Modelos Múltiplos

João Gama
jgama@liacc.up.pt

Modelos Múltiplos

- Diferentes algoritmos de aprendizagem exploram:
  - Diferentes linguagens de representação.
  - Diferentes espaços de procura.
  - Diferentes funções de avaliação de hipóteses.
- Como poderemos explorar estas diferenças ?
  - Será possível obter um conjunto de classificadores cuja performance é melhor que a performance de cada classificador individual ?
- Observação:
  - Não existe um algoritmo que seja o melhor para todos os problemas
    - Resultados experimentais: Projecto Statlog
    - Resultados Teóricos: “No free lunch”

Erro Correlacionado

- Uma condição necessária:
  - *Um conjunto melhora sobre os classificadores individuais se estes discordam entre si.* Hansen & Salamon - 1990
- O *erro correlacionado* é uma métrica da diversidade entre as predições de dois algoritmos.
- *Erro correlacionado*:
  - Probabilidade de dois classificadores cometerem o mesmo erro dado que um deles comete um erro.

$$\phi_{i,j} = p(\hat{f}_i(x) = \hat{f}_j(x) \mid \hat{f}_i(x) \neq f(x) \vee \hat{f}_j(x) \neq f(x))$$

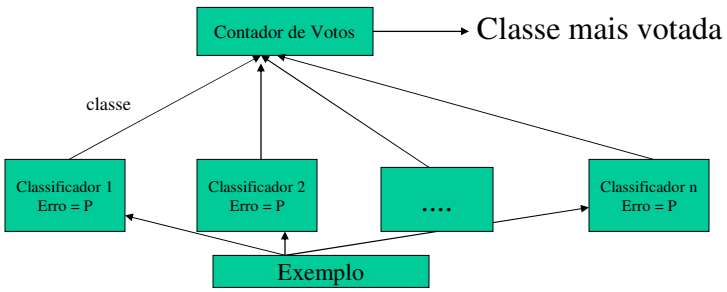
Observado	0	0	0	1	0	1	1	0
Algoritmo A	1	1	1	1	1	1	0	1
Algoritmo B	0	1	1	0	1	1	0	0

$$\phi_{A,B} = 4/7 = 0.57$$

- Será uma condição suficiente?

Modelos Múltiplos

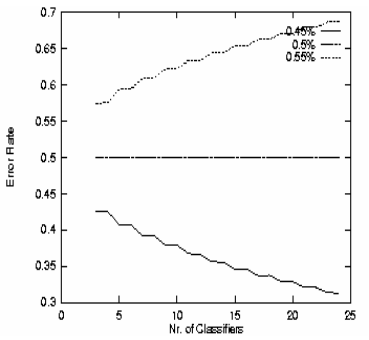
- Um estudo em simulação:
  - Considere um problema de duas classes equi-prováveis:
    - $P(Classe_1) = P(Classe_2)$
  - Numero de classificadores:[3..25]
    - Com a mesma probabilidade de cometer erros.
    - $P\_erro(Classificador_i) = \{0.45;0.5;0.55\}$
  - O modelo múltiplo é obtido por agregação dos vários classificadores
    - As predições dos classificadores são agregadas por votação uniforme.



4

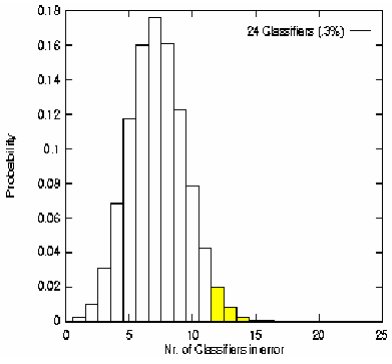
Modelos Múltiplos – Uma Simulação

- Estudo da variação do erro de um conjunto de classificadores variando o numero de classificadores agregados:
  - A probabilidade de erro de cada classificador é:
    - P = 0.5 (escolha aleatória de uma das classes)
      - A probabilidade de erro do conjunto é constante: 0.5
    - P > 0.5
      - A probabilidade de erro do conjunto cresce linearmente com o numero de classificadores
    - P < 0.5
      - A probabilidade de erro do conjunto diminui linearmente com o numero de classificadores
- Uma condição necessária:
  - A taxa de erro de um conjunto de classificadores diminui em relação à taxa de erro dos classificadores individuais se:
    - Cada classificador individual do conjunto tiver uma performance melhor que uma escolha aleatória.



