Implantação e Avaliação de um Algoritmo de Ranking Baseado em Classificação

Igor Giusti

26 de fevereiro de 2012

Sumário

Conceito e Exemplos

- Rank significa uma posição particular, mais alta ou mais baixa que outras (dicionário Cambridge);
- Ranking é uma coleção dessas posições;
- Exemplos:
  - Ranking das seleções de futebol masculino da FIFA;
  - Ranking do IDH dos países (Índice de Desenvolvimento Humano);
  - Ranking da competição da Yahoo! sobre Learning to Rank.

Características de um Ranking

- Rankings são ordenados de acordo com critérios. Sobre esses critérios define-se um modelo, encarregado de impor uma ordem total ou parcial sobre os elementos do ranking;
- Características relevantes de um modelo:
  - Computabilidade;
  - Potencial de Automatização.

Problema Proposto

Problema

Dado um conjunto de documentos em que cada documento pode receber um rótulo, atribuir uma posição a cada elemento do conjunto tendo como insumo pares documento-rótulo ou uma relação de ordem total ou parcial entre documentos;

- Área de pesquisa chamada de Learning to Rank;
- Técnica de ranking reduzido a classificação é uma solução possível.

Classificação e Ranking

Classificação Tarefa de atribuir rótulos a cada elemento de um dado conjunto tendo como entrada pares elemento-rótulo.

Ranking Tarefa de atribuir posições a cada elemento de um dado conjunto tendo como entrada pares elemento-rótulo, uma relação de ordem parcial entre os elementos, ou uma relação de ordem total entre os elementos.

- Algoritmos de classificação e ranking podem compartilhar o mesmo tipo de entrada, pares elemento-rótulo;
- Técnica de Ranking Reduzido a Classificação propõe compor um algoritmo de ranking a partir de um algoritmo de classificação.

Descrição do Problema

Problema

Dado um conjunto composto por elementos aos quais é possível atribuir uma classe de valor 0 ou 1, deseja-se encontrar uma permutação dos elementos de maneira que os elementos que apresentem maior chance de pertencer a classe 0 devem preceder os com maior chance de receber o rótulo 1.

- Solução trivial, permutar os elementos do conjunto com base nas probabilidades das previsões de um classificador;
- Solução trivial pode implicar em erro alto na ordenação dependendo da performance do classificador;
- Técnica de Ranking Reduzido a Classificação é uma alternativa.

Ranking Reduzido à Clasificação

- Possui um limite teórico máximo de erro bastante reduzido se comparado à solução trivial;
- Dividido em etapas de treinamento e ordenação;
- Treinamento efetuado pelo algoritmo AUC-Train;
- Ordenação efetuada pelo algoritmo Tournament;

Definições

Base

Uma base B é um conjunto de instâncias com cardinalidade n.

$$B = \{i_1, ..., i_n\} \qquad |B| = n$$

Instância

Uma instância I é uma tupla $\langle A, C \rangle$ na qual A é um vetor de atributos com cardinalidade m e C é a classe da instância e pode valer 0 ou 1.

$$I = \langle A, C \rangle \qquad A = (a_1, ..., a_m) \qquad C = x | x \in \{0, 1\}$$

- Pode-se acessar os atributos de uma instância I através de I(A) e classe através de I(C);
- Vetores de atributos podem ser concatenados através do operador binário $||$.

Exemplos

Base

panorama	temperatura	humidade	ventoso	adequado
ensolarado	quente	alta	falso	não
ensolarado	quente	alta	verdadeiro	não
nublado	quente	alta	falso	sim
chuvoso	branda	alta	falso	sim
chuvoso	frio	normal	falso	sim
chuvoso	frio	normal	verdadeiro	não
nublado	frio	normal	verdadeiro	sim
ensolarado	branda	alta	falso	não

Instância

- $I = \langle(\textit{ensolarado}, \textit{quente}, \textit{alta}, \textit{falso}), \textit{não}\rangle$
- $I(A) = (\textit{ensolarado}, \textit{quente}, \textit{alta}, \textit{falso})$
- $I(C) = \textit{não}$

Medidas de Desempenho

Acurácia

- Razão entre o número de instâncias classificadas corretamente e o total de instâncias em uma base;
- Erros de classificação afetam pontualmente;
- Medida padrão para desempenho de classificadores.

AUC

- Considera a relação entre as instâncias, em vez de uma simples razão como a acurácia;
- Erros de classificação podem afetar a AUC mais intensamente;
- Medida proposta para a avaliação de ranking.

AUC-Train

Algoritmo 1: AUC-Train

$S' = \{\langle (x_1, x_2), 1(y_1 < y_2)\rangle \ : \ (x_1, y_1), (x_2, y_2) \in S \text{ and } y_1 \neq y_2;$
return $c = A(S')$

Decomposição

1. Particionar a base de treinamento em duas bases S_0 e S_1; S_0 possui as instâncias de classe 0 e S_1 possui as instâncias de classe 1;
2. Combinar as partições S_0 e S_1 de forma que a base de treinamento S' possua todos os pares entre instâncias de classe 1 e de classe 0;
3. Aplicar um algoritmo de aprendizagem A à base S' obtendo um classificador c como saída.

