Mineração de Dados

Algoritmos para Classificação

Sumário

- Algoritmo 1R
- Naïve Bayes
- Árvore de Decisão
- Regras
- Modelos Lineares (Regressão)
- Baseados em Instância (Vizinhos mais Próximos)

Modelos Lineares

- Trabalham mais naturalmente com atributos numéricos
- Técnica padrão para previsão numérica: regressão linear
  - Resultado é uma combinação linear de atributos

$$x = w_0 + w_1 a_1 + w_2 a_2 + \ldots + w_k a_k$$

- Pesos são calculados para os dados de treinamento
- Valor previsto para a primeira instância $\mathbf{a}^{(1)}$

$$w_0 a_0^{(1)} + w_1 a_1^{(1)} + w_2 a_2^{(1)} + \ldots + w_k a_k^{(1)} = \sum_{j=0}^{k} w_j a_j^{(1)}$$

3

Regressão

- A análise de regressão representa os dados através de um modelo linear aditivo, onde o modelo inclui um componente sistemático e um aleatório.

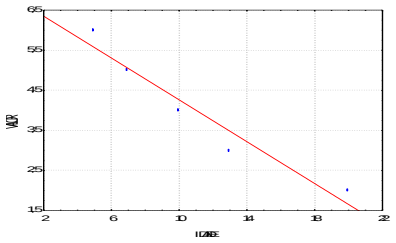
$$Y = f(X) + \varepsilon$$

  - f descreve a relação entre X e Y.
  - ε são os erros aleatórios.
  - Y = variável resposta ou dependente;
  - X = variável independente, concomitante, covariável ou variável preditora.

4

Minimizando o Erro Quadrático

- Escolhe k +1 coeficientes para minimizar o erro quadrático nos dados de treinamento:
- Erro quadrático: $$\sum_{i=1}^{n}\left(x^{(i)} - \sum_{j=0}^{k} w_j a_j^{(i)}\right)^2$$



5

Regressão para Classificação

- Qualquer técnica de regressão pode ser usada para classificação
  - Treinamento: executa uma regressão para cada classe, ajustando a saída para 1 quando as instâncias de treinamento pertencem a classe, e para 0 quando isto não acontece
  - Previsão: prevê a classe correspondente ao modelo com o maior valor de saída (valor de pertinência)
- Para regressão linear isto é conhecido como regressão linear multi-resposta

6

Regressão a Par

- Outra forma de utilizar regressão para fazer classificação
  - Utiliza-se uma função de regressão para cada par de classes, usando apenas as instâncias destas duas classes
  - Uma saída de +1 é atribuída a um membro do par e uma de -1 ao outro membro
- A predição é feita através de votação
  - A classe que recebe mais votos é a que é predita
  - Alternativa: "não sei" quando não há acordo
- Este método é freqüentemente mais preciso, mas também é mais caro computacionalmente

Regressão Logística

- Problema: algumas suposições são violadas quando se aplica regressão linear em problemas de classificação
- Regressão *Logística*: alternativa para regressão linear
  - Desenvolvida para problemas de classificação
  - Tenta estimar probabilidades de classes diretamente
    - Usa o método da máxima verossimilhança
  - Usa o modelo linear:

$$\log\left(\frac{P}{1-P}\right) = w_0 a_0 + w_1 a_1 + w_2 a_2 + \ldots + w_k a_k$$

P= Probabilidade da Classe

Discussão de modelos lineares

- Não são apropriados se dados exibem dependências não-lineares
- Mas pode servir como blocos construtores para esquemas mais complexos
- Exemplo: regressão linear multi-resposta define um hiperplano para quaisquer duas classes dadas:

$$(w_0^{(1)} - w_0^{(2)})a_0 + (w_1^{(1)} - w_1^{(2)})a_1 + (w_2^{(1)} - w_2^{(2)})a_2 + \ldots + (w_k^{(1)} - w_k^{(2)})a_k > 0$$

Comentários sobre Métodos Básicos

- Minsky e Papert (1969) mostraram que classificadores lineares têm limitações
  - Não podem aprender o XOR por exemplo
  - Combinações deles podem (redes neurais por exemplo)

Aprendizagem Baseada em Instância (IBL) ou aprendizagem preguiçosa

- Simplesmente armazena os exemplos de treinamento
- Deixa a generalização de f só para quando uma nova instância precisa ser classificada
- A cada nova instância, uma f nova e local é estimada
- Métodos: vizinhos mais próximos, regressão localmente ponderada, raciocínio baseado em casos, etc.

