

Créditos

- Parte do material a seguir consiste de adaptações e extensões dos originais cedidos gentilmente por:
  - Prof. Dr. André C. P. L. F. Carvalho
  - Prof. Dr. Eduardo R. Hruschka
  - Tan et al., *Introduction to Data Mining*, Addison-Wesley, 2006

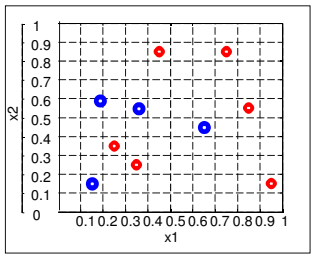
Sumário

Introdução

- Alguns conjuntos de dados podem ter um número muito grande de atributos
  - Por exemplo, bases de expressão gênica
    - poucas centenas ou dezenas de objetos...
    - mas milhares de atributos (bases com até 20 mil)
- Principal Problema:
  - no. de atributos = no. de dimensões
  - **Maldição da Dimensionalidade**

Maldição da Dimensionalidade

- Suponha atributos numéricos
  - objetos = pontos no espaço Euclidiano
    - valores dos atributos = coordenadas



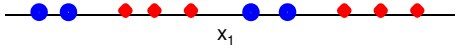
Maldição da Dimensionalidade

- Hiper-volume do espaço cresce de forma exponencial com a adição de novos atributos
  - Objetos formados por 1 atributo com 10 possíveis valores discretos: 10 possíveis objetos distintos
  - Objetos formados por 5 atributos com 10 possíveis valores discretos: 10^5 possíveis objetos
  - Dificuldade em problemas com poucos exemplos e muitos atributos
    - Dados se tornam muito esparsos

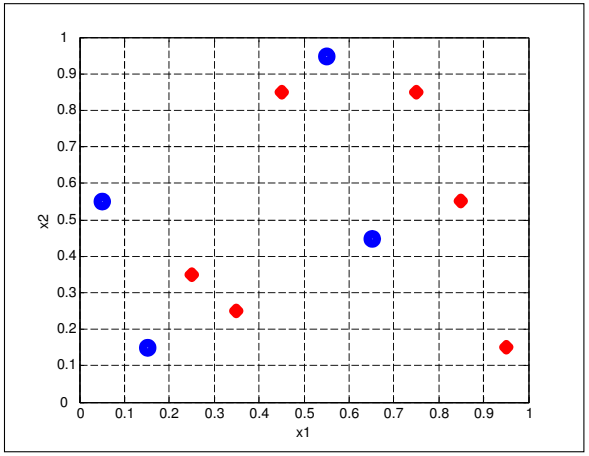
Maldição da Dimensionalidade

- **Problemas Críticos com Dados Esparsos**
  - **Problema 1:**
    - Muitas regiões do espaço dos atributos podem não estar representadas pelo conjunto de dados disponíveis
      - é muito provável que o modelo aprendido não represente bem (generalize) essas regiões
      - overfitting (e, portanto, impacto de ruído) é potencializado

Maldição da Dimensionalidade



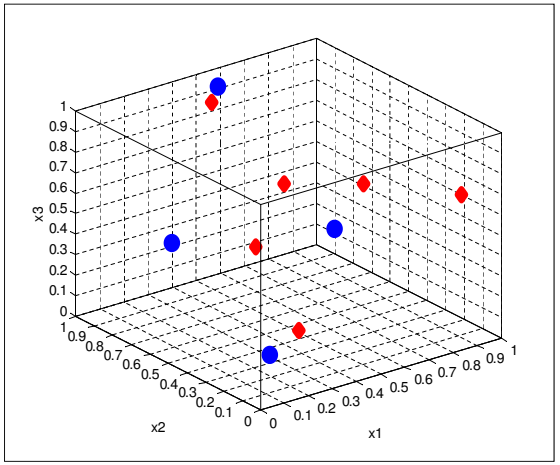
Maldição da Dimensionalidade



Exemplos positivos

Exemplos negativos

Maldição da Dimensionalidade



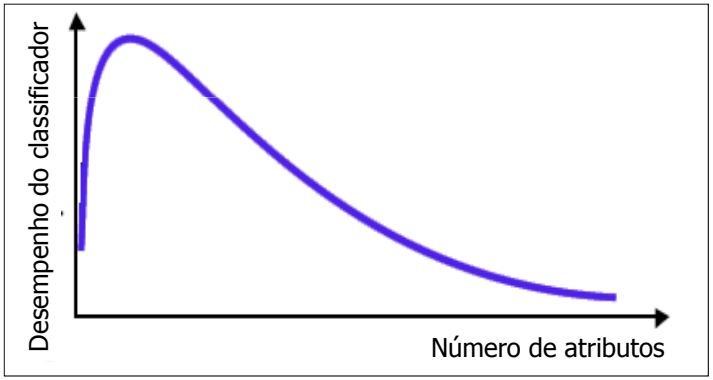
Exemplos positivos

Exemplos negativos

Maldição da Dimensionalidade

- Número de exemplos necessários para manter desempenho cresce exponencialmente com o número de atributos
- Na prática, usualmente o número de exemplos de treinamento é fixo
  - Não se pode obter exemplos à vontade
  - Logo, o desempenho do algoritmo de DM tende a se degradar a partir de um determinado no. de atributos
    - mesmo que sejam atributos úteis

Maldição da Dimensionalidade

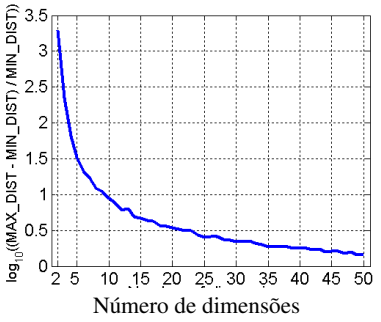


Maldição da Dimensionalidade

- **Problemas Críticos com Dados Esparsos**
  - **Problema 2:**
    - Exemplos parecem equidistantes
      - prejudica o desempenho de algoritmos que operam fundamentalmente com base em medidas de distância
      - p. ex.: KNN, algoritmos de clustering, detecção de outliers, ...

Exemplo

- Gerar aleatoriamente 500 pontos
  - Computar diferença entre distâncias máxima e mínima entre os pares de pontos



Redução de Dimensionalidade

- Principais Benefícios
  - Melhora eficácia de algoritmos de DM
    - ao eliminar atributos irrelevantes e/ou redundantes
  - Reduz o tamanho necessário da amostra
    - ao lidar com a maldição da dimensionalidade
  - Melhora a eficiência computacional dos algoritmos
  - Simplifica modelo gerado e facilita interpretação
  - Facilita visualização dos dados

Redução de Dimensionalidade

- Pode-se realizar essencialmente através de:
  - **Extração de Características**
  - **Seleção de Atributos**

Extração de Características

- Trata-se de extrair, a partir dos dados brutos, características de alto nível. Por exemplo:
  - bordas, contornos, etc. a partir de imagens
    - pixels não são bons atributos para reconhecimento de face...
  - componentes de freqüência em sinais de áudio
    - amostras de áudio no domínio do tempo podem não ser apropriadas para reconhecimento de voz...

