

Mineração de Dados

05 - Técnicas Alternativas de Classificação

Marcos Roberto Ribeiro



Instituto Federal Minas Gerais - Campus Bambuí

2018

Classificadores Baseados em Regras

- Os classificadores baseado em regras classificam os registros usando uma coleção de regras **if ... Then ...**

Exemplo: Regras para Classificar Vertebrados

r_1 : (Gera cria: não) \wedge (Criatura voadora: sim) \rightarrow Pássaro

r_2 : (Gera cria: não) \wedge (Criatura aquática: sim) \rightarrow Peixe

r_3 : (Gera cria: sim) \wedge (Sangue quente: sim) \rightarrow Mamífero

r_4 : (Gera cria: não) \wedge (Criatura voadora: não) \rightarrow Réptil

r_5 : (Criatura aquática: semi) \rightarrow Anfíbio

Cobertura de Regra

- Uma regra r cobre um registro x se a condição de r corresponder aos atributos de x

Exemplo

Nome	Sangue quente	Cobertura de pele	Gera cria	Criatura aquática	Criatura Aérea	Possui Pernas	Hiberna
Falcão	sim	penas	não	não	sim	sim	não
Urso cinzento	sim	pelos	sim	não	não	sim	sim

- Considere, r_1 : (Gera cria: não) \wedge (Criatura voadora: sim) \rightarrow Pássaro
- r_1 cobre o “urso cinzento”, mas não cobre o “falcão”

Qualidade da Classificação de uma Regra

- A qualidade de uma regra de classificação pode ser avaliada usando medidas como cobertura e precisão

$$\text{Cobertura} = \frac{|A|}{|D|}$$

$$\text{Precisão} = \frac{|A \cap y|}{|D|}$$

$|A|$: Número de registros que satisfazem a condição de r

$|A \cap y|$: Número de registros que satisfazem a condição e o consequente de r

$|D|$: Número total de registros

Funcionamento de Classificadores Baseados em Regras

Classificador

r_1 : (Gera cria: não) \wedge (Criatura voadora: sim) \rightarrow Pássaro

r_2 : (Gera cria: não) \wedge (Criatura aquática: sim) \rightarrow Peixe

r_3 : (Gera cria: sim) \wedge (Sangue quente: sim) \rightarrow Mamífero

r_4 : (Gera cria: não) \wedge (Criatura voadora: não) \rightarrow Réptil

r_5 : (Criatura aquática: semi) \rightarrow Anfíbio

Dados

Nome	Sangue quente	Cobertura de pele	Gera cria	Criatura aquática	Criatura Aérea	Possui Pernas	Hiberna
Lêmure	sim	pelos	sim	não	não	sim	não
Tartaruga	não	escamas	não	semi	não	sim	não
Tubarão	não	escamas	sim	sim	não	não	não

- O primeiro vertebrado (lêmure) é de sangue quente e gera seus filhotes. Isto dispara a regra r_3 e, portanto, é classificado como mamífero
- O segundo vertebrado(tartaruga) dispara as regras r_4 e r_5 . Como as

Funcionamento de Classificadores Baseados em Regras

Dados

Nome	Sangue quente	Cobertura de pele	Gera cria	Criatura aquática	Criatura Aérea	Possui Pernas	Hiberna
Lêmure	sim	pelos	sim	não	não	sim	não
Tartaruga	não	escamas	não	semi	não	sim	não
Tubarão	não	escamas	sim	sim	não	não	não

- O primeiro vertebrado (lêmure) é de sangue quente e gera seus filhotes. Isto dispara a regra r_3 e, portanto, é classificado como mamífero
- O segundo vertebrado(tartaruga) dispara as regras r_4 e r_5 . Como as classes previstas pelas regras são contraditórias (répteis ou anfíbios), é necessário resolver este conflito.
- Nenhuma das regras é aplicável a um tubarão. Nesse caso, precisamos assegurar que o classificador continue fazendo uma previsão confiável, mesmo que um registro de teste não seja coberto por nenhuma regra.

Propriedades Importantes do Conjunto de Regras

- O exemplo anterior ilustra duas propriedades importantes do conjunto de regras:

Regras mutuamente exclusivas: As regras em um conjunto de regras R são mutuamente exclusivas se não houver duas regras em R acionadas pelo mesmo registro. Essa propriedade garante que cada registro seja coberto por no máximo uma regra.

Regras completas: Um conjunto de regras R tem cobertura completa se houver uma regra para cada combinação de valores de atributo. Essa propriedade garante que cada registro seja coberto por pelo menos uma regra.