Modelos Múltiplos – Uma Simulação

- Considere um modelo múltiplo obtido agregando por votação uniforme:
  - 23 classificadores.
  - A probabilidade de erro de cada classificador é 30%.
- Dado exemplo a classificar
  - O modelo múltiplo classifica o exemplo incorrectamente se e só se:
    - 12 ou mais classificadores classificam o exemplo incorrectamente.
  - A probabilidade do modelo múltiplo errar é dada pela área sob a curva da distribuição binomial.
    - No caso em estudo a área é de 0.026.
    - Muito menor que o erro de cada classificador.



Condições Necessárias

- "To achieve higher accuracy the models should be diverse and each model must be quite accurate"
  Ali & Pazzani 96

- Condições Necessárias
  - Os classificadores devem ter uma performance melhor que uma escolha aleatória (*random guess*)
  - Os classificadores devem cometer erros não correlacionados.
    - Diferentes tipos de erros.
    - Erros em diferentes regiões do espaço.

Um modelo Teórico: Bayesian Model Averaging

- Dados:
  - Um conjunto de treino $D = \{< \vec{x}_i, c_i >, i = 1,...,n\}$
  - Um conjunto de modelos H={h_1,...., h_t}
- Dado um exemplo não classificado x,
  - a média Bayesiana de modelos classifica **x**, na classe que maximiza:

$$P(c \mid x, D, H) \propto \sum_{h \in H} P(c \mid x, h) P(h \mid D)$$

    - P(c | x, h) –
      - probabilidade de o modelo *h* classificar na classe ***c*** o exemplo ***x***.
    - A verosimilhança dos dados dado o modelo *P(h|D)*.

Média Bayesiana de Modelos

- Verosimilhança:
  - Pelo teorema de Bayes $P(h \mid D) = \frac{P(h)}{P(D)} \prod_{i=1}^{n} p(\vec{x}_i, c_i \mid h)$
    - Assumindo que os exemplos são independentes.
  - $p(x_i, c_i \mid h) = p(x_i \mid h)\, p(c_i \mid x_i, h)$
- O modelo uniforme de ruído da classe:
  - A classe associada a cada exemplo está errada com probabilidade ε
    - $P(c_i \mid x_i, h) = 1-\varepsilon$ se h classifica x_i correctamente na classe c_i
    - $P(c_i \mid x_i, h) = \varepsilon$ se h classifica incorrectamente x_i.

$$P(h \mid D) \propto P(h)(1-\varepsilon)^{s} \varepsilon^{n-s}$$

    - Onde s representa o numero de exemplos correctamente classificados
    - O nível de ruído pode ser estimado pela média do erro dos modelos.

Média Bayesiana de Modelos

- Em problemas complexos H não é enumerável.
  - P(h) é difícil de estimar.
- Critica (ver P.Domingos, T. Dietterich):
  - Para conjuntos de treino pequenos (em relação ao espaço das hipótese H)
    - Muitas hipóteses terão P(h | D) aproximados
    - O conjunto funciona
  - Para conjuntos de treino grandes
    - Tipicamente uma hipótese um elevado p(h | D)
    - O conjunto reduz-se a uma hipótese.

Podem os Modelos Múltiplos funcionar na pratica?