Extração de Características

- Um tipo particular de extração de características é a **Transformação do Espaço de Atributos**
  - Gera um novo conjunto de atributos a partir da combinação de projeções dos atributos originais
  - **PCA (linear ou não linear)**: Atributos são ortogonais e ordenados segundo a parcela de informação que carregam
    - Descarte dos atributos menos representativos permite obter um novo espaço de dimensão menor que o original
      - retendo a maior parte possível da informação

Extração de Características

- **Transformação do Espaço de Atributos**
  - Vantagens:
    - Simples e relativamente rápida computacionalmente
  - Desvantagens:
    - Limitada a atributos numéricos
    - Interpretabilidade dos atributos originais é perdida
      - o que é proibitivo em determinados cenários de aplicação

Dimensionality Reduction: PCA

Dimensions = 206

Seleção de Atributos

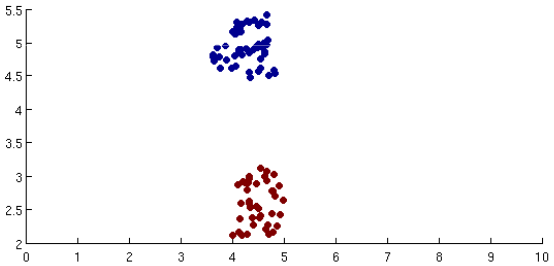
- **Seleção de atributos** assume que os atributos existentes já estão em uma forma apropriada, porém:
  - parte deles pode ser **irrelevante**
  - parte deles pode ser **redundante**
- Essa parte pode ser significativa e comprometer a qualidade do processo de DM
  - em áreas como bioinformática e text mining, por ex., é comum que centenas a milhares de atributos sejam desnecessários

Seleção de Atributos

- Atributos irrelevantes × redundantes
  - Irrelevantes
    - Não possuem informação útil para a tarefa em questão
      - Por exemplo, "nome" de uma ação para previsão do seu valor
      - Caso extremo: valor constante para todos os exemplos
  - Redundantes
    - Possuem a mesma informação útil para a tarefa em questão
      - Por ex., "salário" e "IR retido na fonte" p/ análise de crédito
      - Caso extremo: valores iguais ou proporcionais

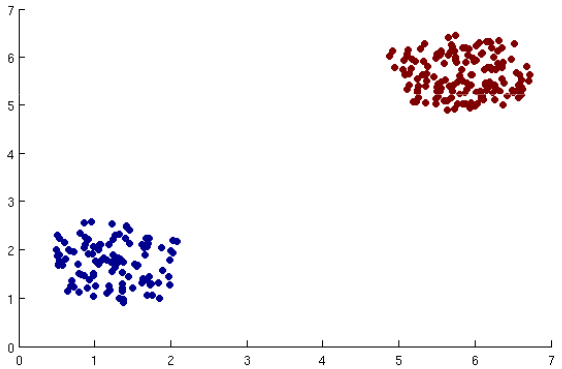
Seleção de Atributos

- Exemplo de Atributo Irrelevante



Seleção de Atributos

- Exemplo de Atributos Redundantes



Seleção de Atributos

- Pode ser feita por:
  - **Ordenação**
    - ou ranking
  - **Seleção de Subconjunto**

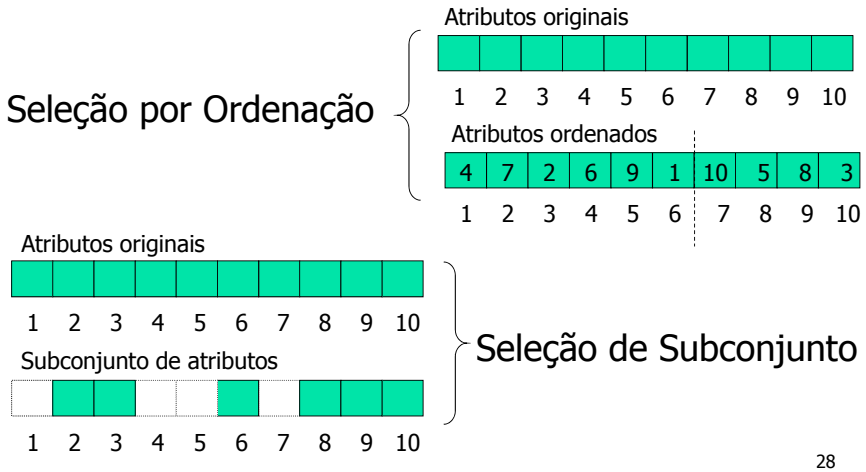
Seleção de Atributos

- **Ordenação** (ranking)
  - Avalia os atributos individualmente e os ordena de acordo com algum critério, selecionando uma parte
    - tipicamente selecionam-se os melhores posicionados
  - Critério pode levar em conta diferentes fatores
    - Por exemplo, relevância:
      - para discriminar classes individualmente (classificação)
      - para prever a saída individualmente (regressão)

Seleção de Atributos

- **Seleção de Subconjunto**
  - Seleciona um subconjunto de atributos mutuamente interessantes, segundo algum critério

Seleção de Atributos



Ordenação

- Exemplo de Ordenação
  - Usar o algoritmo 1-Rule para rankear os atributos de acordo com o erro de classificação individual
  - Tomar os n_s atributos com menor erro e utilizar em outro classificador mais sofisticado (e.g. RNA)

Ordenação

- Vantagem da Ordenação
  - Quantidade linear de avaliações do critério
    - igual ao número **n** de atributos
  - Contraste com a Seleção de Subconjunto: no. de possíveis combinações de $\mathbf{n_s}$ dentre **n** atributos é...

$$\binom{n}{n_s} = \frac{n!}{(n-n_s)!n_s!}$$

Ordenação

- Deficiência da Ordenação
  - Muitas vezes não captura inter-dependências entre atributos
    - atributos podem ser inúteis sozinhos porém úteis em conjunto
    - ou podem ser tão úteis sozinhos quanto em conjunto (redundantes)
  - Melhores n_s atributos dificilmente constituem o melhor subconjunto de n_s atributos
  - O melhor subconjunto é aquele mais complementar !