- Juntas, essas propriedades garantem que cada registro seja coberto por exatamente uma regra
- Se o conjunto de regras não estiver completo, então uma regra padrão $r_d : () \rightarrow y_d$, deve ser adicionada para cobrir os casos restantes (onde y_d é a classe majoritária)

Resolvendo Conflitos de Regras I

- Se o conjunto de regras não for mutuamente exclusivo, um registro poderá ser coberto por várias regras
- Existem duas maneiras de resolver esse problema:
 - Regras ordenadas
 - Regras Não Ordenadas

Regras Ordenadas

- Nesta abordagem, as regras são ordenadas em ordem decrescente de sua prioridade, que pode ser definida de várias maneiras (precisão, cobertura, etc.).
- O modelo faz a classificação usando regra mais alta que cobre o registro.
- Isso evita o problema de ter classes conflitantes previstas por várias regras de classificação.
- **Vantagem:** classificação mais rápida usando a primeira regra encontrada

Resolvendo Conflitos de Regras II

Regras Não Ordenadas

- Essa abordagem permite que um registro acione várias regras
- A classificação considera o consequente destas regras como um voto para uma determinada classe
- Os votos são então computados para determinar a classe
- O registro geralmente é atribuído à classe que recebe o maior número de votos
- **Vantagem:** Menos suscetível a erros causados pela primeira regra encontrada

Características de Classificadores Baseados em Regras

- A expressividade de um conjunto de regras é se aproxima à de uma árvore de decisão. Tanto os classificadores baseados em regras quanto os de árvore de decisão criam partições retilíneas do espaço de atributos e atribuem uma classe a cada partição. No entanto, se o classificador baseado em regras permitir que várias regras sejam acionadas para um determinado registro, um limite de decisão mais complexo poderá ser construído.
- Os classificadores baseados em regras geralmente são usados para produzir modelos descritivos que são mais fáceis de interpretar, mas oferecem desempenho comparável ao classificador de árvore de decisão
- A abordagem de ordenação baseada em classe adotada por muitos classificadores baseados em regras é bem adequada para manipular conjuntos de dados com distribuições de classes desequilibradas

Classificadores de Vizinhos Mais Próximos I

- Árvore de decisão e classificadores baseados em regras são **classificadores ávidos** porque são projetados para aprender um modelo que mapeie os atributos de entrada para o rótulo de classe
- Uma estratégia oposta seria atrasar o processo de modelagem dos dados de treinamento até que seja necessário classificar os exemplos de teste
- Técnicas que empregam essa estratégia são conhecidas como **classificadores preguiçosos**.
- Um exemplo de um classificador preguiçoso é o **classificador de Rota**, que memoriza todos os dados de treinamento e executa a classificação apenas se os atributos de uma instância de teste corresponderem exatamente a um dos exemplos de treinamento
- Uma desvantagem dessa abordagem é que alguns registros de teste podem não ser classificados porque não correspondem a nenhum exemplo de treinamento

Classificadores de Vizinhos Mais Próximos II

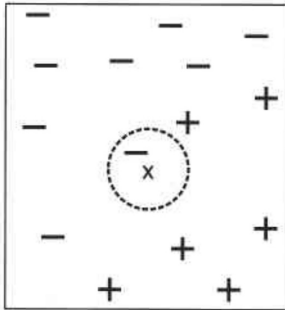
- Uma maneira de tornar essa abordagem mais flexível é encontrar todos os exemplos de treinamento que são relativamente semelhantes aos atributos do exemplo de teste
- Esses exemplos, conhecidos como vizinhos mais próximos, podem ser usados para determinar o rótulo de classe do exemplo de teste
- Um classificador de vizinho mais próximo representa cada exemplo como um ponto de dados em um espaço d -dimensional, onde d é o número de atributos
- Dado um exemplo de teste, calculamos sua proximidade ao resto dos pontos de dados no conjunto de treinamento, usando uma medida de distância
- Os k -vizinhos mais próximos de um registro z referem-se aos k pontos que estão mais próximos de z

Preparação dos Dados

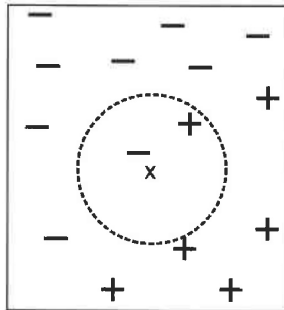
- Necessita de normalização dos dados
- Isso evita que certos atributos dominem completamente a medida de distância
- Exemplo:
 - Altura de um adulto: 1.4m a 2.1m
 - Peso de um adulto: 50Kg a 130Kg
 - Faixa salarial: R\$400 a R\$30.000

Exemplo

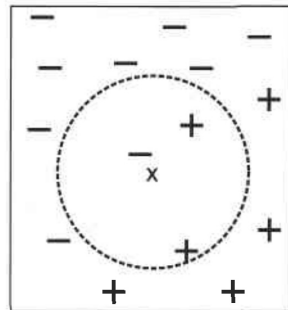
$k = 1$



$k = 2$

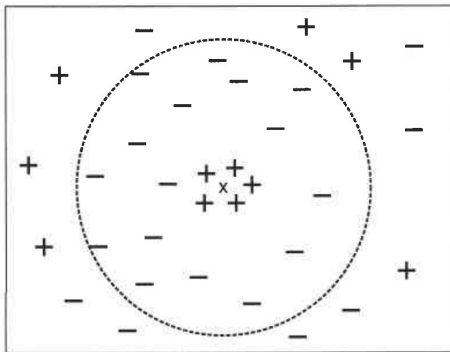


$k = 3$



- A classe no novo registro é a classe da maioria dos k -vizinhos mais próximos
- Se houver empate, podemos reduzir o valor de k gradativamente