AUC-Train: Particionamento

Função particionar(S)

$S_0, S_1 \leftarrow \emptyset$;
para todo $i \in S$ **faça**
  se $i(C) = 0$ **então**
    $S_0 \leftarrow S_0 \cup \{i\}$;
  senão
    $S_1 \leftarrow S_1 \cup \{i\}$;
  fim
fim
retorna S_0, S_1

Complexidade

- $O(n)$

AUC-Train: Combinação

Função combinar(S_α, S_β)

$S_C \leftarrow \emptyset$;
para todo $\alpha \in S_\alpha$ **faça**
  para todo $\beta \in S_\beta$ **faça**
    $S_C \leftarrow S_C \cup \{mesclar(\alpha, \beta)\}$;
  fim
fim
retorna S_C

Função mesclar(α, β)

retorna $\langle \alpha(A) || \beta(A), 1 \cdot (\alpha(C), \beta(C)) \rangle$

Complexidade

- Melhor caso: $O(n)$
- Caso médio e pior caso: $O(n^2)$

AUC-Train: Otimizações

Desvantagem

O custo computacional adicionados pelas etapa de particionamento e principalmente pela etapa de combinação é alto.

Sugestões de Otimização

- Votação
- Amostragem

Há uma complementaridade entre as duas estratégias escolhidas para otimização do algoritmo AUC-Train. Enquanto uma preza por melhorar o tempo de execução, o outra preza por dar maior segurança na classificação.

AUC-Train: Otimização - Amostragem

Função amostragem(S_α, S_β, p)

$S \leftarrow \emptyset$;
para todo $\alpha \in S_\alpha$ **faça**
  para todo $\beta \in Ss_\beta$ *tal que* $Ss_\beta \subseteq S_\beta \wedge |Ss_\beta| = p$ **faça**
    $S \leftarrow S \cup \{mesclar(\alpha, \beta)\}$;
  fim
fim
retorna S

Algoritmo de Treinamento

Algoritmo 2: Treinamento

$C \leftarrow \emptyset$;
$S_0, S_1 \leftarrow particionar(S)$;
se $i = 1$ *e* $p = todas$ **então**
    $S' \leftarrow combinar(S_0, S_1, mesclar)$;
    $S' \leftarrow combinar(S_1, S_0, mesclar)$;
    $C \leftarrow \{A(S')\}$;
fim
senão se $i > 1$ *e* $0 < p \leq min(|S_0|, |S_1|)$ **então**
    para $i \leftarrow 1; i \rightarrow n$ **faça**
        $S' \leftarrow amostragem(S_0, S_1, p, mesclar)$;
        $S' \leftarrow S' \cup amostragem(S_1, S_0, p, mesclar)$;
        $C \leftarrow C \cup \{A(S')\}$;
    fim
fim
retorna C

Tournament

Algoritmo 3: Tournament

For $x \in U$, let $deg(x) = |\{x' : c(x, x') = 1, x' \in U\}|$;
Sort U in descending order of deg(x), breaking ties arbitrarily

Decomposição

1. Obter a pontuação para todas as instâncias na base B;
2. Ordenar as instâncias na base B.

Tournament: Efeito Colateral - Votação

Função votacao(C, i)

```
classe ← 0;
zero ← 0;
para todo c ∈ C faça
    se c(i) = 0 então
        zero ← zero + 1;
    senão
        zero ← zero − 1;
    fim
fim
se zero < 0 então
    classe ← 1;
fim
retorna classe
```

Algoritmo de Ordenação

Algoritmo 4: Ordenação

para todo $\alpha \in B$ **faça**
  para todo $\beta \in B \wedge \beta \neq \alpha$ **faça**
    $i \leftarrow \langle \alpha(A) || \beta(A) \rangle$;
    $classe \leftarrow votacao(C, i)$
    se $classe = 1$ **então**
      $pontuacao[\alpha] \leftarrow pontuacao[\alpha] + 1$
    senão
      $pontuacao[\beta] \leftarrow pontuacao[\beta] + 1$
    fim
  fim
fim
ordenar B com base em *pontuacao*

Complexidade

- $O(n^2)$ chamadas à função votacao + $O(f_{sort}(n))$;

Premissas

- Um classificador que apresente erro de α na acurácia, pode apresentar um erro máximo de $\alpha \cdot n$ na ordenação onde n é o tamanho da base a ser ordenada;
- Para o mesmo classificador com erro de α na acurácia, a técnica de Ranking Reduzido a Classificação reduz o erro máximo na ordenação para $\alpha \cdot 2$;
- Quanto maior o desbalanceamento entre as classes em uma base, maior a chance de intensificação do erro na ordenação.

Características das Bases

Observações

- Foram escolhidas bases com diferentes níveis de desbalanceamento a fim de verificar as premissas.
- Algumas bases que tratavam originalmente de problemas multiclasse precisaram ser convertidas para bases binárias.

Bases	Classe		
	Minoritária	Majoritária	Distribuição
breast-cancer	85	201	30% - 70%
vehicle	199	647	23% - 77%
hepatitis	32	123	20% - 80%
glass	29	185	13% - 87%
yeast	20	463	4% - 96%

Tabela: Dados sobre as bases usadas para *ranking*

Classificadores Avaliados

- Árvore de Decisão C4.5 (trees.J48)
- Naïve Bayes (bayes.NaiveBayes)
- Curva Logística (functions.Logistic)
- Support Vector Machine (functions.SMO)

Estratégia de avaliação

Os testes executaram através de validação cruzada com 10 partições nas seguintes configurações:

1. Somente o classificador;
2. O classificador como base para a técnica de *ranking reduzido a classificação* original;
3. O classificador como base para o algoritmo de ranking com configurações de 1 par por instância e variando o número de classificadores na votação entre 1 e 20;
4. O classificador como base para o algoritmo de ranking com configurações de 1 classificador na votação e variando o número de pares por instância entre 1 e 20.

