MAC0460 – EP5

DCC / IME-USP — Primeiro semestre de 2019

Entrega até: 17/06

Objetivos: O objetivo do EP5 é exercitar o uso da biblioteca `scikit-learn`[1] (https://scikit-learn.org), multiclassificação (SVM e redes neurais), e validação cruzada.

Tarefa: Em resumo, neste EP deve ser utilizado um subconjunto do MNIST para treinar, validar e testar os algoritmos SVM (*Support Vector Machine*) e MLP (*Multi Layer Perceptron*). A validação dever ser feita usando a estratégia de *cross-validation*, para comparar os dois algoritmos. O teste deve ser feito com o algoritmo que obteve melhor desempenho no *cross-validation*. Os detalhes estão descritos a seguir.

Dados a serem usados: Vamos usar o **dataset do MNIST** (o mesmo do EP4), porém desta vez vamos consider múltiplas classes (especificamente, as classes de 0 a 4).

O conjunto de teste é o próprio conjunto de teste do MNIST, porém restritos às classes de interesse. O conjunto de treinamento+validação deverá ser construído de forma a conter 500 exemplos de cada classe de interesse.

- Utilize os seguintes nomes para o dataset original MNIST:

  `(x_train,y_train)` e `(x_test,y_test)`, como no EP4

- Selecione apenas dados das classes de interesse e crie os datasets `(X_train,Y_train)` e `(X_test,Y_test)` que serão efetivamente utilizados nos experimentos.

  - `X_train`: deve conter 500 exemplos de cada classe j, $j = 0, 1, 2, 3, 4$, aleatoriamente selecionados de `x_train`. Cada exemplo em `X_train` deve estar no formato "flattened" (*array* unidimensional) e com valores normalizados para o intervalo [0,1].
  - `X_test`: deve ser construído de forma similar ao `X_train`, exceto pelo fato de que deverá conter todos os exemplos das classes $j = 0, 1, 2, 3, 4$ que estão em `x_test`
  - o array de rótulos `Y_train` deve conter os rótulos correspondentes aos exemplos que estão em `X_train` (idem para `Y_test` com respeito a `X_test`)

- embaralhe `(X_train,Y_train)` — este é o dataset que será usado para treinamento+validação

Cross-validation: Sim, o `scikit-learn` já tem o *cross-validation* implementando, mas ele não poderá ser usado neste EP. Neste EP vocês deverão implementar a técnica de *five-fold cross-validation* e avaliar os classificadores SVM e redes neurais. Deve ser calculado o *cross-validation accuracy* de cada um deles. Dado que o conjunto de treinamento `(X_train,Y_train)` já foi embaralhado, os *folds* podem ser os primeiros 1/5 dos dados, o segundo 1/5 dos dados, e assim por diante.

[1] É uma biblioteca Python que implementa vários algoritmos e técnicas usadas em *machine learning*.

Para treinar os classificadores SVM e rede neural, pode-se utilizar a implementação disponível no `scikit-learn`.

- **SVM**: use o kernel RBF e escolha um valor para os parâmetros γ e C. Para os demais parâmetros, sugerimos os valores *default*. Valor referência: $\gamma = 0.05$ e $C = 5$.
- Redes neurais do tipo MLP (*multi-layer perceptron*): construa uma rede neural com duas camadas ocultas, e escolha o número de nós nas camadas ocultas e o *learning rate* inicial. Para os demais parâmetros, sugerimos os valores *default*.

Você pode alterar alguns parâmetros e fazer o *cross-validation* repetidas vezes. Mas não verifique o desempenho dele no conjunto de teste, nunca, enquanto estiver ajustando os parâmetros.

Teste final: Escolha o algoritmo (e parâmetros) com melhor desempenho no *cross-validation*, treine-o novamente sobre o conjunto de treinamento completo `(X_train,Y_train)` e calcule a acurácia com respeito ao conjunto de teste `(X_test,Y_test)`

O que entregar Entregar um relatório sucinto descrevendo como foi implementado cada uma das partes acima e contendo as informações importantes que permitam entender os dados usados e os resultados obtidos. Inclua no seu relatório ao menos as seguintes informações:

- *shape* de `X_train` e `Y_train`
- *shape* de `X_test` e `Y_test`
- valor mínimo e máximo de `X_train`, `Y_train`, `X_test` e `Y_test`
- Em cada iteração do *cross-validation*, quantos exemplos de cada classe há no *fold* de validação e a acurácia obtida.
- *cross-validation accuracy* (média das acurácias por *fold*)
- características do conjunto de teste e desempenho do algoritmo escolhido com respeito ao conjunto de teste (acurácia e matriz de confusão)

Comente os resultados obtidos. Caso tenha testado valores diferentes para os parâmetros, inclua uma descrição no relatório. Se você fizer o EP em um *notebook* e ele estiver bem organizado com comentários e *prints* suficientes, pode entregar um `html` ou `pdf` do seu *notebook* como relatório (cuidado com a legibilidade).

Além disso, entregar o código implementado (*notebook* ou o programa em arquivo `.py`). O código só será executado/considerado caso as informações presentes no relatório forem ambíguas ou inconclusivas.

A organização, clareza e completude do relatório vale até 20% da nota total.

Postem as dúvidas e discussões no "Fórum de discussão" do PACA.

Também recomendamos que compartilhem no Fórum os resultados obtidos pelo seu modelo final no conjunto de teste (mas depois disso não convém fazer novos ajustes de parâmetros).