O Melhor Valor de k



- Se k é muito pequeno, pode haver *overfitting* por causa de ruídos nos dados
- Por outro lado, se k for muito grande, podem ocorrer classificações erradas porque sua lista de vizinhos mais próximos pode incluir pontos de dados localizados longe de sua vizinhança

Características de Classificadores de Vizinhos Mais Próximos

- Algoritmos de aprendizado baseados em instâncias utilizam medidas de distância para classificar novos dados
- Classificadores preguiçosos não exigem a construção de modelos. No entanto, classificar um exemplo de teste pode ser bastante caro, pois precisamos calcular os valores de proximidade individualmente entre os exemplos de teste e treinamento.
- Já caso dos classificadores ávidos, consomem um tempo razoável na construção de modelos. Uma vez que um modelo foi construído, classificar um exemplo de teste é extremamente rápido.
- Necessitam de grande espaço para armazenar todo o conjunto de dados de treinamento e são mais lentos para avaliar
- Podem ser paralelizado

Características de Classificadores de Vizinhos Mais Próximos II

- Os classificadores de vizinho mais próximo podem produzir limites de decisão configurados arbitrariamente. Esses limites fornecem uma representação de modelo mais flexível em comparação com a árvore de decisão e os classificadores baseados em regras que geralmente são restritos a limites de decisão retilíneos
- Os classificadores de vizinho mais próximo são úteis para grandes conjuntos de dados que possuem poucos atributos cada

Classificadores Bayesianos

- Em muitos casos, o relacionamento entre o conjunto de atributos e a classe é não-determinístico (a classe não pode ser prevista com certeza, mesmo para registros idênticos a alguns dos exemplos de treinamento)
- Esta situação pode surgir devido a ruídos ou à presença de certos fatores de confusão que afetam a classificação, mas não são incluídos na análise
- Por exemplo, considere a tarefa de prever se uma pessoa corre risco de doença cardíaca com base na dieta e na frequência de exercícios
- Embora a maioria das pessoas que tenha alimentação saudável e se exercitam, corre menos risco de doenças cardíacas, elas ainda podem acontecer devido a outros fatores (hereditariedade, fumante etc.)
- Além disto, determinar se a alimentação é saudável ou os exercícios são suficiente está sujeita a interpretação, o que pode introduzir incertezas no problema de aprendizagem
- Os classificadores bayesianos são interessantes porque eles informa a probabilidade de um registro ser de uma determinada classe

O Teorema de Bayes I

- Considere um jogo de futebol entre duas equipes rivais: Time A e Time B
- Suponha que o Time A ganhe 65% das vezes e Team B ganha nas demais partidas
- Entre os jogos vencidos pelo Time A, apenas 30% deles são jogando no campo do Time B
- Por outro lado, as 75% das vitórias do Time B são jogando em casa
- Se o Time B for sediar um jogo entre as duas equipes, quem vencerá?
- Esta questão pode ser respondida usando o conhecido Teorema de Bayes
- Seja X e Y um par de variáveis aleatórias, sua **probabilidade conjunta** $P(X = x, Y = y)$, refere-se à probabilidade de que a variável X assumo o valor x e a variável Y assumo o valor y

O Teorema de Bayes II

- Uma **probabilidade condicional** é a probabilidade de que uma variável assuma um determinado valor, dado que o valor de outra variável é conhecido. Por exemplo, a probabilidade condicional $P(Y = y|X = x)$ refere-se à probabilidade da variável Y assumir o valor y , dado que a variável X possui o valor x
- As probabilidades conjuntas e condicionais para X e Y estão relacionadas da seguinte maneira:

$$P(X, Y) = P(Y|X) \times P(X) = P(X|Y) \times P(Y) \quad (1)$$

- O Teorema de Bayes está relacionado com as duas últimas expressões da Equação (1):

$$P(Y|X) = \frac{P(X|Y) \times P(Y)}{P(X)} \quad (2)$$

Usando o Teorema de Bayes para Classificação

- Seja X um conjunto de atributos e Y a classe, devemos calcular a **probabilidade posterior** $P(Y|X)$
- A probabilidade da classe sem considerar os valores dos demais atributos, $P(Y)$, é chamada de **probabilidade anterior**
- Durante o treinamento, precisamos aprender as probabilidades posteriores $P(Y|X)$ para cada combinação de X e Y com base nos dados de treinamento
- Conhecendo estas probabilidades, um registro de teste X' pode ser classificado encontrando a classe Y' com maior probabilidade posterior $P(Y'|X')$

Exemplo

ID	Casa Própria	Estado Civil	Renda	Inadimplente
1	sim	solteiro	125K	não
2	não	casado	100K	não
3	não	solteiro	70K	não
4	sim	casado	120K	não
5	não	divorciado	95K	sim

ID	Casa Própria	Estado Civil	Renda	Inadimplente
6	não	casado	60K	não
7	sim	divorciado	220K	não
8	não	solteiro	85K	sim
9	não	casado	75K	não
10	não	solteiro	90K	sim

