Aprendizado de Máquina:
Métodos para Avaliação de Classificadores

Prof. Arnaldo Candido Junior
UTFPR – Medianeira

Estimativa de erro

- Várias medidas podem ser usadas para avaliar um classificador
  - Acurácia (será usada como exemplo)
  - Taxa de erro (bastante comum nos exemplos da literatura)
  - Precisão, cobertura, medida-f, …
- Regressão/agrupamento: tem suas próprias medidas: erros quadrático médio, silhueta, …

Estimativa de erro (2)

- Preferências de times extraídos de uma turma com 30 alunos reflete preferência dos brasileiros?
- Será que as medidas obtidas (ex. acurácia) são confiáveis? Elas são importantes para:
  - Tratar under/overfitting
  - Comparar modelos induzidos (ex.: duas versões de uma mesma rede neural)
  - Comparar algoritmos (ex.: rede neural vs SVM)

Estimativa de erro (3)

- Geralmente não é possível medir com exatidão a acurácia do modelo, deve ser **estimada**
- Não é possível usar o próprio conjunto de treino para calcular as medidas: propenso a **overfitting**
- Estratégia: dividir dataset em subconjuntos de treinamento, teste e, opcionalmente, validação (se o algoritmo permitir)

Métodos

- Principais estratégias:
  - Hold-out
  - Randon subsampling
  - Cross validation
  - Leave-one-out
  - Bootstrap

Hold-out

- Indicado para grandes datasets. Por exemplo, mais de 1000 instâncias (isso varia de problema para problema)
- Também conhecido como split-sample, é a técnica mais simples para estimativa de erro
- Faz uma divisão única do dataset e roda o algoritmo uma única vez
- Divisões típicas: 1/2, 2/3, 3/4 ou 4/5 das instâncias para treino e o restante para teste

Hold-out (2)

- Pequena quantidade de dados: resultados não são muito confiáveis. Exemplo para acurácia:
  - Conjunto de treino pequeno: algoritmo pode cometer underfitting e acurácia estimada pode ficar baixa
  - Conjunto de teste pequeno: estimativa da acurácia é menos confiável, variância alta, acurácia real pode ser muito diferente

Hold-out (3)

- Idealmente, classes devem estar igualmente distribuídas em treinamento e teste
  - Se a classe A aparecer mais no treino do que no teste, classificador será pouco testado com uma classe que ele aprendeu bem
  - Se a classe A aparecer mais no teste do que no treino, classificador será muito testado com uma classe que ele aprendeu pouco
  - Resultado: estimativa fica pessimista

Random Subsampling

- Indicado para datasets médios ou pequenos, pois pode obter uma estimativa mais precisa da acurácia
- Ideia geral: definir vários conjuntos de teste e de treino usando amostragem e treinar vários classificadores
- Idealmente, um conjunto de treino e seu respectivo teste são **partições** do dataset, ou seja, não possuem instâncias repetidas
  - Amostragem sem reposição

Random Subsampling (2)

- Processo: dividir o dataset em treino e teste **n** vezes, treinando um classificador a cada iteração e calculando acurácia
- Acurácia real é estimada a partir da média das acurácias das diferentes partições
- Desvio padrão pequeno é um indicativo que a acurácia real é próxima da estimativa obtida

Random Subsampling (3)

- Exemplo para o dataset: $\hat{x}_1, \hat{x}_2, \hat{x}_3, \hat{x}_4, \hat{x}_5, \hat{x}_6, \hat{x}_7, \hat{x}_8$

	Treino	Validação	Teste
Part. 1	$\hat{x}_2, \hat{x}_4, \hat{x}_6, \hat{x}_7$	$\hat{x}_5, \hat{x}_8$	$\hat{x}_1, \hat{x}_3$
Part. 2	$\hat{x}_3, \hat{x}_4, \hat{x}_5, \hat{x}_8$	$\hat{x}_1, \hat{x}_7$	$\hat{x}_2, \hat{x}_6$
Part. 3	$\hat{x}_3, \hat{x}_4, \hat{x}_5, \hat{x}_7$	$\hat{x}_2, \hat{x}_8$	$\hat{x}_1, \hat{x}_6$

Random Subsampling (4)

- O algoritmo original não controla número de vezes que uma instância é usada para treinamento e teste
- Algumas instâncias podem ser mais utilizados para treinamento que outras
  - O mesmo se aplica ao teste