Aprendizagem Baseada em Instância

- Forma mais simples de aprendizado
  - Instâncias de treinamento são usados para identificar qual a classe que mais se parece com a nova instância que se quer identificar
  - As próprias instâncias representam o conhecimento
- Função de similaridade define o que é aprendido
- Este aprendizado é um tipo de aprendizado preguiçoso
- Métodos:
  - Vizinho mais próximo
  - K-vizinhos mais próximos: Método mais antigo (1967) e difundido

Aprendizado Baseado em Instâncias

- Função distância define o que é aprendido
- Instâncias são *representadas* por pontos num espaço n dimensional $\Re^n$
- Maior parte dos esquemas baseados em instâncias usam distância Euclidiana

$$\sqrt{(a_1^{(1)} - a_1^{(2)})^2 + (a_2^{(1)} - a_2^{(2)})^2 + ... + (a_k^{(1)} - a_k^{(2)})^2}$$

  $\mathbf{a}^{(1)}$ e $\mathbf{a}^{(2)}$: duas instâncias com *k* atributos
- Calcular a raiz quadrada é desnecessário quando se quer comparar distâncias

13

Aprendizado Baseado em Instâncias

- Distância Manhattan (city-block)
  - Adiciona diferenças sem elevar ao quadrado
  - Em um plano que contém os pontos P_1 e P_2, respectivamente com as coordenadas (x_1,y_1) e (x_2,y_2), é definido por:
  - $| x_1-x_2| + | y_1-y_2|$

14

Aprendizado Baseado em Instâncias

- Coeficiente de Correlação de Pearson:
  - mede o nível de relacionamento entre duas variáveis

$$r = \frac{\sum\left(x_i - \bar{x}\right)\left(y_i - \bar{y}\right)}{\sqrt{\left[\sum\left(x_i - \bar{x}\right)^2 \sum\left(y_i - \bar{y}\right)^2\right]}}$$

- onde x e y são os valores medidos de ambas as variáveis e $\bar{x}$ e $\bar{y}$ são respectivamente suas médias
- r assume apenas valores entre -1 e 1.

Aprendizado Baseado em Instâncias

- Coeficiente de Correlação de Pearson:

$$d(x,y) = \frac{(1 - r(x,y))}{2} \quad (1) \qquad d(x,y) = 1 - |r(x,y)| \quad (2)$$

- Em (1) variáveis com alta correlação tem medida de dissimilaridade próximas a zero, enquanto variáveis com alta correlação negativa terão níveis de dissimilaridades próximas a 1.
- Em (2) variáveis com alta correlação positiva ou negativa terão coeficiente de dissimilaridade próximos a zero.

Aprendizado Baseado em Instâncias

- Potências mais elevadas:
  - Aumentam a influência de grandes diferenças às custas de pequenas diferenças
  - Outras métricas de distâncias podem ser mais apropriadas em circunstâncias especiais

Normalização e outras considerações

- Diferentes atributos são medidos em diferentes escalas, então se a distância Euclidiana for usada diretamente, os efeitos de alguns atributos pode ser completamente minimizados por outros que tenham escalar maiores.

$\Rightarrow$ precisam ser *normalizados*:

$$a_i = \frac{v_i - \min v_i}{\max v_i - \min v_i} \quad \text{ou} \quad a_i = \frac{v_i - Avg(v_i)}{StDev(v_i)}$$

v_i: o valor real de i

Função de Distância: Atributos Nominais

- Atributos nominais: distância é igual a 1 se os valores são diferentes e 0 se são iguais
- Nesse caso os valores já estão em escala

- Todos os atributos são igualmente importantes ?
  - Dar pesos para os atributos pode ser necessário

19

Função de Distância: Valores Faltantes

- Para atributos nominais, assume-se que a característica é a maior diferenca possível entre todas os outras valores
  - Se ambos são faltantes ou diferentes: a diferenca entre eles é 1
  - A diferença é 0 (zero) somente se eles não são faltantes e se são iguais.