Desempenho: Árvore de decisão C4.5 (trees.J48)

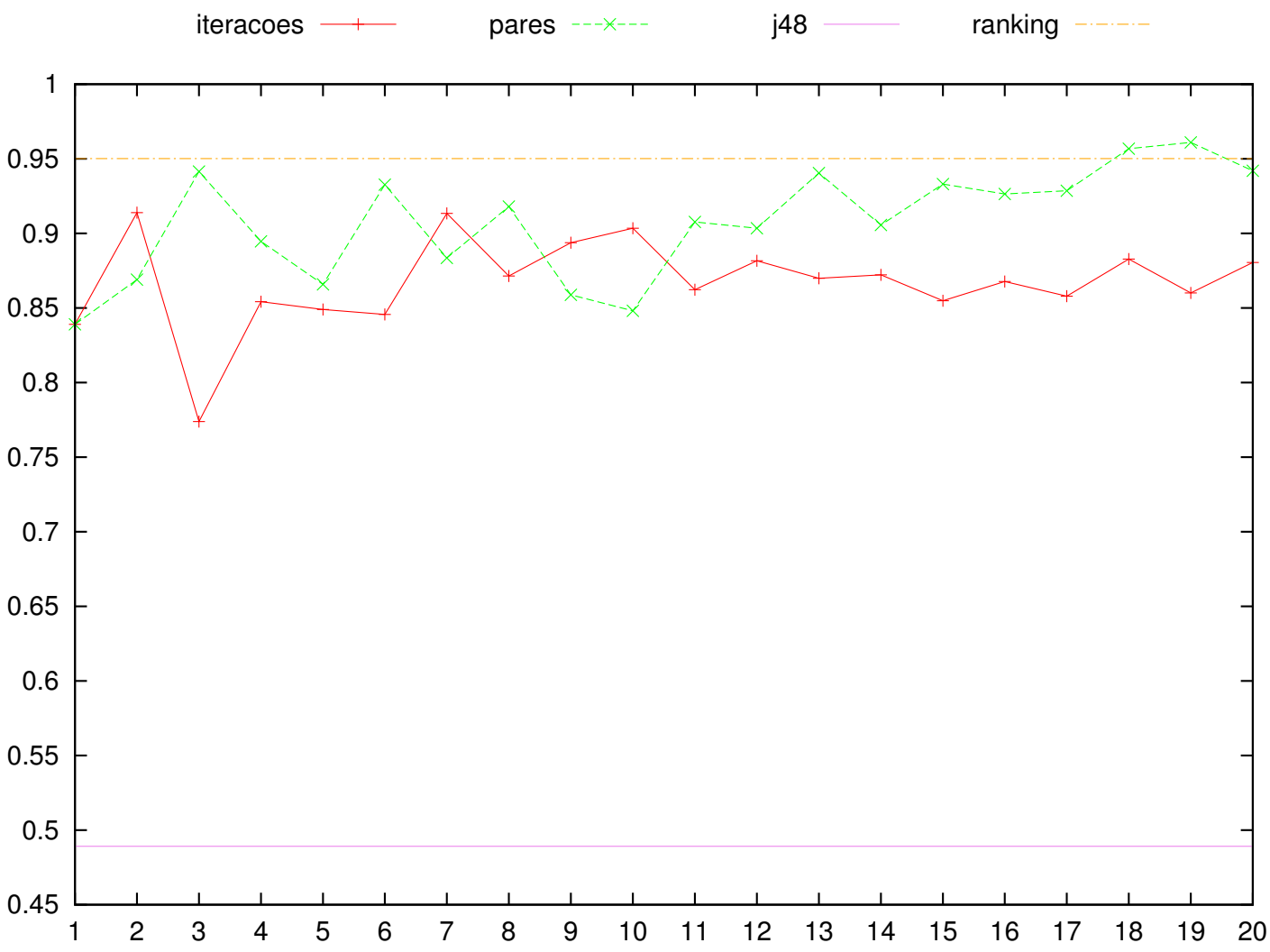


Figura: Gráfico de desempenho para a base Yeast

Desempenho: Naïve Bayes (bayes.NaiveBayes)