- Como classificar o registro $X = (\text{não}, \text{casado}, 120K)$?
- Precisamos calcular e comparar as probabilidades $P(\text{sim}|X)$ e $P(\text{não}|X)$

$$P(Y|X) = \frac{P(X|Y) \times P(Y)}{\cancel{P(X)}}$$

- Podemos ignorar o denominador porque ele é constante para todas as classes

O Classificador Bayes Simples

- Também chamado de **Naive Bayes** por que não considera a independência condicional entre os atributos
- Com isto não é necessário calcular a probabilidade condicional para cada combinação de atributos
- A probabilidade de cada classe pode então ser calculada com a fórmula:

$$P(Y|X) = P(Y) \times \prod_{i=1}^d P(X_i|Y) \quad (3)$$

- Onde X_i é o i -ésimo atributo de X e d é o número de atributos
- Como calcular uma probabilidade $P(X_i, Y)$ de um atributo X_i ?

Probabilidade Condicional de Atributos Categorizados

- Para um atributo categórico X_i , a probabilidade condicional $P(X_i = x|Y = y)$ é calculada de acordo com a fração de instâncias na classe y com valor x no atributo X_i
- Exemplo: $P(\text{Casa própria} = \text{sim}, \text{Inadimplente} = \text{não}) = \frac{3}{7}$

Probabilidade Condicional de Atributos Contínuos

- Uma possibilidade é discretizar os atributos contínuos e calcular as probabilidades como se fosse categóricos
- Porém, discretizações mal feitas podem levar baixa precisão
- Outra possibilidade é supor uma distribuição normal dos dados e calcular densidade de probabilidade

Densidade de Probabilidade

$$P(X_i = x | Y = y_j) = \frac{1}{\sqrt{2 \times \pi \times \sigma_{i,j}^2}} \times e^{-\frac{x - \mu_{i,j}}{2 \times \sigma_{i,j}^2}}$$

Desvio Padrão

$$\sigma_{i,j} = \sqrt{\frac{\sum_{i=1}^n (x_{i,j} - \mu)^2}{n_j - 1}}$$

$\mu_{i,j}$: Média dos valores de X_i nos registros com classe y_j

$\sigma_{i,j}$: Desvio padrão dos valores de X_i nos registros com classe y_j

$x_{i,j}$: Valor de X_i nos registros com classe y_j

n_j : Quantidade de registros com classe y_j

Exemplo: Atributo Renda

Inadimplente = sim

$$\mu = \frac{95 + 85 + 90}{3} = 90$$

$$\sigma = \sqrt{\frac{(95 - 90)^2 + \dots + (90 - 90)^2}{3 - 2}} \simeq 5$$

Inadimplente = não

$$\mu = \frac{125 + 100 + 75}{3} = 110$$

$$\sigma = \sqrt{\frac{(125 - 110)^2 + \dots + (75 - 110)^2}{7 - 1}} \simeq 54.54$$

Exemplo de Probabilidade

$$P(\text{Renda} = 120 | \text{Inadimplente} = \text{n\~ao}) = \frac{1}{\sqrt{2 \times \pi} \times 54.54} \times e^{-\frac{(120 - 110)^2}{2 \times 54.54^2}} = 0.0072$$

Exemplo de Classificador Bayesiano Simples

ID	Casa Própria	Estado Civil	Renda	Inadimplente
1	sim	solteiro	125K	não
2	não	casado	100K	não
3	não	solteiro	70K	não
4	sim	casado	120K	não
5	não	divorciado	95K	sim
6	não	casado	60K	não
7	sim	divorciado	220K	não
8	não	solteiro	85K	sim
9	não	casado	75K	não
10	não	solteiro	90K	sim

- O treino do classificador consiste em calcular as probabilidades condicionais dos atributos
- No caso dos atributos contínuos, calculamos a média e o desvio padrão

- $P(\text{Casa Própria} = \text{sim} | \text{não}) = 3/7$
- $P(\text{Casa Própria} = \text{não} | \text{não}) = 4/7$
- $P(\text{Casa Própria} = \text{sim} | \text{não}) = 0/3 = 0$
- $P(\text{Casa Própria} = \text{sim} | \text{não}) = 3/3 = 1$
- $P(\text{Estado Civil} = \text{solteiro} | \text{não}) = 2/7$
- $P(\text{Estado Civil} = \text{divorciado} | \text{não}) = 1/7$
- $P(\text{Estado Civil} = \text{casado} | \text{não}) = 4/7$
- $P(\text{Estado Civil} = \text{solteiro} | \text{sim}) = 2/3$
- $P(\text{Estado Civil} = \text{divorciado} | \text{sim}) = 1/3$
- $P(\text{Estado Civil} = \text{casado} | \text{sim}) = 0/3 = 0$
- Renda (para inadimplente = não):
 - Média = 110
 - Desvio padrão = 54.54
- Renda (para inadimplente = sim):
 - Média = 90
 - Desvio padrão = 5

Classificando Registros

- Calculamos as probabilidades posteriores da classe e escolhemos aquela com maior valor