20

Função de Classificação

- A função de classificação $\hat{f}$
  - Caso seja discreta, seu resultado é aquele que aparecer mais vezes entre os *k* vizinhos mais próximos (V = conjunto de valores possíveis da função)

$$f : \Re^n \rightarrow V$$

  - Caso seja contínua, seu resultado é a média dos resultados dos *k* vizinhos mais próximos

$$f : \Re^n \rightarrow \Re$$

Função de Classificação

- x = < idade(x), altura(x), peso(x)>,
  onde adimplente pode ser “sim”, “não”]
- Exemplo de treinamento = (x,f(x)), onde f(x) é a função de classificação a ser aprendida
  - joão = (<36, 1.80, 76>, ???) a ser classificado
  - josé = (<30, 1.78, 72>, sim)
  - maria = (<25, 1.65, 60>, sim)
  - anastácia = (<28, 1.60, 68>, não)
- Distância
  - $d(\text{joão,josé}) = [(36-30)^2 + (1.80-1.78)^2 + (76-72)^2]^{1/2} = (36+0.0004+16)^{1/2} = 7{,}21$
  - $d(\text{joão,maria}) = (121+0.0225+256)^{1/2} = 19{,}41$

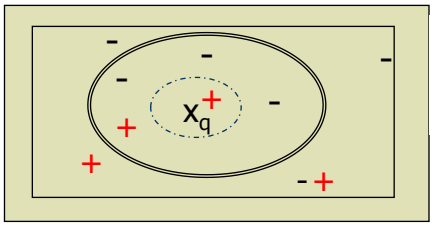
k vizinhos mais próximos

- Treinamento
  - Adicione cada instância de treinamento <x,f(x)> na lista instancias_treinamento
- Para cada instância x_q a ser classificada
  - Chame de $x_1,x_2,...x_k$ as *k* instâncias mais *próximas* de x_q na lista *instancias_treinamento*
  - Retorna
    - Caso discreto $\hat{f}(x_q) \leftarrow \arg\max_{v \in V} \sum_{i=1}^{k} \delta(v, f(x_i))$

      onde $\delta(a,b)$ é igual a 1 se $a = b$ e 0 se $a \neq b$
    - Caso contínuo $\hat{f}(x_q) \leftarrow \frac{\sum_{i=1}^{k} f(x_i)}{k}$

k vizinhos mais próximos: exemplo

- Caso discreto



k = 1 classifica x_q como +

k = 5 classifica x_q como -

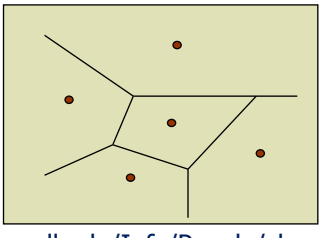
  - O valor de k é determinante na classificação

k vizinhos mais próximos: exemplo

- Caso contínuo
  - exemplo = filme = <ano, bilheteria>
  - classificação f = recomendação r $\in$ Z, r = [1...5]
  - $r(x_1) = 4$, $r(x_2) = 3$, $r(x_3) = 5$, $r(x_4) = 2$
  - para k = 3 e supondo que x_1, x_2 e x_3 são os mais próximos de x_q, temos
  - $f(x_q) = (4+3+5)/3 = 4$

k vizinhos mais próximos

- Visualização da "superfície de decisão", para k = 1
  - Diagrama de Voronoi => poliedro convexo para cada instância de treinamento.
  - As instâncias dentro do poliedro são completamente classificados pela instância associada



http://www.cs.cornell.edu/Info/People/chew/chew.html

k vizinhos mais próximos

- Problema da dimensionalidade
  - Para calcular a distância entre os pontos, o método utiliza todos os atributos da instância
- Conseqüências
  - pode custar caro
  - atributos irrelevantes podem deturpar a classificação
- Refinamentos
  - Atribuir pesos ω_j aos atributos de maneira que minimize a taxa de erro de classificação (Importância do atributo)
  - Usar a técnica de validação cruzada para automaticamente escolher os pesos
  - Eliminar atributos do espaço de instâncias