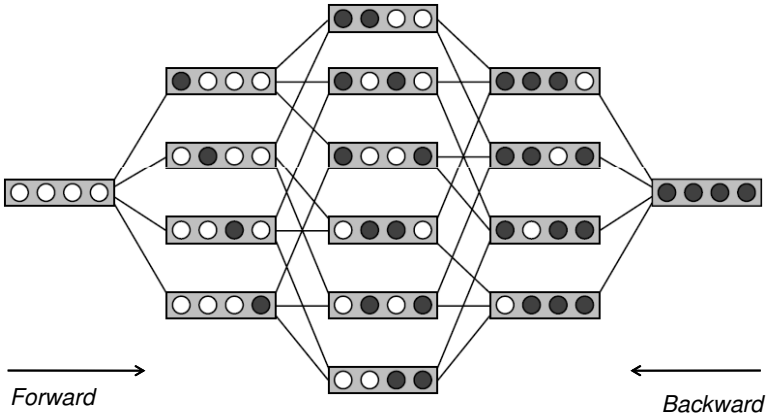
Seleção de Subconjunto

- Problema de busca
- Para *n* atributos: $2^n - 1$ subconjuntos possíveis
  - Avaliar todos os subconjuntos é inviável
- Busca heurística
  - Alguns subconjuntos são **avaliados** segundo algum critério
    - até que um critério de **parada** seja satisfeito
  - A escolha de quais subconjuntos serão avaliados depende da **estratégia de busca** utilizada

Seleção de Subconjunto

- **Estratégias de Busca**
  - Backward Elimination
    - Começa com todos os atributos e remove um por vez
  - Forward Selection
    - Começa sem nenhum atributo e inclui um atributo por vez
  - Bidirectional Search
    - Busca pode começar em qualquer ponto e atributos podem ser adicionados e removidos
  - Random Search
    - Ponto de partida da busca e atributos a serem removidos ou adicionados são decididos de forma estocástica

Seleção de Subconjunto



Seleção de Subconjunto

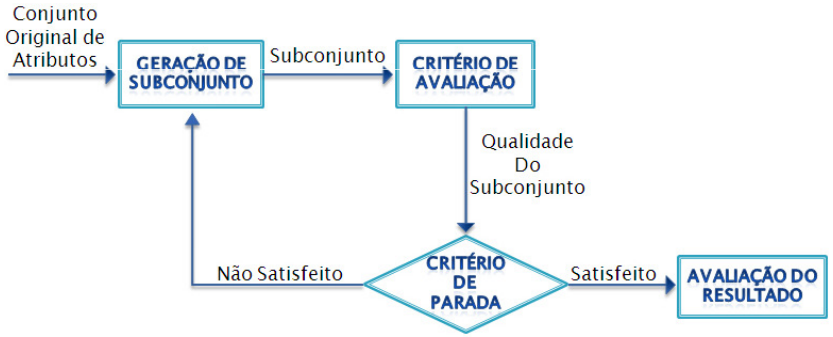
- **Critérios de Avaliação**
  - Inerente ao método de seleção de atributos
  - Critérios independentes (Filtros – discussão subsequente)
    - Medidas de correlação
    - Medidas de informação
    - Medidas de dependência
    - Medidas de consistência
  - Critérios dependentes (*Wrappers* – discussão subsequente)
    - Algoritmo alvo usado para a tarefa de interesse

Seleção de Subconjunto

- **Critério de Parada**
  - Dependente do método de busca utilizado
    - Número máximo de iterações
    - Valor de avaliação obtido é satisfatório
    - ...

Seleção de Subconjunto

- Visão geral do processo de seleção de subconjunto



Seleção de Atributos

- Categorias de Métodos (taxonomia independente):
  - **Embarcados (Embedded)**
    - seleção de atributos ocorre naturalmente e internamente como parte do algoritmo de DM
  - **Wrappers**
    - seleção de atributos envolve o algoritmo de DM alvo, que é usado para guiar o processo de seleção
  - **Filtros**
    - seleção de atributos é realizada a priori e não envolve o algoritmo de DM a ser aplicado depois (algoritmo alvo)

Abordagens Embarcadas

- Seleção de atributos faz parte da estratégia de aprendizado do modelo
  - ocorre naturalmente e internamente como parte do algoritmo de DM
- Já estudamos um exemplo clássico:
  - Classificadores baseados em Árvores de Decisão !
    - Realizam procedimento guloso de **seleção de subconjunto**

Wrappers

- Utilizam alguma estratégia para executar uma busca (no espaço de subconjuntos de atributos) ou ordenação de forma guiada pelo algoritmo de DM alvo
  - O uso do algoritmo alvo implica guiar a seleção em direção aos atributos que **maximizam o desempenho alvo**
  - No entanto, em geral implica tornar o método **muito custoso** (possivelmente proibitivo) em termos computacionais

Wrapper Naive Bayes

- Algoritmo Guloso de **Seleção de Subconjunto**:
  - Selecione o melhor classificador Naive Bayes com um único atributo (avaliando todos em um conjunto de dados de teste)
  - Enquanto houver melhora no desempenho do classificador faça
    - Selecione o melhor classificador Naive Bayes com os atributos já selecionados anteriormente adicionados a um dentre os atributos ainda não selecionados

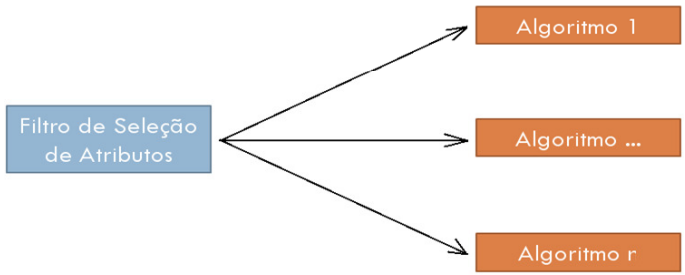
- **Nota**: Apesar de ser um wrapper, o algoritmo acima é relativamente rápido devido à sua simplicidade e também à eficiência computacional do Naive Bayes !

Filtros

- Utilizam alguma estratégia para executar uma busca (no espaço de subconjuntos de atributos) ou ordenação, guiada apenas por propriedades intrínsecas aos próprios dados
  - Seleção NÃO envolve o algoritmo de DM alvo
  - Por um lado, isso tem como **desvantagem** guiar a seleção de forma indireta, o que pode levar a resultados inferiores
  - Por outro lado, isso tipicamente apresenta como **principal vantagem** a rapidez de processamento

Filtros

- Outra Vantagem: seleção não depende do algoritmo que será posteriormente utilizado



- Veremos a seguir exemplos dos critérios de seleção de alguns filtros

Filtros (Ordenação)

- **Diff**
- **Golub**
- **SVM RFE**
- **Relief**

Filtros de Ordenação

- **Diff**
  - Atributos numéricos
  - Diferença entre a média dos valores para duas classes

$$Diff(A) = \mu_+ - \mu_-$$

A - Atributo
μ_+ - Média para o atributo na classe +
μ_- - Média para o atributo na classe -

  - Selecionar os n_s atributos com maior valor em módulo

Exercício

- Dada a base de dados abaixo crie um *rank* dos atributos de acordo com o critério *Diff*