Exemplo: $X = (\text{Casa Própria} = \text{não}, \text{Estado Civil} = \text{casado}, \text{Renda} = 120K)$

$$\begin{aligned} P(X|\text{não}) &= P(\text{Casa Própria} = \text{não}|\text{não}) \times P(\text{Estado Civil} = \text{casado}|\text{não}) \\ &\quad \times P(\text{Renda} = 120|\text{não}) \\ &= 4/7 \times 4/7 \times 0.0072 = 0.0024 \end{aligned}$$

$$\begin{aligned} P(X|\text{sim}) &= P(\text{Casa Própria} = \text{não}|\text{sim}) \times P(\text{Estado Civil} = \text{casado}|\text{sim}) \\ &\quad \times P(\text{Renda} = 120|\text{sim}) \\ &= 1 \times 0 \times (1.2 \times 10^{-9}) = 0 \end{aligned}$$

Como $P(X|\text{não}) > P(X|\text{sim})$, o registro é classificado como **inadimplente = não**

Problema da Frequência Zero

- Quanto determinado valor não aparece no treinamento, mas aparece no teste, probabilidade será zero

$$\begin{aligned}P(X|\text{sim}) &= P(\text{Casa Própria} = \text{não}|\text{sim}) \times P(\text{Estado Civil}=\text{casado} \mid \text{sim}) \\&\quad \times P(\text{Renda} = 120|\text{sim}) \\&= 1 \times 0 \times (1.2 \times 10^{-9}) = 0\end{aligned}$$

- Não importa as probabilidades dos outros atributos (o resultado é sempre zero)
- Zerar a probabilidade a posteriori é muito radical
 - A base de treinamento pode não ser totalmente representativa
 - Classes minoritárias podem ter valores raros
- Uma solução possível é usar o **estimador de Laplace**
 - Adicionamos 1 unidade fictícia para cada combinação de valor-classe
 - Valores sem exemplos de treinamento passam a conter 1 exemplo
 - As probabilidades nunca serão zero

Exemplo de Estimador Laplace

- Somamos um exemplo no numerador e a quantidade valores por classe no denominador
 - $P(\text{Casa Própria} = \text{sim}|\text{não}) = 3/7 \rightarrow \frac{3+1}{7+2}$
 - $P(\text{Casa Própria} = \text{não}|\text{não}) = 4/7 \rightarrow \frac{3+1}{7+2}$
 - $P(\text{Casa Própria} = \text{sim}|\text{não}) = 0/3 \Rightarrow \frac{0+1}{3+2}$
 - $P(\text{Casa Própria} = \text{sim}|\text{não}) = 3/3 \Rightarrow \frac{3+1}{3+2}$
 - $P(\text{Estado Civil} = \text{solteiro}|\text{não}) = 2/7 \rightarrow \frac{2+1}{7+3}$
 - $P(\text{Estado Civil} = \text{divorciado}|\text{não}) = 1/7 \rightarrow \frac{1+1}{7+3}$
 - $P(\text{Estado Civil} = \text{casado}|\text{não}) = 4/7 \rightarrow \frac{4+1}{7+3}$
 - $P(\text{Estado Civil} = \text{solteiro}|\text{sim}) = 2/3 \rightarrow \frac{2+1}{3+3}$
 - $P(\text{Estado Civil} = \text{divorciado}|\text{sim}) = 1/3 \rightarrow \frac{1+1}{3+3}$
 - $P(\text{Estado Civil} = \text{casado}|\text{sim}) = 0/3 \rightarrow \frac{0+1}{3+3}$
- Isso deve ser feito para todas as classes
- Caso contrário, estamos inserindo viés nas probabilidades de apenas uma classe

Características de Classificadores Bayesianos Simples

- Robustos para pontos de ruído isolados (apresentam pouco impacto na probabilidade)
- Podem tratar valores faltantes (ignorando-os)
- Robustos para lidar com atributos irrelevantes (a probabilidade se torna quase uniformemente distribuída)
- Atributos correlacionados podem degradar o desempenho porque a suposição de independência condicional não é mais válida para tais atributos

Métodos de Grupo

- As técnicas de classificação que vimos até agora, com exceção do método do vizinho mais próximo, preveem as classes de exemplos desconhecidos usando um único classificador
- Uma alternativa para melhorar a previsão de classificação é agregar as previsões de múltiplos classificadores
- Essas técnicas são conhecidas como os **métodos de grupo**
- Um método de grupo constrói um conjunto de classificadores a partir de dados de treinamento e executa a classificação, votando nas previsões feitas por cada classificador
- Usualmente, os métodos de grupo melhor previsão do que classificadores únicos

Raciocínio do Método de Grupo

- Considere um conjunto de 25 classificadores base, cada um com uma taxa de erro de 35% ($\epsilon = 0.35$)
- Um possível método de grupo pode prever a classe, tomando a votação majoritária sobre as previsões feitas pelos 25 classificadores base
- Se os classificadores base forem idênticos, o conjunto classificará erroneamente os mesmos exemplos preditos incorretamente pelos classificadores base (permanecendo a mesma taxa de erro)
- Por outro lado, se os classificadores base forem independentes (seus erros não forem correlacionados), então o conjunto fará uma predição errada somente se mais da metade dos classificadores base preverem incorretamente
- Nesse caso, a taxa de erro do classificador de conjunto será:

$$\epsilon_{\text{grupo}} = \sum_{i=13}^{25} \binom{25}{i} \epsilon^i (1 - \epsilon)^{(25 - i)} = 0.06$$