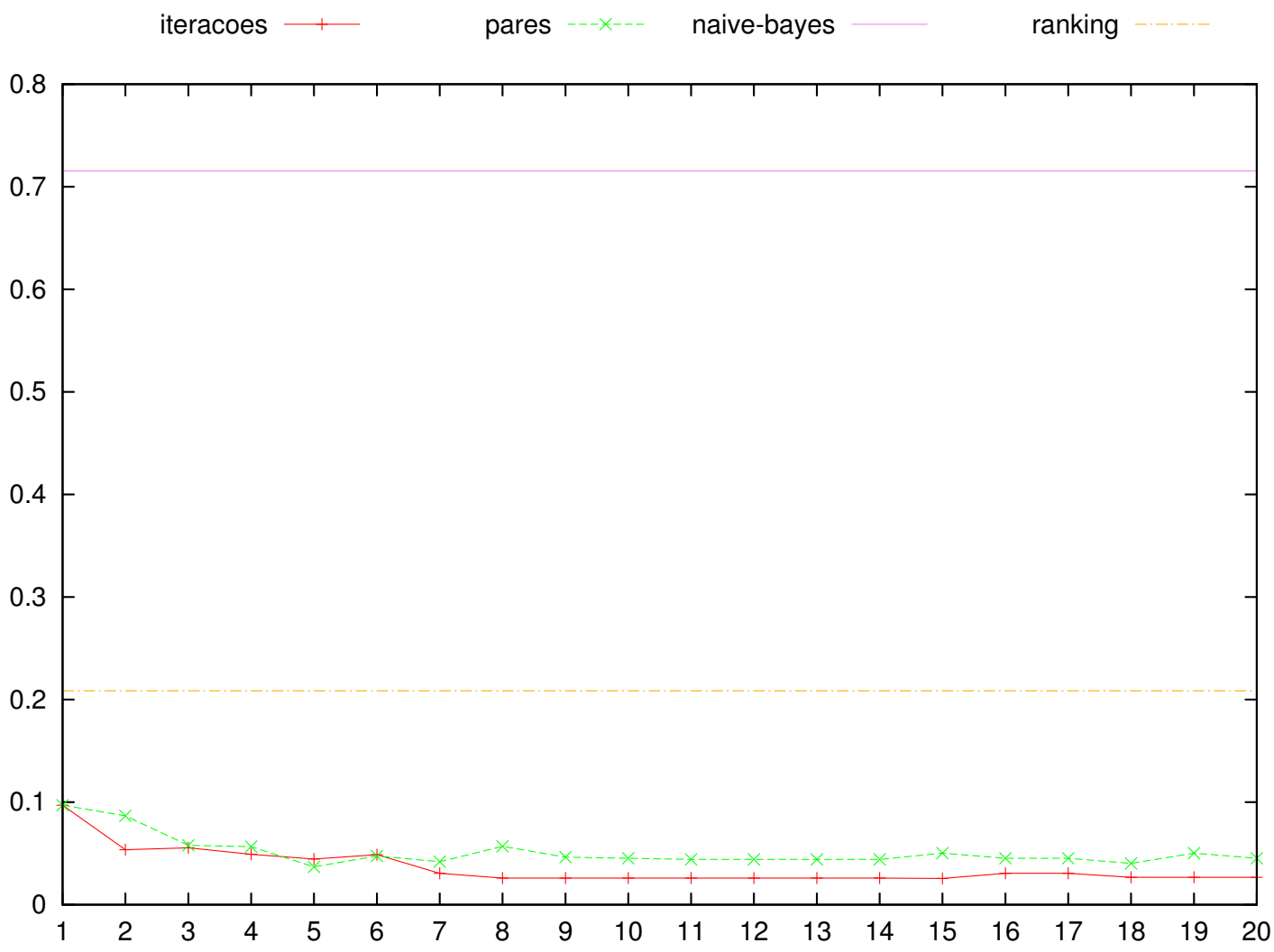


Figura: Gráfico de desempenho para a base Breast Cancer

Desempenho: Naïve Bayes (bayes.NaiveBayes)