K-Fold Cross-Validation

- Indicado para datasets médios ou pequenos
  - Um dos métodos mais populares para avaliar classificadores
- Cada instância participa o mesmo número de vezes no treinamento, e apenas uma vez do teste
- Processo: dividir dataset em **k** partições mutuamente exclusivas
- A cada iteração, uma delas é usada para testar o modelo e as demais **k-1** partições para treinar

K-Fold Cross-Validation (2)

- Normalmente, k=10 (validação cruzada de 10 partes)
- Exemplo para k=3
  - Classificador 1: treinado sobre a partição 1 e 2 e testado com a partição 3
  - Classificador 2: treino usando 1 e 3 e teste usando 2
  - Classificador 3: treino usando 2 e 3 e teste usando 1

K-Fold Cross-Validation (3)

- Dataset: $\{\hat{x}_1, \hat{x}_2, \hat{x}_3, \hat{x}_4, \hat{x}_5, \hat{x}_6\}$
- Partição 1: $\{\hat{x}_1, \hat{x}_2\}$
- Partição 2: $\{\hat{x}_3, \hat{x}_4\}$
- Partição 3: $\{\hat{x}_5, \hat{x}_6\}$

	Treino+Validação	Teste
Rodada 1	$\hat{x}_1, \hat{x}_2, \hat{x}_3, \hat{x}_4$	$\hat{x}_5, \hat{x}_6$
Rodada 2	$\hat{x}_1, \hat{x}_2, \hat{x}_5, \hat{x}_6$	$\hat{x}_3, \hat{x}_4$
Rodada 3	$\hat{x}_3, \hat{x}_4, \hat{x}_5, \hat{x}_6$	$\hat{x}_1, \hat{x}_2$

Leave-One-Out

- Indicado para datasets muito pequenos
  - Equivale ao k-fold cross validation com k do tamanho do dataset inteiro
  - A cada iteração, uma instância é utilizado para testar o modelo e as n-1 restantes para o treinamento
  - O número de modelos induzidos é igual ao número de instâncias do dataset

Leave-One-Out (2)

- Sua estimativa de erro é praticamente não tendenciosa
- Computacionalmente caro se for utilizado em datasets maiores
  - 10-fold cross validation aproxima razoavelmente bem o leave-one-out em boa parte dos datasets

Leave-One-Out (3)

- Pode ser pessimista se dataset está balanceado
  - Vai haver um leve desbalanceamento ao remover-se uma instância para teste
  - Classificador que sempre vicia na classe majoritária nunca não vai acertar classe da instância de teste

Validação Cruzada Repetida

- Roda a validação cruzada várias vezes
  - Mudando a forma de criar a partição em cada rodada
  - Mais caro: um classificador é induzido em cada parte de cada validação cruzada
  - Objetivo: obter uma estimativa mais confiável da acurácia

Validação Cruzada Repetida (2)

- Variante popular: 5x2 cross validation (Dietterich, 1998)
  - 5 rodadas
  - Cada uma dividindo teste e validação em duas partições de mesmo tamanho (50-50)
  - Testes empíricos sugerem que 5 rodadas é um bom valor

Bootstrap

- Indicado para datasets muito pequenos
- Funciona melhor que cross-validation para neste tipo de dataset
  - Com poucas instâncias, conjunto de treinamento pode ser muito diferente do conjunto de teste
  - Acurácia estimada pode ser muito **pessimista**

Bootstrap (2)

- Forma mais simples de bootstrap:
  - Ao invés de usar subconjuntos dos dados, usar subamostras (pode repetir instâncias)
  - Usa sorteio com reposição para gerar vários conjuntos de treino, cada um têm o mesmo tamanho do dataset original
  - Conjunto de treino é usado para testar
  - Pode ser muito **otimista**

Bootstrap (3)

- Bootstrap 632:
  - Estatisticamente, cada conjunto de treino contém ≈ 63,2% das instâncias originais
  - As demais 36,8% são usadas para teste
  - Acurácia de cada conjunto de teste pode ser **pessimista**
  - Usa acurácia de cada conjunto de treino para balancear esse efeito

Bootstrap (4)

- Cálculo usado Bootstrap 632:

$$acc_{boot} = \frac{1}{b}\sum_{i=1}^{b} 0.632\, acc(teste_i) + 0.368\, acc(treino_i)$$

  - Resultados empíricos bons, porém pode ser otimista caso o classificador memorize o conjunto de treino

Créditos

- Adaptado de:
  - Notas de aula do Prof. Dr. André C. P. L. F. de Carvalho - ICMC-USP