Att1	Att2	Att3	Att4	Att5	Att6	Att7	Att8	Classe
45.6	11.0	2.3	4.0	12.3	42.9	670.9	12.3	1
54.7	9.5	5.4	4.3	45.5	34.8	553.2	54.2	2
30.1	12.0	8.5	7.4	31.0	46.6	689.0	23.6	1
32.3	10.0	3.7	59.5	23.8	52.4	664.2	80.5	1
44.2	10.5	5.0	6.6	54.3	66.8	600.3	76.3	2
50.2	11.5	8.2	9.7	36.5	15.3	610.4	11.0	2
36.5	9.0	9.0	3.2	33.3	23.3	676.6	43.9	1

Filtros de Ordenação

- **Golub**
  - Valores do atributo devem ser diferentes para classes diferentes
  - Valores do atributo devem variar pouco para a mesma classe

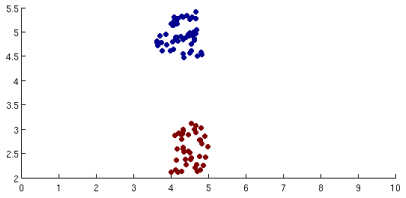
$$P(A) = \frac{\mu_+ - \mu_-}{\sigma_+ + \sigma_-}$$

μ_+ - Média do atributo na classe +
μ_- - Média do atributo na classe -
σ_+ - Desvio padrão para atributo na classe +
σ_- - Desvio padrão para atributo na classe -

Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J., Coller, H., Loh, M., Downing, J., Caligiuri, M., Bloomfield, C., Lander, E. S., Molecular Classifiation of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring, Science, 286 (1999) 531-537.

Filtros de Ordenação

- **Golub**
  - Reflete a diferença entre as classes relativa aos seus desvios padrão
  - Quanto maior a compactação e separação das classes
    - Maior o valor da avaliação para o atributo (em módulo)
  - Selecionar os n_s atributos com maior valor em módulo



Exercício

- Dada a base de dados abaixo crie um *rank* dos atributos de acordo com o critério de Golub

Att1	Att2	Att3	Att4	Att5	Att6	Att7	Att8	Classe
45.6	11.0	2.3	4.0	12.3	42.9	670.9	12.3	1
54.7	9.5	5.4	4.3	45.5	34.8	553.2	54.2	2
30.1	12.0	8.5	7.4	31.0	46.6	689.0	23.6	1
32.3	10.0	3.7	59.5	23.8	52.4	664.2	80.5	1
44.2	10.5	5.0	6.6	54.3	66.8	600.3	76.3	2
50.2	11.5	8.2	9.7	36.5	15.3	610.4	11.0	2
36.5	9.0	9.0	3.2	33.3	23.3	676.6	43.9	1

Filtros de Ordenação

- **SVM RFE**
  - Baseia-se na idéia de que
    - atributos pouco importantes pouco influenciam na definição do hiperplano separador de classificadores tipo SVMs
  - A cada passo um classificador SVM é treinado
  - Os pesos do classificador resultante são utilizados para criar um rank dos atributos
  - O atributo com menor rank é removido e o processo continua até que não reste nenhum atributo

Guyon, I., Weston, J., Barnhill, S., and Vapnik, V. (2002), Gene Selection for Cancer Classification using Support Vector Machines, Machine Learning, 46: 389–422.

Implementado no WEKA: SVMAttributeEval

Filtros de Ordenação

- **Relief**
  - Bons atributos
    - Valores iguais para instâncias de uma mesma classe
    - Valores diferentes para instâncias de classes diferentes
  - Escolher aleatoriamente uma instância e encontrar
    - Instância mais próxima de classe diferente (nearest miss)
    - Instância mais próxima de mesma classe (nearest hit)
  - Peso de cada atributo A é definido como

    W(A) = P(valores diferentes de A para uma instância e sua nearest miss)
    -
    P(valores diferentes de A para uma instância e sua nearest hit)
  - Como estimar as probabilidades?

Filtros de Ordenação

- **Relief**

```
Para todo A, W(A) = 0        /* A = 1, ..., n                */
Para i de 1 até m            /* m = tamanho da amostra       */
  Selecionar aleatoriamente uma instância R
  Encontrar as instâncias
      M: Mais próxima de mesma classe
      D: Mais próxima de classe diferente
  Para A de 1 até n
      W(A) = W(A) + diff(A,R,D)/m - diff(A,R,M)/m
  FimPara
FimPara
```

`diff(A,R,X)` é a diferença entre o valor do atributo A em R e M (ou D)

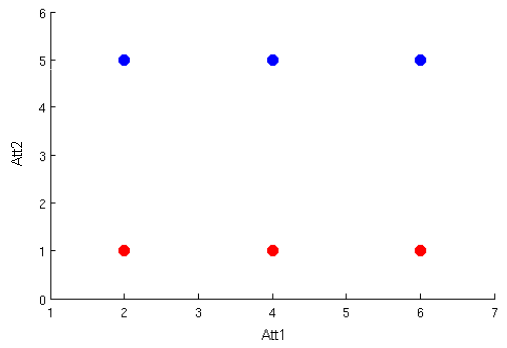
RELIEF-F: Extensão que utiliza *k* vizinhos nas estimativas

Kenji Kira, Larry A. Rendell, *A Practical Approach to Feature Selection*, In Ninth International Workshop on Machine Learning, 249-256, 1992.

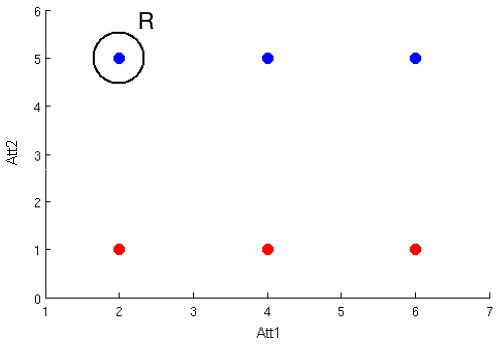
Implementado no WEKA: ReliefFAttributeEval

