Universidade Federal do Ceará
Departamento de Estatística e Matemática Aplicada
Introdução à Mineração de Dados (CC0101), período 2019.1
Professor: Tibérius O. Bonates (`tb@ufc.br`).

Atividade 6 – Classificação semi-supervisionada com aprendizado por auto-treinamento.

1 Descrição

Nesta atividade, você irá construir um novo classificador usando o `scikit-learn`. Seu código irá treinar um tipo de classificador pré-existente utilizando um conjunto de dados que contendo tanto observações que possuem informação de classe quanto observações que **não** possuem informação de classe. Isto é, o conjunto de treinamento é da forma $D = D_A \cup A_B$:

$$D_A = \{(\mathbf{x}^i, y^i) : i = 1, \ldots, m_A\} \text{ e } D_B = \{\mathbf{x}^i : i = m_A + 1, \ldots, m_A + m_B\}.$$

Este tipo de situação de aprendizado é chamada de *aprendizado não-supervisionado*. A estratégia de aprendizado que seu código deve usar é a de *auto-aprendizado*: o programa iterativamente atribui classe às observações de D_B, com base em informação aprendida a partir de D_A e das observações de D_B que já tiveram uma classe atribuída.

Auto-aprendizado é um tipo de *wrapper* (invólucro) sobre um classificador existente. Seja `F` o classificador-base existente que você deseja estender para o caso semi-supervisionado. Para atribuir uma classe para a observação $\mathbf{x}^{m_A+1}$ seu código deve treinar um classificador `F` sobre D_A e utilizá-lo para classificar $\mathbf{x}^{m_A+1}$. Seja c^{m_A+1} o valor que você obtém para $\texttt{F}(\mathbf{x}^{m_A+1})$. O conjunto D_A é aumentado com uma nova observação:

$$D_A := D_A \cup \{(\mathbf{x}^{m_A+1}, c^{m_A+1})\},$$

e m_A é incrementado em uma unidade: $m_A := m_A + 1$. O processo é repetido para a classificação de $\mathbf{x}^{m_A+2}$ a partir de um classificador `F` treinado sobre o conjunto D_A atualizado. Novamente, o conjunto $_A$ é atualizado para conter a observação $(\mathbf{x}^{m_A+2}, y^{m_A+2})$, e oprocesso é repetido até que uma informação de classe tenha sido atribuída a cada exemplo de D_B. A esta altura, um novo classificador `F` deve ser treinado sobre o conjunto D_A final, que contém $m_A + m_B$ exemplos. É este último classificador que será utilizado para classificar novos exemplos.

Uma variação deste algoritmo diz respeito ao número de exemplos que D_B que são classificados e adicionados a D_A em cada iteração. A descrição acima é uma das formas possíve sd e proceder: apenas uma observação é classificada em cada iteração. No outro extremo, podemos realizar o processo em uma única iteração, na qual todas as observações de D_B são classificadas de acordo com o classificador aprendido sobre D_A. De forma geral, podemos fazer esse processo com conjuntos contendo $1 \leqslant \texttt{k} \leqslant m_B$ observações. Seu código deve aceitar um parâmetro `k` e deve atribuir novas classes a `k` observações de D_B em cada iteração (naturalmente, na última iteração é possível que tenhamos menos que `k` obsevações, se m_b não for múltiplo de `k`).

O usuário do seu novo classificador vai fornecer como parâmetro um classificador base `F` e um valor inteiro positivo `k`. O parâmetro referente ao vetor de classes do método `fit` do seu classificador conterá valores iguais a `-1` para identificar as **observações sem informação de classe**. Como exemplo, assuma que seu novo classificador é implementado em uma classe de nome `ClassificadorSemiSuper`. Seu código deve aceitar o seguinte tipo de uso:

```
>>> clf = ClassificadorSemiSuper(classificador, 3) # grupos de 2 exemplos
>>> clf.fit(observacoes, classe) # 'classe' poderá conter valores iguais a -1
>>> clf.score(teste, classeTeste)
     0.8716
```

Você poderá utilizar como ponto de partida o código visto em sala para criação de um novo classificador, a partir do zero, usando o `scikit-learn`, disponível no SIGAA: data 02/05/2019, item "classificador da classe mais frequente".

2 Entrega

- Pontuação total pela atividade: 4,0 pontos (quatro pontos).
- Entregáveis (**total de TRÊS entregáveis**):
  - Código Python.
  - Arquivo contendo o conjunto de dados utilizado nos testes do relatório, para que o professor possa rodar seu código e replicar seu resultados.
  - Relatório em formato PDF (não será aceito outro formato), contendo uma tabela que mostre o desempenho do seu classificador com o conjunto de dados escolhido. O desempenho de seu código deve ser medido em termos de um mesmo classificador-base, variando apenas o parâmetro `k` (utilize, no mínimo, 10 valores diferentes para `k` em sua tabela).

    Para cada valor de `k`, você deve utilizar um só experimento de validação, do tipo *hold-out*, com 66% de treinamento e 33% de teste. Os 66% de treinamento devem ter metade dos dados (isto é, $\sim$ 33% do total) sem informação de classe. Ou seja, dos 66% de exemplos que consttituem o conjunto de treinamento, metade deve ter classe igual a `-1` no vetor de classes fornecido ao método `fit`.
- Data de entrega: 27 de junho de 2019, 23:59:59, via SIGAA ou via e-mail.
- Data de entrega com adiamento automático: Não tem.

3 Mais detalhes

- A atividade pode ser feita individualmente ou em dupla.