Condições para Criação de um Bom Classificador em Grupo

- 1) Os classificadores base devem ser independentes entre si
- 2) Os classificadores base devem ser melhores do que do que classificadores com suposições aleatórias (menos de 50% de erros)
 - Na prática, é difícil garantir independência total entre os classificadores
 - Contudo, existe chance de ocorrerem melhorias nas previsões

Métodos para Criar um Classificador

- Manipulando o Conjunto de Treinamento
- Manipulando as Características de Entrada
- Manipulando os Rótulos de Classes
- Manipulando o Algoritmo de Aprendizagem

Manipulando o Conjunto de Treinamento

- Nessa abordagem, vários conjuntos de treinamento são criados por amostragem sobre os dados originais, de acordo com alguma distribuição de amostragem
- A distribuição amostral determina a probabilidade de um exemplo ser selecionado para treinamento e pode variar de um teste para outro
- Um classificador é construído a partir de cada conjunto de treinamento usando um algoritmo de aprendizado específico
- **Bagging** e **boosting** são dois exemplos de métodos conjuntos que manipulam seus conjuntos de treinamento

Manipulando as Características de Entrada

- Nesta abordagem, um subconjunto de atributos de entrada é escolhido para formar cada conjunto de treinamento
- O subconjunto pode ser escolhido aleatoriamente ou com base na recomendação de especialistas no domínio
- Alguns estudos mostraram que essa abordagem funciona muito bem com conjuntos de dados que contêm recursos altamente redundantes

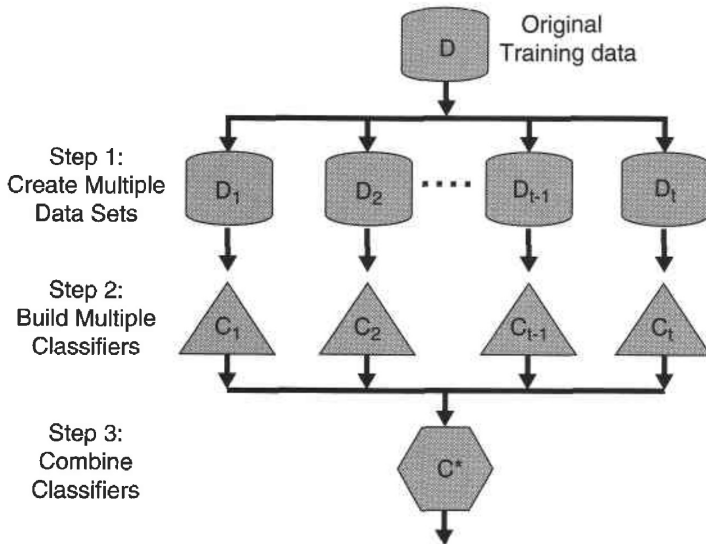
Manipulando os Rótulos de Classes

- Este método pode ser usado quando o número de classes é suficientemente grande
- Os dados de treinamento são transformados em um problema de classe binária, particionando aleatoriamente as classes em dois subconjuntos disjuntos, A_0 e A_1 . Os exemplos de treinamento do subconjunto A_0 são atribuídos à classe 0, enquanto aqueles que pertencem ao subconjunto A_1 são atribuídos à classe 1
- Os exemplos remarcados são então usados para treinar um classificador base
- Repetindo as etapas de reclassificação de classe e construção de modelo várias vezes, é obtido um conjunto de classificadores base
- Quando um exemplo de teste é apresentado, cada classificador C_i é usada para prever seu rótulo de classe
- Se o exemplo de teste for predito como classe 0, todas as classes pertencentes a A_0 receberão um voto. Por outro lado, se for previsto que seja classe 1, todas as classes pertencentes a A_1 receberão um voto
- Os votos são computados e a classe que recebe o maior voto é atribuída ao exemplo de teste

Manipulando o Algoritmo de Aprendizagem

- Muitos algoritmos de aprendizagem podem ser manipulados de tal forma que a aplicação do algoritmo várias vezes nos mesmos dados de treinamento pode resultar em modelos diferentes
- Por exemplo, uma rede neural artificial pode produzir diferentes códigos alterando sua topologia de rede ou os pesos iniciais dos links entre os neurônios
- Da mesma forma, um conjunto de árvores de decisão pode ser modificado, em vez de escolher o melhor atributo de divisão em cada nó, podemos escolher aleatoriamente um dos principais k atributos para dividir