Filtros de Ordenação

- Exemplo de aplicação do Relief



Filtros de Ordenação



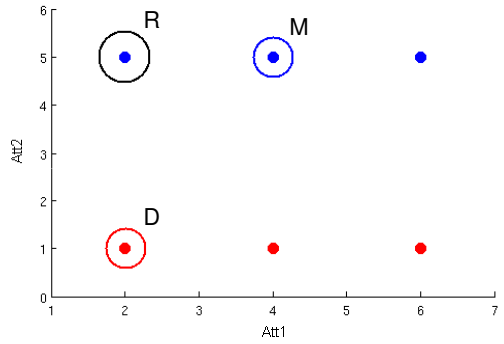
Peso dos Atributos	
W(Att1)	0
W(Att2)	0

`W(A) = W(A) + diff(A,R,D)/m - diff(A,R,M)/m`

`diff(A,R,X)` é a diferença entre o valor do atributo A em R e X
`X = M:` Instância mais próxima de mesma classe
`X = D:` Inst. mais próxima de classe diferente

m = 2
Distância máxima Att1: 4
Distância máxima Att2: 4

Filtros de Ordenação



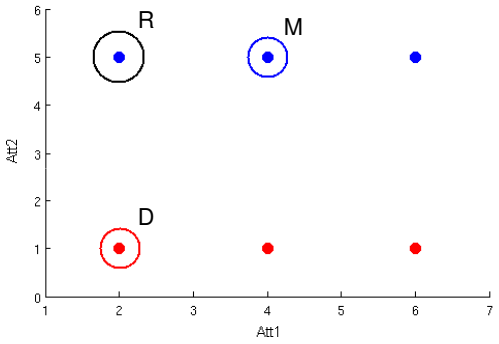
Peso dos Atributos	
W(Att1)	0
W(Att2)	0

`W(A) = W(A) + diff(A,R,D)/m - diff(A,R,M)/m`

`diff(A,R,X)` é a diferença entre o valor do atributo A em R e X
`X = M:` Instância mais próxima de mesma classe
`X = D:` Inst. mais próxima de classe diferente

m = 2
Distância máxima Att1: 4
Distância máxima Att2: 4

Filtros de Ordenação



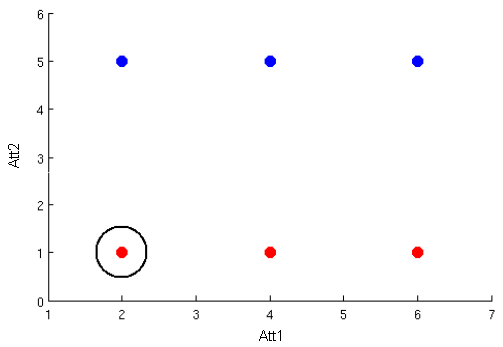
Peso dos Atributos	
W(Att1)	- 0.25
W(Att2)	+ 0.50

`W(A) = W(A) + diff(A,R,D)/m - diff(A,R,M)/m`

`diff(A,R,X)` é a diferença entre o valor do atributo A em R e X
`X = M:` Instância mais próxima de mesma classe
`X = D:` Inst. mais próxima de classe diferente

m = 2
Distância máxima Att1: 4
Distância máxima Att2: 4

Filtros de Ordenação



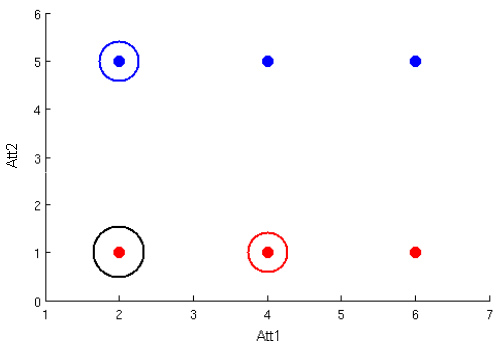
Peso dos Atributos	
W(Att1)	- 0.25
W(Att2)	+ 0.50

`W(A) = W(A) + diff(A,R,D)/m - diff(A,R,M)/m`

`diff(A,R,X)` é a diferença entre o valor do atributo A em R e X
`X = M:` Instância mais próxima de mesma classe
`X = D:` Inst. mais próxima de classe diferente

m = 2
Distância máxima Att1: 4
Distância máxima Att2: 4

Filtros de Ordenação



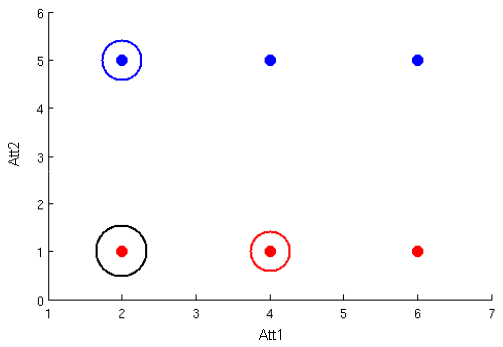
Peso dos Atributos	
W(Att1)	- 0.25
W(Att2)	+ 0.50

`W(A) = W(A) + diff(A,R,D)/m - diff(A,R,M)/m`

`diff(A,R,X)` é a diferença entre o valor do atributo A em R e X
`X = M:` Instância mais próxima de mesma classe
`X = D:` Inst. mais próxima de classe diferente

m = 2
Distância máxima Att1: 4
Distância máxima Att2: 4

Filtros de Ordenação



Peso dos Atributos	
W(Att1)	- 0.50
W(Att2)	+ 1.00

`W(A) = W(A) + diff(A,R,D)/m - diff(A,R,M)/m`

`diff(A,R,X)` é a diferença entre o valor do atributo A em R e X
`X = M:` Instância mais próxima de mesma classe
`X = D:` Inst. mais próxima de classe diferente

m = 2
Distância máxima Att1: 4
Distância máxima Att2: 4

Exercícios

- Repita o exemplo anterior
  - Agora com m = 6
- Aplique o algoritmo Relief na base abaixo

Outlook	Temp	Humidity	Windy	Play
Sunny	Hot	High	False	YES
Sunny	Hot	High	False	NO
Overcast	Hot	Normal	True	YES
Overcast	Cool	High	False	YES
Overcast	Mild	Normal	True	YES
Overcast	Mild	High	False	NO
Rainny	Mild	Normal	True	YES
Rainny	Mild	Normal	True	YES
Rainny	Mild	Normal	True	NO
Rainny	Mild	Normal	True	NO

Filtros (Seleção de Subconjunto)

- **Las Vegas Filter (LVF ou CSE)**
- **Correlation-based Feature Selection (CFS)**

LVF

- **Las Vegas Filter (LVF)**
  - Consistency Subset Eval (CSE) no WEKA
- Estratégia de busca
  - Geração aleatória de subconjuntos
- Critério de avaliação dos subconjuntos gerados
  - Consistência
- Critério de parada
  - Número máximo de iterações

Liu H. and Setiono R., *A Probabilistic Approach to Feature Selection - A Filter Solution,* In Proceedings of the 13th International Conference on Machine Learning.