- Um algoritmo de aprendizagem efectua uma procura num espaço de hipóteses H.
- A escolha de um único modelo tem vários problemas:
  - Estatísticos
    - O volume de dados é pequeno em relação ao espaço das hipóteses.
    - Decisões sem suporte estatístico.
  - Computacionais
    - Procura heurística.
    - Máximos Locais
  - Representação
    - A função que governa o fenómeno não está em H.
- A utilização de modelos múltiplos pode minimizar qualquer um destes problemas.

Modelos Múltiplos

- Combinação de predições
  - Votação Uniforme
  - Votação Pesada
  - Soma de distribuições
- Gerar Modelos
  - Modelos Homogéneos
    - Bagging
    - Boosting

Combinação de Predições

- Combinação de predições
  - Votação uniforme
    - Para classificar um exemplo, cada classificador vota numa classe
    - Um exemplo é classificado na classe com maior numero de votos.
  - Votação pesada
    - A predição de cada classificador é pesada por uma estimativa "a priori" da qualidade do classificador.
- Vantagens
  - Simplicidade
  - Aplicável em inúmeras situações
- Desvantagens
  - Não tem em conta o exemplo a classificar
  - Não faz selecção dos classificadores

Fusão de Classificadores

- Para combinar algoritmos que retornam uma distribuição de probabilidades na classificação de um exemplo
  - Soma de distribuições
  - Média aritmética
  - Produto
  - Média geométrica
  - Máximo
  - Mínimo

Um problema com j classes
Combinar as predições de m algoritmos

$$P_j = \sum_{k=1}^{m} p_{k,j}$$

$$P_j = \sum_{k=1}^{m} \frac{p_{k,j}}{m}$$

$$P_j = \prod_{k=1}^{m} p_{k,j}$$

$$P_j = \sqrt[m]{\prod_{k=1}^{m} p_{k,j}}$$

$$P_j = \max_k (p_{k,j})$$

$$P_j = \min_k (p_{k,j})$$

–Classifying a test example
»Suppose the outputs:
»Cl1 (0.9,0.1),
»Cl2 (1,0),
»Cl3 (0.6,0.4)
»The Sum Rule:
»(2.5, 0.5)
»Final Prediction
»(0.83,0.17)

Combinação de predições

- Selecção de Classificadores:
  - "Model applicability induction"
    - Ortega, 95
  - Caracteriza regiões do espaço dos atributos onde cada modelo faz predições correctas.

- Tendo em conta o exemplo a classificar:
  - "Composite Learner"
    - Ting, 97
  - Para cada exemplo de teste escolhe o classificador com maior confiança na sua própria predição.

Combinação de Modelos Homogéneos

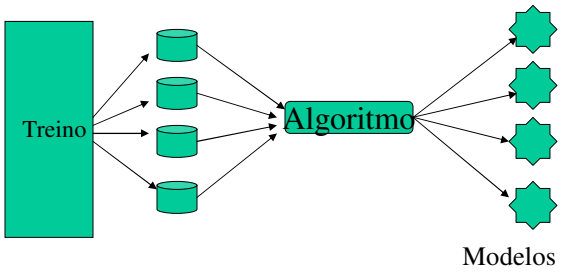
1. Bagging (Boostrap Aggregation)
2. Ada-Boosting (Adaptive boosting)

Bagging

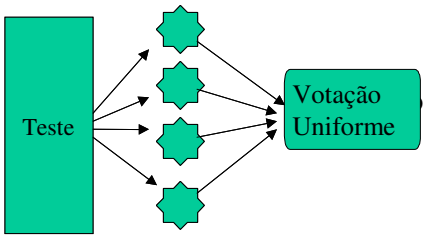
- **Aprendizagem:**
  - Obter *N* réplicas, com reposição, do conjunto de treino.
    - As amostras têm o mesmo numero de exemplos do conjunto de treino.
    - É usual usar 25 amostras.
  - Para cada amostra gerar um classificador.
- **Aplicação**
  - Para cada exemplo de teste
    - Determina a classe predita por cada classificador.
    - As predições são agregadas por voto uniforme.
      - O exemplo é classificado na classe mais votada.