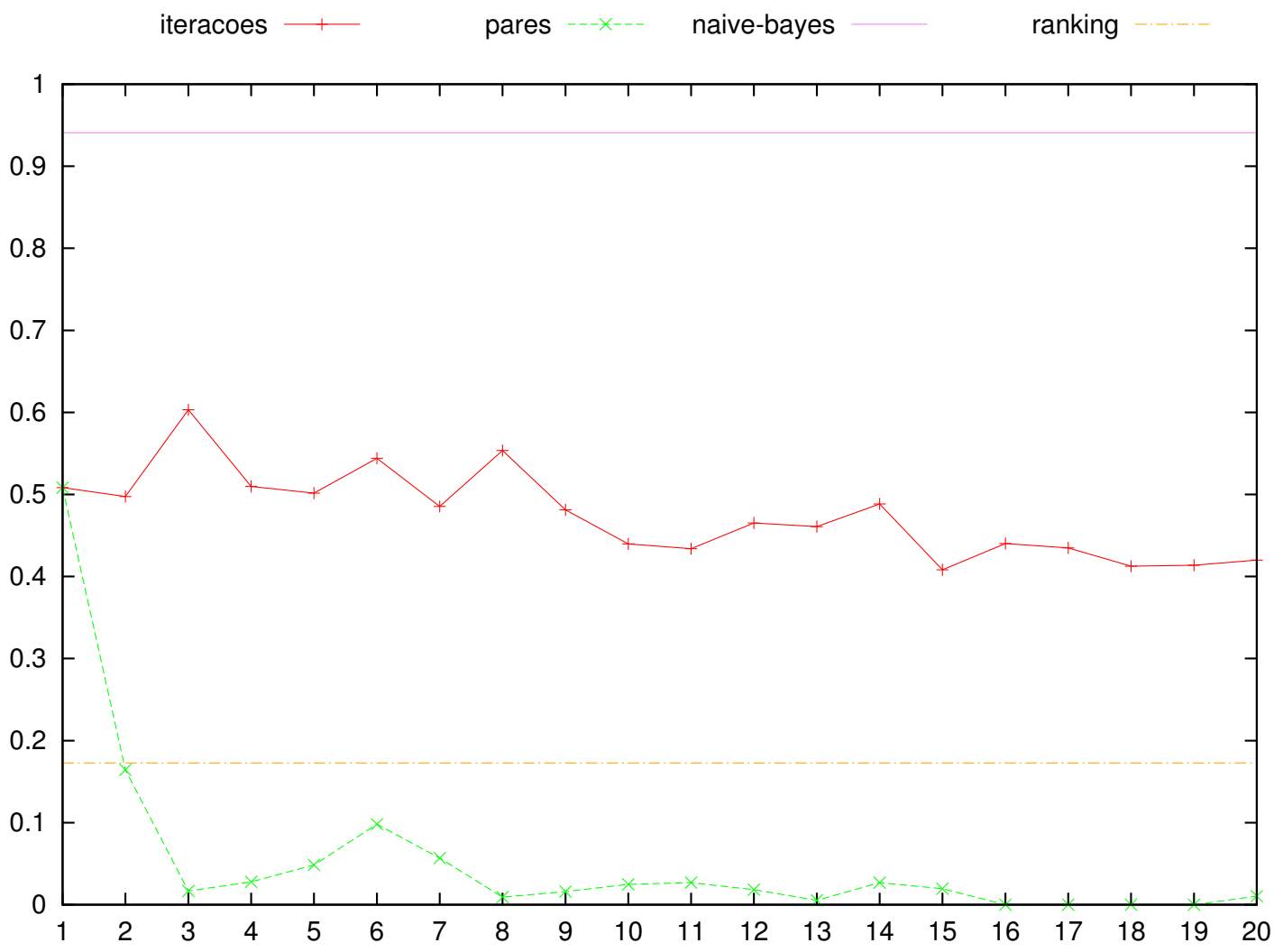


Figura: Gráfico de desempenho para a base Glass

Desempenho: Naïve Bayes (bayes.NaiveBayes)

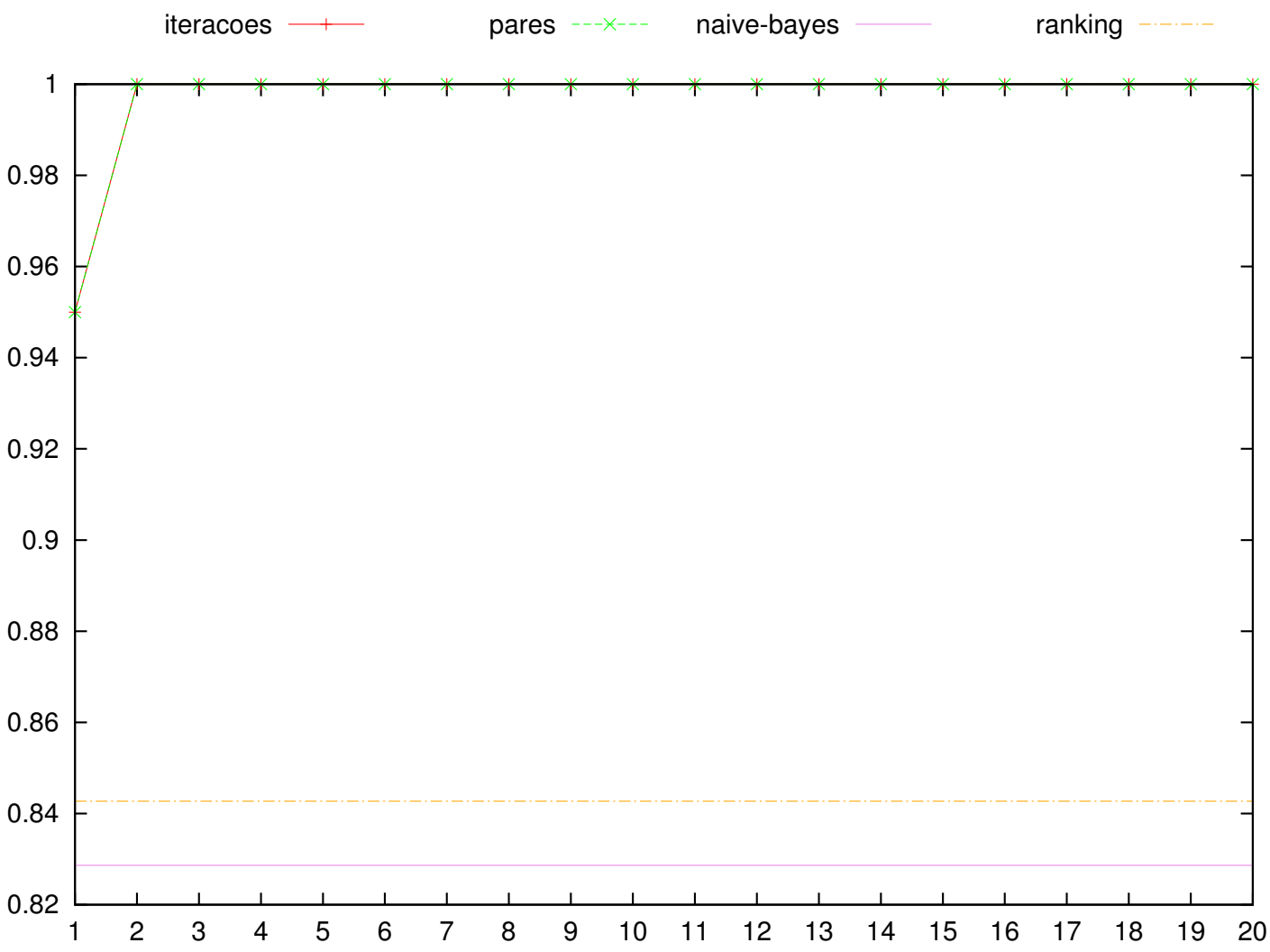


Figura: Gráfico de desempenho para a base Yeast

Desempenho: Curva Logística (functions.Logistic)