27

k vizinhos mais próximos

- Refinamento: distância para o vizinho
  - ponderar a contribuição de cada um dos k vizinhos de acordo com sua distância ao ponto de consulta x_q
  - melhora robustez

    - Caso discreto

$$\hat{f}(x_q) \leftarrow \arg\max_{v \in V} \sum_{i=1}^{k} \omega_i \delta(v, f(x_i))$$

    - Caso contínuo

$$\hat{f}(x_q) \leftarrow \frac{\sum_{i=1}^{k} \omega_i f(x_i)}{\sum_{i=1}^{k} \omega_i} \quad \text{onde} \quad \omega_i \equiv \frac{1}{d(x_i, x_q)}$$

  -

28

Discussão do algoritmo 1-NN

- Geralmente bastante preciso:
  - Estatísticos tem usado k-NN desde 1950
  - Se $n \rightarrow \infty$ e $k/n \rightarrow 0$, erro se aproxima do mínimo
- Lento:
  - Versões mais simples precisam examinar inteiramente os dados de treinamento para efetuar a previsão
- Assume que todos os atributos são igualmente importantes
  - Solução: seleção de atributos ou uso de pesos
- Possíveis soluções contra instâncias ruidosas:
  - Usar uma maioria de votos sobre os k vizinhos mais próximos
  - Remover instâncias ruidosas do conjunto de dados

29

Regressão Localmente Ponderada (RLP)

- Generalização de vizinhos mais próximos
- Constrói uma aproximação explicita de uma função $f(x_q)$ em uma região próxima de x_q

30

Regressão Localmente Ponderada (RLP)

- Localmente
  - A aproximação é definida na vizinhança de x_q e servirá exclusivamente para sua classificação
- Ponderada
  - A contribuição de cada instância é ponderada pela distância entre estas e x_q
- Regressão
  - Designa o problema de encontrar uma função de aproximação

31

Regressão Localmente Ponderada (RLP)

- Descrição
  - Construir uma aproximação $\hat{f}(x)$ que ajuste os valores das instâncias de treinamento na vizinhança de x_q.
  - A aproximação é então usada para calcular o valor ponto x_q.
  - A descrição de $\hat{f}$ é apagada, pois a função de aproximação será construída para cada instância a ser consultada

32

Regressão Localmente Ponderada (RLP)

- Função de aproximação mais comum

$$\hat{f}(x) = \omega_0 + \omega_1 a_1(x) + \ldots + \omega_n a_n(x)$$

- Escolher ω_i que minimiza a soma dos quadrados dos erros em relação ao conjunto de treinamento D

$$\mathrm{E}(x_q) = \frac{1}{2} \sum_{x \in D} \left(f(x) - \hat{f}(x) \right)^2$$

33

Regressão Localmente Ponderada (RLP)

- Outras propostas para minimizar o erro:
  - Erro quadrático sobre os k-vizinhos mais próximos

$$E(x_q) = \frac{1}{2} \sum_{x \in \text{ k vizinhos mais próximos de } x_q} \left(f(x) - \hat{f}(x) \right)^2$$

34

Regressão localmente ponderada

- Erro quadrático ponderado em D

$$E(x_q) = \frac{1}{2} \sum_{x \in D} \left(f(x) - \hat{f}(x) \right)^2 K(d(x_q, x))$$

- Onde $K(d(x_q,x))$ é uma função que penaliza grandes distâncias entre os pontos

Regressão localmente ponderada

- Combinação das duas anteriores

$$E(x_q) = \frac{1}{2} \sum_{x \in \text{ k vizinhos mais próximos de } x_q} \left(f(x) - \hat{f}(x) \right)^2 K(d(x_q, x))$$

Regressão localmente ponderada

- Várias funções para cálculo da distância
  - Distância euclidiana bastante usada

- Várias funções de aproximação
  - Constante, linear e quadrática
  - Funções mais complexas são evitadas
    - Custo de ajustamento
    - As funções mais simples fornecem aproximações boas sobre uma região suficientemente pequena do espaço de instâncias