Implementado no WEKA: ConsistencySubsetEval

LVF – Algoritmo

```
Entradas:
 No_MAX_ITER
 D = conj. total atributos
 n = no. de atributos
 t = limiar aceitável de
     inconsistência
```

```
1.  S_best = D; C_best = n;
2.  Para i = 1 até No_MAX_ITER
3.      S = conjAleatorio(D);
4.      C = numAtributos(S);
5.      Se (C < C_best)
6.          Se (inconCheck(S) < t)
7.              S_best = S;
8.              C_best = C;
9.              imprima(S);
10.     Senão
11.         Se (C = C_best) e (inconCheck(S) < t)
12.             imprima(S);
13. Fim
```


Taxa de Inconsistência

- **inconCheck**(S) – Taxa de Inconsistência
  - Duas ou mais instâncias são inconsistentes se
    - todos os valores dos atributos são iguais, exceto a classe
  - Para cada conjunto de instâncias inconsistentes
    - Contagem de inconsistência (i): Número de instâncias inconsistentes menos o no. de instâncias da classe mais freqüente do conjunto
      - representa o menor número de instâncias que precisam mudar de classe para que o conjunto deixe de ser inconsistente !
  - Taxa de inconsistência (Ti): Soma das contagens de inconsistência dividida pelo número total de instâncias

LVF – Cálculo de InconCheck

Outlook	Temp	Humidity	Windy	Play
Sunny	Hot	High	False	YES
Sunny	Hot	High	False	NO
Overcast	Hot	Normal	True	YES
Overcast	Cool	High	False	YES
Overcast	Mild	Normal	True	YES
Overcast	Mild	High	False	NO
Rainny	Mild	Normal	True	YES
Rainny	Mild	Normal	True	YES
Rainny	Mild	Normal	True	NO
Rainny	Mild	Normal	True	NO

Considere que a base de dados ao lado contenha apenas um subconjunto S de atributos selecionados aleatoriamente pela rotina **conjAleatorio**(D)

LVF – Cálculo de InconCheck

Outlook	Temp	Humidity	Windy	Play
Sunny	Hot	High	False	YES
Sunny	Hot	High	False	NO
Overcast	Hot	Normal	True	YES
Overcast	Cool	High	False	YES
Overcast	Mild	Normal	True	YES
Overcast	Mild	High	False	NO
Rainny	Mild	Normal	True	YES
Rainny	Mild	Normal	True	YES
Rainny	Mild	Normal	True	NO
Rainny	Mild	Normal	True	NO

1º conjunto de instâncias inconsistentes detectado

LVF – Cálculo de InconCheck

Outlook	Temp	Humidity	Windy	Play
Sunny	Hot	High	False	YES
Sunny	Hot	High	False	NO
Overcast	Hot	Normal	True	YES
Overcast	Cool	High	False	YES
Overcast	Mild	Normal	True	YES
Overcast	Mild	High	False	NO
Rainny	Mild	Normal	True	YES
Rainny	Mild	Normal	True	YES
Rainny	Mild	Normal	True	NO
Rainny	Mild	Normal	True	NO

No. de instâncias:
YES: 1
NO: 1

No. de instâncias da classe mais frequente:
max(1,1) = 1

Contagem de Inconsistência:

$$i = (2-1) = 1$$

LVF – Cálculo de InconCheck

Outlook	Temp	Humidity	Windy	Play
Sunny	Hot	High	False	YES
Sunny	Hot	High	False	NO
Overcast	Hot	Normal	True	YES
Overcast	Cool	High	False	YES
Overcast	Mild	Normal	True	YES
Overcast	Mild	High	False	NO
Rainny	Mild	Normal	True	YES
Rainny	Mild	Normal	True	YES
Rainny	Mild	Normal	True	NO
Rainny	Mild	Normal	True	NO

2º conjunto de instâncias inconsistentes detectado

LVF – Cálculo de InconCheck

Outlook	Temp	Humidity	Windy	Play
Sunny	Hot	High	False	YES
Sunny	Hot	High	False	NO
Overcast	Hot	Normal	True	YES
Overcast	Cool	High	False	YES
Overcast	Mild	Normal	True	YES
Overcast	Mild	High	False	NO
Rainny	Mild	Normal	True	YES
Rainny	Mild	Normal	True	YES
Rainny	Mild	Normal	True	NO
Rainny	Mild	Normal	True	NO

No. de instâncias:
YES: 2
NO: 2

No. de instâncias da classe mais frequente:
max(2,2) = 2

Contagem de Inconsistência:

$$i = (4-2) = 2$$

LVF – Cálculo de InconCheck

Outlook	Temp	Humidity	Windy	Play
Sunny	Hot	High	False	YES
Sunny	Hot	High	False	NO
Overcast	Hot	Normal	True	YES
Overcast	Cool	High	False	YES
Overcast	Mild	Normal	True	YES
Overcast	Mild	High	False	NO
Rainny	Mild	Normal	True	YES
Rainny	Mild	Normal	True	YES
Rainny	Mild	Normal	True	NO
Rainny	Mild	Normal	True	NO

$$Ti = \frac{(1+2)}{10} = 0.3$$

LVF

- Dentro do número máximo de iterações
  - seleciona o menor subconjunto com Taxa de Inconsistência aceitável
- Podem ser utilizadas outras estratégias de busca
  - não simplesmente aleatórias
- Algoritmo original opera somente sobre atributos categóricos

CFS

- **CFS**
  - Correlation-based Feature Selection
- Geração de subconjuntos
  - Forward Selection
- Critério de avaliação dos subconjuntos gerados
  - Baseado em correlações
- Critério de parada
  - Adição de atributos não melhora o valor de avaliação

Hall, M.A., *Correlation-Based Feature Selection for Machine Learning*, Ph.D thesis, University of Waikato, New Zealand , 1998.

Implementado no WEKA: CfsSubsetEval

CFS

- Idéia básica
  - Um bom subconjunto de atributos possui
    - Atributos altamente correlacionados com a classe
    - Atributos pouco ou descorrelacionados entre si
  - Em outras palavras
    - Bons subconjuntos são formados por atributos relevantes e não redundantes
- Como quantificar?

CFS

- Critério de Avaliação

$$M_S = \frac{k\overline{r_{cf}}}{\sqrt{k + k(k-1)\overline{r_{ff}}}}$$

k - Número de atributos

$\overline{r_{ff}}$ - Correlação média entre pares atributo-atributo

$\overline{r_{cf}}$ - Correlação média entre pares atributo-classe

CFS

- Como calcular a correlação entre atributos?
  - Utiliza a *Symmetrical Uncertainty*
    - Usa o Ganho de Informação, porém corrige seu viés de favorecer atributos com mais valores
    - Normalizada entre [0,1]

$$SU(X,Y) = 2\frac{\Delta_{info}}{(Entropia(X) + Entropia(Y))}$$

  - Atributos numéricos?
    - Discretizar (abordagem original, implementada no Weka)
    - Usar outra medida de correlação (p. ex. Pearson para r_{cf})

CFS

- Exemplo de Aplicação

Outlook	Temp	Humidity	Windy	Play
Sunny	Hot	High	False	YES
Sunny	Hot	High	False	NO
Overcast	Hot	Normal	True	YES
Overcast	Cool	High	False	YES
Overcast	Mild	Normal	True	YES
Overcast	Mild	High	False	NO
Rainny	Mild	Normal	True	YES
Rainny	Mild	Normal	True	YES
Rainny	Mild	Normal	True	NO
Rainny	Mild	Normal	True	NO