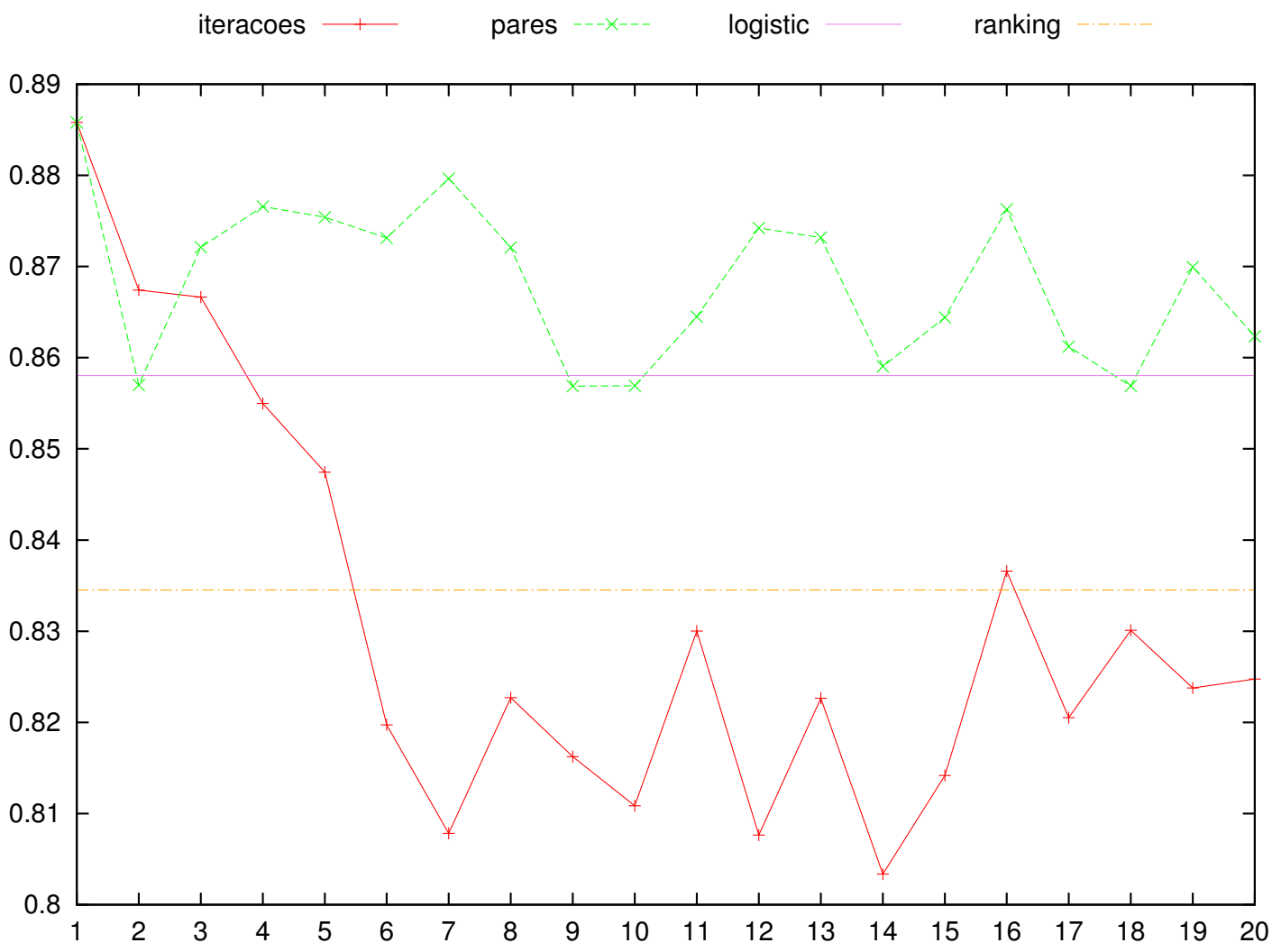


Figura: Gráfico de desempenho para a base Yeast

Desempenho: Support Vector Machine (functions.SMO)