Bagging

- **Aprendizagem:**

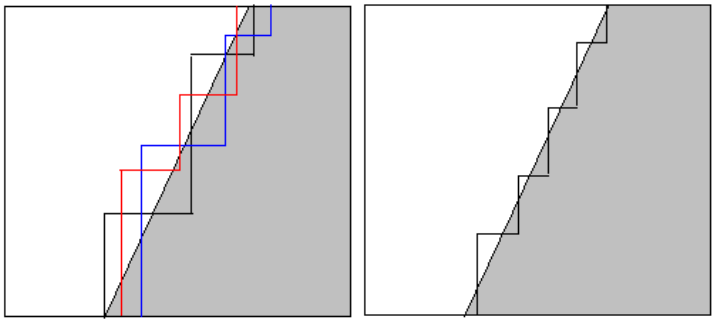


- **Aplicação**



Porque Funciona ?

- Escolhendo o voto maioritário sobre muitos modelos, reduz a variabilidade aleatória dos modelos individuais.
  - Por exemplo, em arvores de decisão
    - A escolha do atributo de teste para um nó
    - A escolha dos pontos de referência nos atributos reais.



Bagging

- Características
  - Requer Algoritmos instáveis.
    - Algoritmos sensíveis a pequenas variações do conjunto de treino.
    - Árvores de Decisão, Redes Neuronais
  - Fácil de implementar com qualquer algoritmo.
  - Fácil de implementar em ambientes paralelos.

- A redução de erro observada é devida á redução na componente da variância. (Breiman 92)

Boosting

- O problema
  - Existirá um algoritmo tal que:
    - Dados:
      - Um nível de confiança: δ ($0 < \delta < 0.5$) e
      - Um limite para o erro: ε ($0 < \varepsilon < 0.5$)
    - O algoritmo gere uma hipótese *h* tal que
      - Com probabilidade 1-δ
      - O $\text{erro}_D(h) < \varepsilon$
      - Para qualquer distribuição D dos exemplos?
- *Boosting* é um algoritmo que satisfaz estas condições.

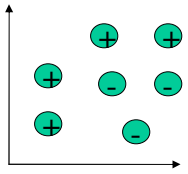
Boosting

- Aprendizagem
  - É um algoritmo iterativo.
  - Associa um peso a cada exemplo.
- Algoritmo:
  - Inicializa o peso de cada exemplo de forma uniforme
  - Iterativamente
    - Gera um classificador usando a actual distribuição dos exemplos.
      - A distribuição é dada pelos pesos
    - Os pesos dos exemplos incorrectamente classificados são incrementados para a iteração seguinte.
  - Os classificadores gerados são agregados por votação pesada.
- Aplicável a qualquer algoritmo de aprendizagem,
  - O algoritmo deverá ser capaz de gerar hipóteses ligeiramente melhores que uma escolha aleatória (*weak learner*).

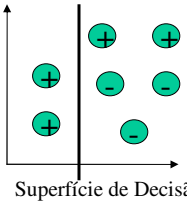
Boosting – Um Exemplo

Weak learner – gera um hiper-plano perpendicular a um dos eixos.