CFS

- Matriz de correlação entre atributos

	Outlook	Temp	Humidity	Wind	Play
Outlook	1.000	0.151	0.016	0.004	0.196
Temp		1.000	0.293	0.030	0.023
Humidity			1.000	0.000	0.156
Wind				1.000	0.050
Play					1.000

CFS

- Após a geração da matriz de correlações é dado inicio ao processo de busca
- Seleção Forward
  - Inicia com um conjunto vazio de atributos
  - A cada iteração é adicionado o atributo que promove o maior crescimento do critério de avaliação (Ms)
- Critério de parada
  - Quando a adição de qualquer atributo não melhora a função de avaliação

Subconjunto	k	$\overline{r_{cf}}$	$\overline{r_{ff}}$	Mérito (Ms)
[ ]	0	-	-	-

$$M_S = \frac{k\overline{r_{cf}}}{\sqrt{k + k(k-1)\overline{r_{ff}}}}$$

	Outlook	Temp	Humidity	Wind	Play
Outlook	1.000	0.151	0.016	0.004	0.196
Temp		1.000	0.293	0.030	0.023
Humidity			1.000	0.000	0.156
Wind				1.000	0.050
Play					1.000

Subconjunto	k	$\overline{r_{cf}}$	$\overline{r_{ff}}$	Mérito (Ms)
[ ]	0	-	-	-
[Outlook]	1	0.196	1.000	
[Temp]	1	0.023	1.000	
[Humidity]	1	0.156	1.000	
[Wind]	1	0.050	1.000	

$$M_S = \frac{k\overline{r_{cf}}}{\sqrt{k + k(k-1)\overline{r_{ff}}}}$$

	Outlook	Temp	Humidity	Wind	Play
Outlook	1.000	0.151	0.016	0.004	0.196
Temp		1.000	0.293	0.030	0.023
Humidity			1.000	0.000	0.156
Wind				1.000	0.050
Play					1.000

Subconjunto	k	$\overline{r_{cf}}$	$\overline{r_{ff}}$	Mérito (Ms)
[ ]	0	-	-	-
[Outlook]	1	0.196	1.000	0.196
[Temp]	1	0.023	1.000	0.023
[Humidity]	1	0.156	1.000	0.156
[Wind]	1	0.050	1.000	0.050

$$M_S = \frac{k\overline{r_{cf}}}{\sqrt{k + k(k-1)\overline{r_{ff}}}}$$

	Outlook	Temp	Humidity	Wind	Play
Outlook	1.000	0.151	0.016	0.004	0.196
Temp		1.000	0.293	0.030	0.023
Humidity			1.000	0.000	0.156
Wind				1.000	0.050
Play					1.000

Subconjunto	k	$\overline{r_{cf}}$	$\overline{r_{ff}}$	Mérito (Ms)
[ ]	0	-	-	-
[Outlook]	**1**	**0.196**	**1.000**	**0.196**
[Temp]	1	0.023	1.000	0.023
[Humidity]	1	0.156	1.000	0.156
[Wind]	1	0.050	1.000	0.050

$$M_S = \frac{k\overline{r_{cf}}}{\sqrt{k + k(k-1)\overline{r_{ff}}}}$$

	Outlook	Temp	Humidity	Wind	Play
Outlook	1.000	0.151	0.016	0.004	0.196
Temp		1.000	0.293	0.030	0.023
Humidity			1.000	0.000	0.156
Wind				1.000	0.050
Play					1.000

Subconjunto	k	$\overline{r_{cf}}$	$\overline{r_{ff}}$	Mérito (Ms)
[ ]	0	-	-	-
[Outlook]	**1**	**0.196**	**1.000**	**0.196**
[Temp]	1	0.023	1.000	0.023
[Humidity]	1	0.156	1.000	0.156
[Wind]	1	0.050	1.000	0.050
[Outlook, Temp]	2	0.109	0.151	
[Outlook, Humidity]	2	0.176	0.016	
[Outlook, Wind]	2	0.123	0.004	

$$M_S = \frac{k\overline{r_{cf}}}{\sqrt{k + k(k-1)\overline{r_{ff}}}}$$

	Outlook	Temp	Humidity	Wind	Play
Outlook	1.000	0.151	0.016	0.004	0.196
Temp		1.000	0.293	0.030	0.023
Humidity			1.000	0.000	0.156
Wind				1.000	0.050
Play					1.000

Subconjunto	k	$\overline{r_{cf}}$	$\overline{r_{ff}}$	Mérito (Ms)
[ ]	0	-	-	-
[Outlook]	**1**	**0.196**	**1.000**	**0.196**
[Temp]	1	0.023	1.000	0.023
[Humidity]	1	0.156	1.000	0.156
[Wind]	1	0.050	1.000	0.050
[Outlook, Temp]	2	0.109	0.151	0.143
[Outlook, Humidity]	2	0.176	0.016	0.246
[Outlook, Wind]	2	0.123	0.004	0.173

$$M_S = \frac{k\overline{r_{cf}}}{\sqrt{k + k(k-1)\overline{r_{ff}}}}$$

	Outlook	Temp	Humidity	Wind	Play
Outlook	1.000	0.151	0.016	0.004	0.196
Temp		1.000	0.293	0.030	0.023
Humidity			1.000	0.000	0.156
Wind				1.000	0.050
Play					1.000

Subconjunto	k	$\overline{r_{cf}}$	$\overline{r_{ff}}$	Mérito (Ms)
[ ]	0	-	-	-
[Outlook]	**1**	**0.196**	**1.000**	**0.196**
[Temp]	1	0.023	1.000	0.023
[Humidity]	1	0.156	1.000	0.156
[Wind]	1	0.050	1.000	0.050
[Outlook, Temp]	2	0.109	0.151	0.143
[Outlook, Humidity]	**2**	**0.176**	**0.016**	**0.246**
[Outlook, Wind]	2	0.123	0.004	0.173

$$M_S = \frac{k\overline{r_{cf}}}{\sqrt{k + k(k-1)\overline{r_{ff}}}}$$

	Outlook	Temp	Humidity	Wind	Play
Outlook	1.000	0.151	0.016	0.004	0.196
Temp		1.000	0.293	0.030	0.023
Humidity			1.000	0.000	0.156
Wind				1.000	0.050
Play					1.000

Subconjunto	k	$\overline{r_{cf}}$	$\overline{r_{ff}}$	Mérito (Ms)
[ ]	0	-	-	-
[Outlook]	1	0.196	1.000	0.196
[Temp]	1	0.023	1.000	0.023
[Humidity]	1	0.156	1.000	0.156
[Wind]	1	0.050	1.000	0.050
[Outlook, Temp]	2	0.109	0.151	0.143
[Outlook, Humidity]	**2**	**0.176**	**0.016**	**0.246**
[Outlook, Wind]	2	0.123	0.004	0.173
[Outlook, Humidity, Temp]	3	0.125	0.153	
[Outlook, Humidity, Wind]	3	0.134	0.006	

$$M_S = \frac{k\overline{r_{cf}}}{\sqrt{k + k(k-1)\overline{r_{ff}}}}$$

	Outlook	Temp	Humidity	Wind	Play
Outlook	1.000	0.151	0.016	0.004	0.196
Temp		1.000	0.293	0.030	0.023
Humidity			1.000	0.000	0.156
Wind				1.000	0.050
Play					1.000