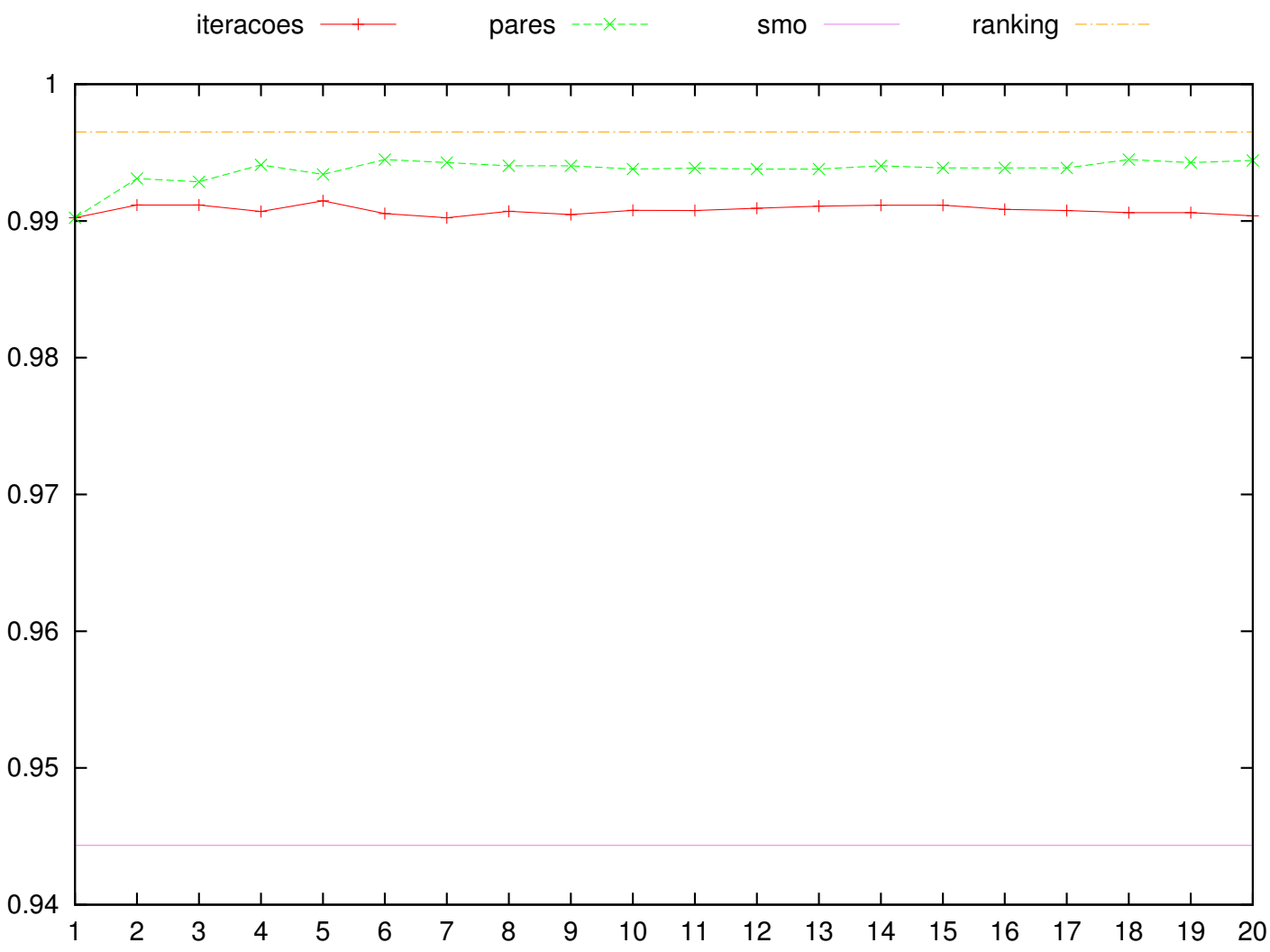


Figura: Gráficos de desempenho para a base Vehicle

Desempenho: Support Vector Machine (functions.SMO)