1ª Iteração

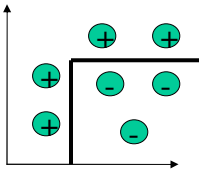


Conjunto de Treino

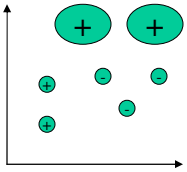


Superfície de Decisão

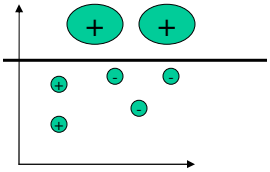
Composição dos 2 classificadores



2ª Iteração



Conjunto de Treino



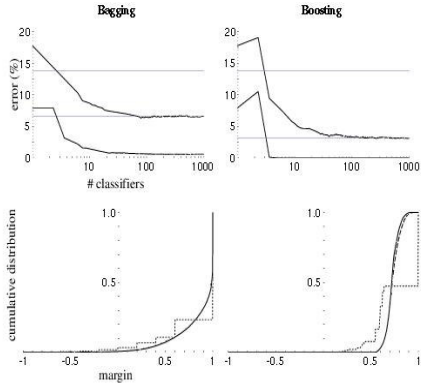
Superfície de Decisão

AdaBoosting

- Input:
  - Conjunto de Dados D,
  - Algoritmo Alg,
  - Nr. de Iterações L_{max}
- Inicializa $w_i = 1/m$ (*i* exemplo, *m* nr. de exemplos)
- Para L=1 até L_{max}
  - $h_L = Alg(D_w)$
  - E_L = Erro de h_L
  - Se $E_L > 0.5$ Ignora h_L
  - Senão
    - $B_L = E_L/(1-E_L)$
    - Para cada i
      - $w_{L+1}(i) = w_L(i)B_L^{1-[h_L(xi) <> yi]}$
- Output
  - $H_f(x) = \text{argmax}_y \text{Sum}_L(\log(1/B_L)[h_L(x)=y]$

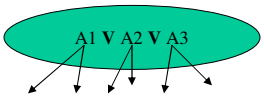
Comparação entre *Bagging* e *Boosting*

- *Bagging*
  - Redução do erro devida á variância.
  - Efectivo com classificadores instáveis
    - Não são reportados exemplos de degradação do erro.
- *Boosting*
  - Redução do erro quer na variância quer no *bias*.
  - Em problemas com ruído pode haver degradação da taxa de erro.



Option Trees

- Aprendizagem
  - Em cada nó é guardada informação sobre testes alternativos.
  - Cada alternativa corresponde a uma arvore.
- Teste
  - Todas as hipóteses alternativas são consultadas.
  - A agregação é efectuada por voto uniforme.
- Redução do erro devido á variância.
- Buntine(1992), Kohavi (1997)



Sumario

- Conjuntos de Classificadores permitem melhorar a performance em relação aos seus componentes.
  - A variabilidade sintáctica é uma propriedade necessária.
  - O erro de um Conjunto de Classificadores melhora, em relação aos seus componentes, quando os classificadores cometem erros não correlacionados.
    - Ali, M. & Pazzani,M.
  - O erro de um Conjunto de Classificadores melhora quando são utilizados classificadores "*radically different types of classifiers*"
    - Tumer & Gosh
  - A utilização de probabilidades de distribuição de classe possibilita pesar a predição do algoritmo.
- Conjuntos de Classificadores
  - Perturbações na distribuição dos exemplos
    - Bagging, Boosting
  - Modelos diferentes
    - Stacking, Cascade

Bibliografia

- Ali and Pazzani, "Error Reduction through learning multiple descriptions", Machine Learning,23, 1996
- Breiman, L. "Stacking Predictors", Machine Learning, 25, 1997
- Breiman,L, "Bagging Predictors", Machine Learning,24, 1997
- Bauer & Kohavi "An empirical comparison of voting classification algorithms: Bagging, Boosting and Variants", Machine Learning, 36, 1999
- Dietterich, T., "Machine Learning Research-Four current directions", AI Magazine, 98
- Freund,Y. and Schapire "Experiments with a new boosting algorithm", ICML96
- Gama,J. "Combining Classifiers by Constructive Induction", ECML98
- Kohavi, R., Kunz,C., "Option Trees with majority votes ", ICML97
- Quinlan,R.,"Bagging, Boosting and C4.5",AAAI96
- Ting, K. & Witten, I, "Stacked Generalization: when it works", IJCAI, 1997
- Wolpert,D. "Stacked Generalization", Neural Networks, N.5