Técnica Genérica para Métodos de Grupo



Técnica Genérica para Métodos de Grupo

- As três primeiras abordagens são métodos genéricos que são aplicáveis a qualquer classificador, enquanto a quarta abordagem depende do tipo de classificador usado
- Para os métodos genéricos, o primeiro passo é criar um conjunto de treinamento D_i a partir dos dados originais D
- Dependendo do tipo de método de grupo usado, os conjuntos de treinamento são idênticos ou pequenas modificações de D
- O tamanho do conjunto de treinamento é mantido, mas a distribuição dos exemplos pode não ser idêntica, ou seja, alguns exemplos podem aparecer várias vezes no conjunto de treinamento, enquanto outros podem não aparecer sequer uma vez
- Um classificador base C_i é então construído a partir de cada conjunto de treinamento D_i
- Os métodos de grupo funcionam melhor com classificadores instáveis, isto é, classificadores base que são sensíveis a pequenas perturbações no conjunto de treino (árvores de decisão, classificadores baseados em regras e redes neurais artificiais)
- Finalmente, um exemplo de teste x é classificado combinando as previsões feitas por todos os classificadores base
- A classe pode ser obtida através de uma votação majoritária sobre as previsões individuais ou ponderando cada previsão com a precisão do classificador base

Bagging I

- O *bagging*, que também é conhecido como agregação de *bootstrap*, é uma técnica que repetidamente coleta amostras (com substituição) de um conjunto de dados de acordo com uma distribuição uniforme de probabilidade
- Cada amostra de *bootstrap* tem o mesmo tamanho que os dados originais
- Como a amostragem é feita com substituição, algumas instâncias podem aparecer várias vezes no mesmo conjunto de treinamento, enquanto outras podem ser omitidas
- Em média, uma amostra de *bootstrap* D_i contém aproximadamente 63% dos dados originais de treinamento
- O *bagging* melhora o erro de generalização reduzindo a variação dos classificadores base
- O desempenho do *bagging* depende da estabilidade do classificador base

Bagging II

- Se um classificador base for instável, o *bagging* ajuda a reduzir os erros associados a flutuações aleatórias nos dados de treinamento
- Se um classificador de base é estável, o *bagging* pode não ser capaz de melhorar significativamente o desempenho dos classificadores de base
- Como cada amostra tem uma probabilidade igual de ser selecionada, o *bagging* não se concentra em nenhuma instância específica dos dados de treinamento, sendo menos suscetível a modelar *overfitting* quando aplicado a dados ruidosos

Boosting

- O *boosting* é um procedimento iterativo usado para alterar de forma adaptativa a distribuição de exemplos de treinamento para que os classificadores base se concentrem em exemplos difíceis de classificar
- Diferente do *bagging*, o *boosting* atribui um peso a cada exemplo de treinamento e pode alterar o peso de forma adaptável ao final de cada rodada de reforço
- Os pesos atribuídos aos exemplos de treinamento podem ser usados das seguintes maneiras:
 - 1) Eles podem ser usados como uma distribuição de amostragem para desenhar um conjunto de amostras de *bootstrap* a partir dos dados originais
 - 2) Eles podem ser usados pelo classificador base para aprender um modelo com tendência para exemplos de maior peso

Funcionamento do Boosting

- Inicialmente, os exemplos recebem pesos iguais, $1/N$, para que tenham a mesma probabilidade de serem escolhidos para treinamento
- Uma amostra é desenhada de acordo com a distribuição amostral dos exemplos de treinamento para obter um novo conjunto de treinamento
- Em seguida, um classificador é induzido a partir do conjunto de treinamento e usado para classificar todos os exemplos nos dados originais
- Os pesos dos exemplos de treinamento são atualizados no final de cada rodada de reforço
- Exemplos que são classificados incorretamente terão seus pesos aumentados, enquanto aqueles que são classificados corretamente terão seus pesos diminuídos
- Isso força o classificador a se concentrar em exemplos difíceis de classificar nas iterações subsequentes

O Algoritmo AdaBoost

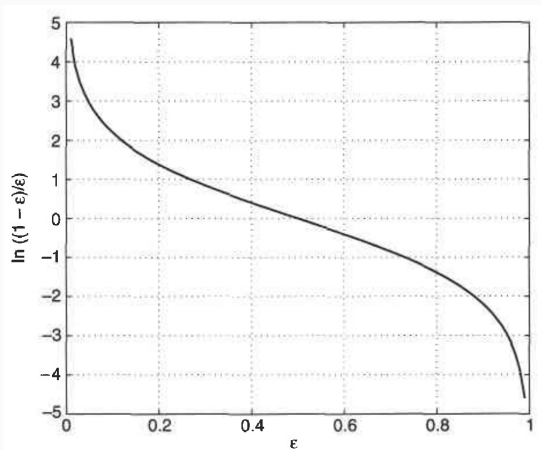
- Seja $\{(x_i, y_i)\}$ tal que $i \in \{1, 2, \dots, N\}$ um conjunto de N exemplos de treinamento
- No algoritmo AdaBoost, a importância de um classificador base C_j depende de sua taxa de erro, que é definida como:

$$\epsilon_j = \frac{1}{N} \left(\sum_{i=1}^N w_i I(C_j(x_i) \neq y_i) \right)$$

- Onde $I(C_j(x_i) \neq y_i) = 1$ se $C_j(x_i) \neq y_i$ for verdadeiro e 0 caso contrário
- A importância de um classificador C_j é dada pelo seguinte parâmetro:

$$\alpha_j = \frac{1}{2} \ln \left(\frac{1 - \epsilon_j}{\epsilon_j} \right)$$

O Parâmetro α_j



- Observe que α_j tem valores altos se a taxa de erro estiver próxima de 0 e valores baixos se a taxa de erro for próxima de 1