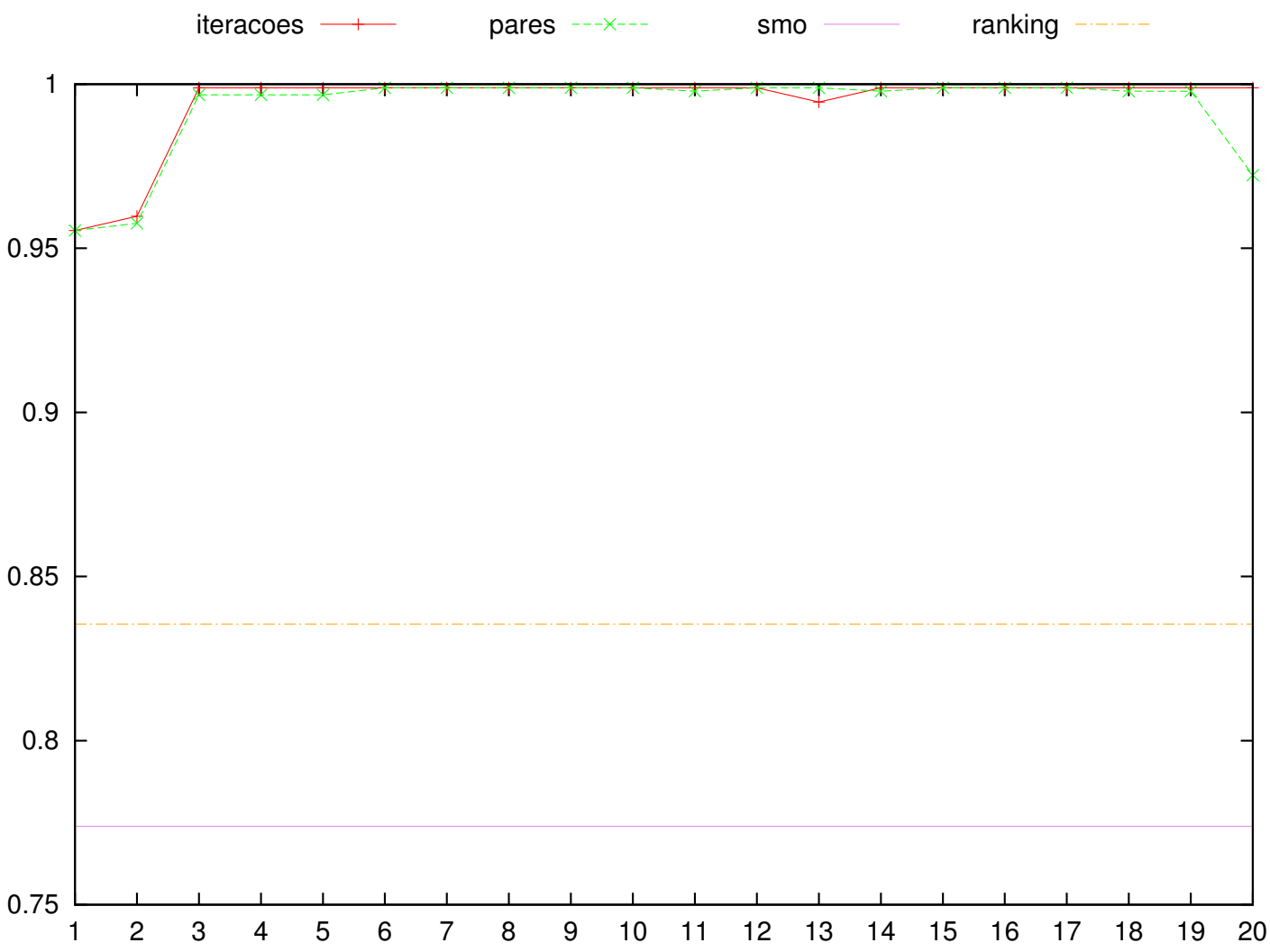


Figura: Gráficos de desempenho para a base Vehicle

Estratégias de Otimização do Treinamento

Votação

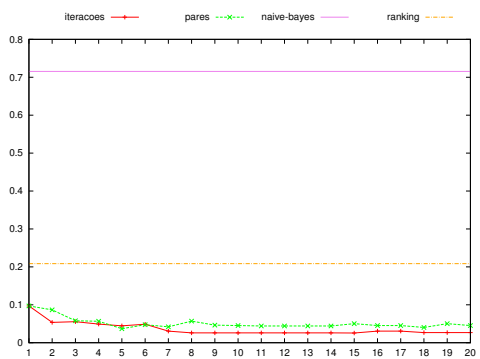
- Resultados com menor variação de AUC;
- Execuções pouco mais longas;

Pares por instância

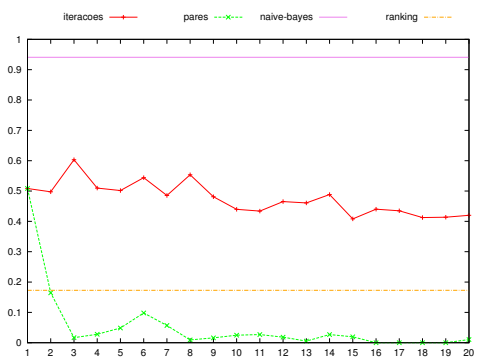
- Resultados com maior variação de AUC;
- Restrição do tempo de execução;

Naïve Bayes

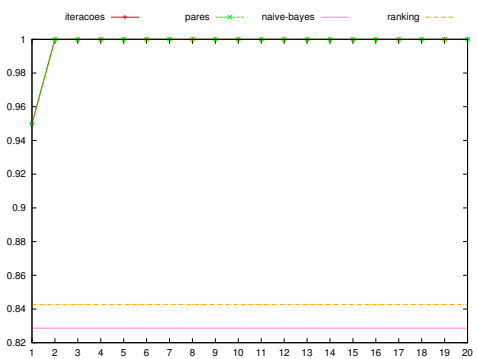
- Resultados péssimos para as bases: Breast-Cancer, Hepatitis e Vehicle;
- Resultado ruim para a base: Glass;
- Resultado excelente para a base Yeast (Principalmente usando votação e amostragem);
- Comportamento bipolar não esclarecido.



(a) Breast cancer



(b) Glass

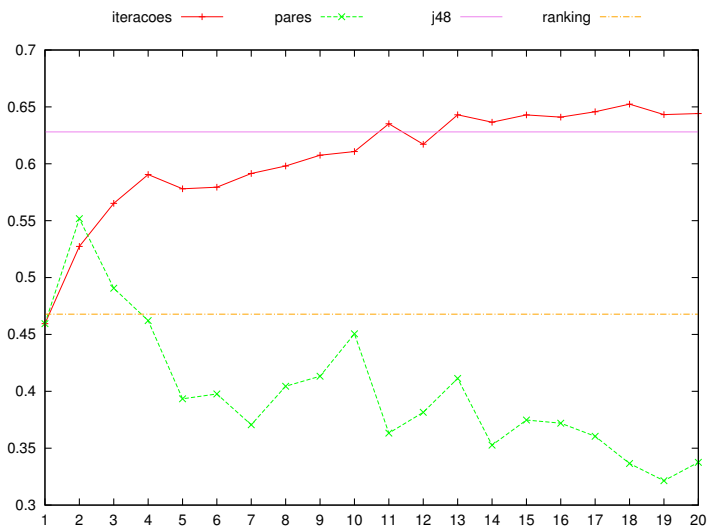


(c) Yeast

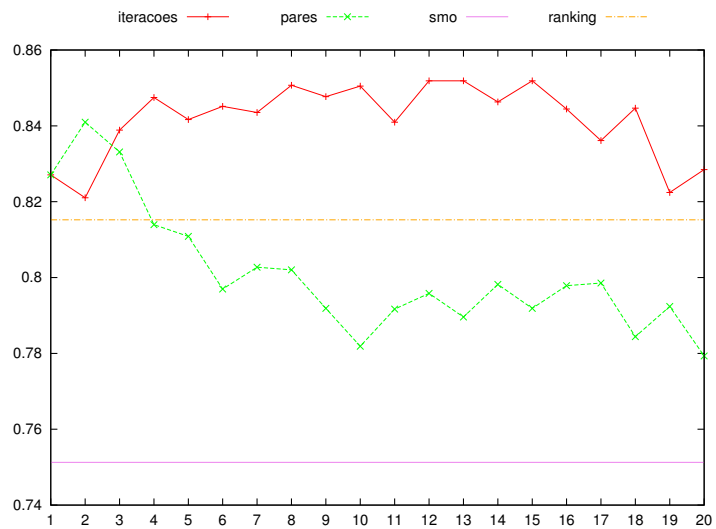
Figura: Gráficos de desempenho para Naïve Bayes

Resultados da Técnica de Ranking Reduzido a Classificação

- Em pouco mais da metade dos casos, a técnica teve melhores resultados sobre o classificador solo;
- Com a árvore de decisão C4.5 e a base Yeast, a técnica obteve uma melhora de aproximadamente 100% sobre o classificador solo;
- A otimização de votação produziu resultados melhores que o classificador solo na maioria dos casos.



(a) C4.5 com a base Yeast



(b) SVM com a base Hepatitis

Figura: Gráficos de desempenho