Subconjunto	k	$\overline{r_{cf}}$	$\overline{r_{ff}}$	Mérito (Ms)
[ ]	0	-	-	-
[Outlook]	1	0.196	1.000	0.196
[Temp]	1	0.023	1.000	0.023
[Humidity]	1	0.156	1.000	0.156
[Wind]	1	0.050	1.000	0.050
[Outlook, Temp]	2	0.109	0.151	0.143
[Outlook, Humidity]	**2**	**0.176**	**0.016**	**0.246**
[Outlook, Wind]	2	0.123	0.004	0.173
[Outlook, Humidity, Temp]	3	0.125	0.153	0.189
[Outlook, Humidity, Wind]	3	0.134	0.006	0.230

$$M_S = \frac{k\overline{r_{cf}}}{\sqrt{k + k(k-1)\overline{r_{ff}}}}$$

	Outlook	Temp	Humidity	Wind	Play
Outlook	1.000	0.151	0.016	0.004	0.196
Temp		1.000	0.293	0.030	0.023
Humidity			1.000	0.000	0.156
Wind				1.000	0.050
Play					1.000

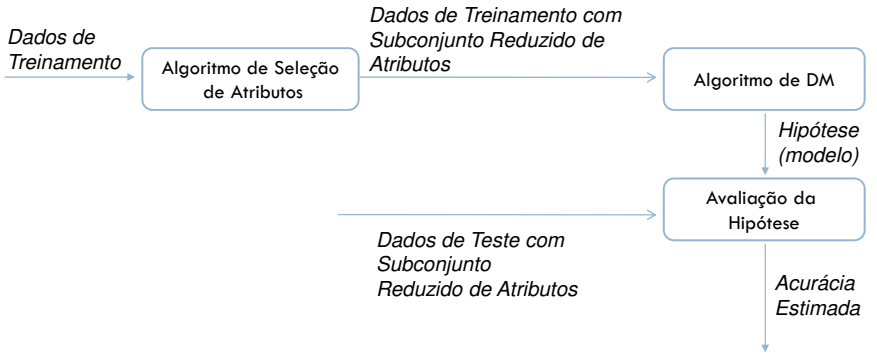
Exercício

- Tomando como base o exemplo anterior
  - Execute o algoritmo CFS, porém utilizando a estratégia de busca Backward Elimination
    - A busca começa com todos atributos
    - Um atributo é removido a cada passo

Comparação de Métodos de Seleção

- Procedimento de validação cruzada
  - Seleção de atributos somente no conjunto de treinamento
- Separar os dados em
  - Conjunto de treinamento
    - Realizar seleção de atributos
    - A partir do resultado obtido construir o modelo
  - Conjunto de teste
    - Avaliar o modelo (apenas atributos selecionados)

Comparação de Métodos de Seleção



Utilizar um dos procedimentos de estimação de erros vistos em aula...

Em Resumo

- Grande quantidade de atributos
  - Por exemplo, bases de expressão gênica:
    - eventualmente poucas centenas ou dezenas de objetos...
    - mas possivelmente milhares de atributos
- Podem implicar
  - Perda de desempenho de algoritmos de DM
  - Custo computacional elevado

Em Resumo

- A **seleção de atributos** assume que parte dos atributos existentes pode ser
  - Irrelevante
  - Redundante
- Podem comprometer a qualidade do processo de mineração de dados

Em Resumo

- Benefícios da Seleção de Atributos
  - Redução de
    - Custo computacional
    - Custo para obtenção de novos dados
    - Armazenamento de dados
  - Aumento da acurácia para os dados disponíveis
  - Melhor Interpretação dos modelos e resultados

Em Resumo

- Estratégias de Seleção
  - Ordenação ou ranking
    - Ordena os atributos segundo algum critério
    - Utiliza-se em conjunto com um método para selecionar os atributos de acordo com o ranking obtido
      - Tipicamente escolher os n_s primeiros atributos
  - Seleção de Subconjunto
    - Considera a interação entre os atributos para escolher o melhor subconjunto que for possível

Em Resumo

- Categorias de Métodos de Seleção
  - Filtros
    - Seleção de atributos é realizada a priori e não envolve o algoritmo de DM a ser aplicado depois (algoritmo alvo)
  - Wrappers
    - Seleção de atributos envolve o algoritmo de DM alvo, que é utilizado para guiar o processo de seleção
  - Embarcados
    - Seleção de atributos ocorre naturalmente e internamente como parte do algoritmo de DM

Filtros × Wrappers

- Vantagens dos Filtros
  - Como não depende de um algoritmo específico
    - Os atributos selecionados podem ser utilizados por diferentes algoritmos
  - Heurísticas para avaliar atributos geralmente possuem baixo custo computacional
    - Usualmente muito mais rápidos que Wrappers
    - Logo, conseguem lidar de forma eficiente com uma grande quantidade de atributos

Filtros × Wrappers

- Desvantagens dos Filtros
  - Ignora interação com o algoritmo de DM
    - Não leva em conta o viés (bias) indutivo do algoritmo
    - Logo, pode produzir modelos pouco eficientes
  - Alguns Filtros ignoram dependências entre atributos

Referências Complementares

- Fayyad, U.; Shapiro, G.; Smyth, P.; Uthurusamy, R.; (1996a); "*Preface*". In: *Advances in Knowledge Discovery and Data Mining*, Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P., Uthurusamy, R., Editors, MIT Press, pp. xiii-xiv.
- Guyon, I.; Elisseeff, A.; (2003); *An Introduction to Variable and Feature Selection*, Journal of Machine Learning Research 3, 1157-1182.
- Liu H. and Yu L., *Toward Integrating Feature Selection Algorithms for Classification and Clustering*, IEEE Transactions on Knowledge and Data Engineering, vol. 17, no. 4, 491-502, 2005.
- Blum A. and Langley P., *Selection of relevant features and examples in machine learning*. *Artificial Intelligence,* 97(1-2):245–271, December 1997.
- Guyon I., *Practical Feature Selection: from Correlation to Causality, In: Mining Massive Data Sets for Security . IOS Press, 2008.*
- Kononenko, I., *Estimating Attributes: Analysis and Extensions of RELIEF*, Proc. of European Conf. on Machine Learning, pp171-182, 1994.
- Reunanen, J., *Overfitting in Making Comparisons Between Variable Selection Methods*, Journal of Machine Learning Research (3), pp. 1371-1382, 2003.