Atualização dos Pesos

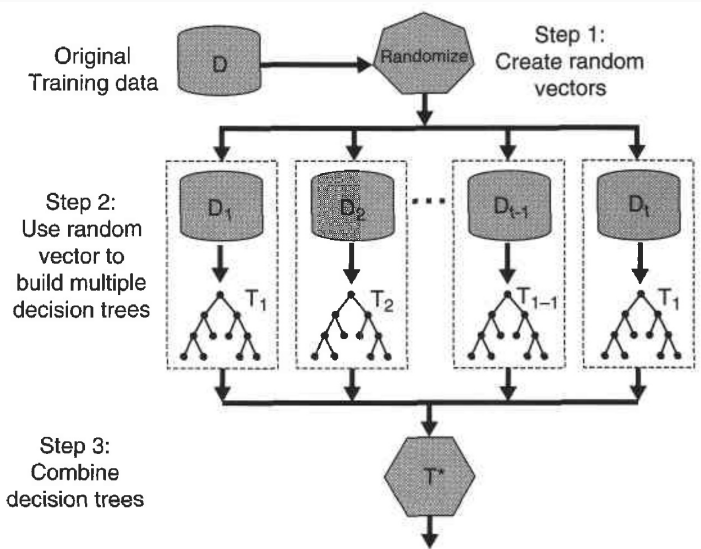
- O parâmetro α_j é usado para atualizar o peso dos exemplos de treinamento
- Seja $w_i^{(j)}$ o peso atribuído ao exemplo (x_i, y_i) na rodada j
- O mecanismo de atualização de pesos do AdaBoost é dado pela equação:

$$w_i^{(j+1)} = \frac{w_i^{(j)}}{Z_j} \times \begin{cases} \exp^{-\alpha_j} & \text{se } C_j(x_i) = y_i, \\ \exp^{\alpha_j} & \text{se } C_j(x_i) \neq y_i, \end{cases}$$

- Onde Z_j é o fator de normalização usado para garantir que $\sum_i w_i^{(j+1)} = 1$
- Desta maneira, o peso dos exemplos classificados incorretamente é aumentado e o peso dos exemplos classificados corretamente é reduzido


- Em vez de usar um esquema de votação majoritária, a previsão feita por cada classificador C_j é ponderada de acordo com α_j
- Essa abordagem permite que a AdaBoost penalize modelos que têm baixa precisão, por exemplo, aqueles gerados nas rodadas de reforço anteriores
- Além disso, se qualquer ciclo intermediário produzir uma taxa de erro maior que 50%, os pesos serão revertidos para seus valores uniformes originais, $w_i = 1/N$, e o procedimento de re-amostragem será repetido

Florestas Aleatórias



Comparação Empírica entre Métodos de Grupo

Base de Dados	#(Att, Clas, Rec)	Decision Tree (%)	Bagging (%)	Boosting (%)	RF (%)
Anneal	(39, 6, 898)	92.09	94.43	95.43	95.43
Australia	(15, 2, 690)	85.51	87.10	85.22	85.80
Auto	(26, 7, 205)	81.95	85.37	85.37	84.39
Breast	(11, 2, 699)	95.14	96.42	97.28	96.14
Cleve	(14, 2, 303)	76.24	81.52	82.18	82.18
Credit	(16, 2, 690)	85.80	86.23	86.09	85.80
Diabetes	(9, 2, 768)	72.40	76.30	73.18	75.13
German	(21, 2, 1000)	70.90	73.40	73.00	74.50
Glass	(10, 7, 214)	67.29	76.17	77.57	78.04
Heart	(14, 2, 270)	80.00	81.48	80.74	83.33
Hepatitis	(20, 2, 155)	81.94	81.29	83.87	83.23
Horse	(23, 2, 368)	85.33	85.87	81.25	85.33
Ionosphere	(35, 2, 351)	89.17	92.02	93.73	93.45
Iris	(5, 3, 150)	94.67	94.67	94.00	93.33
Labor	(17, 2, 57)	78.95	84.21	89.47	84.21
LedT	(8, 10, 3200)	73.34	73.66	73.34	73.06
Lymphography	(19, 4, 148)	77.03	79.05	85.14	82.43
Pima	(9, 2, 768)	74.35	76.69	73.44	77.60
Sonar	(61, 2, 208)	78.85	78.85	84.62	85.58
Tic-tac-toe	(10, 2, 958)	83.72	93.84	98.54	95.82
Vehicle	(19, 4, 846)	71.04	74.11	78.25	74.94
Waveform	(22, 3, 5000)	76.44	83.30	83.90	84.04
Wine	(14, 3, 178)	94.38	96.07	97.75	97.75
Zoo	(17, 7, 101)	93.07	93.07	95.05	97.03

 Quinlan, J. R. (1986).
Induction of decision trees.
Machine learning, 1(1):81–106.

 TAN, P.-N., STEINBACH, M., and KUMAR, V. (2009).
Introdução ao data mining: mineração de dados.
Ciência Moderna, Rio de Janeiro.