

Primeira Parte:

Objetivo: Entender o funcionamento da biblioteca LibSVM através da execução de experimentos de classificação.

1) Baixe e instale a libSVM no seu diretório. Disponível nos links abaixo:

<http://www.csie.ntu.edu.tw/~cjlin/libsvm/>

2) Compile o código para gerar os executáveis
(Verifique se o gnuplot esta instalado!)

> **apt-get install gnuplot**

3) Ler o documento disponível no link abaixo:

<http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf>

4) Baixe as bases de aprendizagem e teste disponíveis no moodle no mesmo pacote:

arquivoTreino.svm

arquivoTeste.svm

5) Analise as bases de treinamento e teste e reporte no relatório:

a) Número de classes ;

b) Número de Instâncias no Treinamento ;

c) Número de Instâncias no Teste ;

6) Execute o script python que acompanha a libSVM, chamado easy.py (**localizado no diretório tools**). Esse script faz a busca pelos parâmetros do kernel Gaussiano (g) e da variável de custo (C). O script gera alguns arquivos. Liste quais são esses arquivos e explique o conteúdo dos mesmos .

> **.easy treino.vet teste.vet**

6.1) Reporte a taxa de reconhecimento no arquivo de teste.

6.3) Reporte o número de vetores de suporte encontrados para cada classe e o total;
(arquivo.vet.predict → NSV)

6.4) Utilize o conjunto de treinamento para treinar um modelo e o mesmo para testar;

6.5) Inverta, utilize o conjunto de teste como treinamento e o de treinamento como teste ;

7) Utilize os parâmetros encontrados pelo easy.py para treinar um svm que estima probabilidades. Para fazer isso você deve usar a opção -b e o comando svm-train. Use também os arquivos normalizados gerados pelo easy.py

> **.svm-train -b 1 -C ?? -g ?? arquivoTreino ArquivoSaida.modelo**

8) Utilize o modelo aprendido "**ArquivoSaida.modelo**" para estimar probabilidade no passo abaixo e utilize o mesmo para reconhecer a base de teste. O arquivo de teste também deve ser o arquivo normalizado.

> **.svm-predict -b 1 arquivoTeste.scale ArquivoSaida.modelo Resultado.predict**

7.1) Compare a taxa de reconhecimento com os resultados do item 5.

Alguma diferença relevante?

7.2) Dê uma idéia de como os valores das probabilidades podem ser utilizados para tornar o sistema de classificação mais robusto.

7.3) Repita a etapa 7 utilizando diferentes kernels (por padrão o SVM utiliza o RBF) ;

7.4) Veja qual deles apresentou o melhor resultado ;

Segunda Parte:

Os dados dos arquivos “treino2.svm” e “teste2.svm”, são descritores de textura extraídos de cartas manuscritas de diferentes N pessoas. Utilizou-se o LBP para extrair estas features.

Utilizando o libSVM avalie:

- 1) Taxa de acerto usando o easy.py (configuração padrão usando RBF) ;
- 2) Taxa de acerto usando gerando modelos com outros kernels (0 , 1, 3):

```
-t kernel_type : set type of kernel function (default 2)
0 -- linear:  $u \cdot v$ 
1 -- polynomial:  $(\gamma u \cdot v + \text{coef0})^{\text{degree}}$ 
2 -- radial basis function:  $\exp(-\gamma |u-v|^2)$ 
3 -- sigmoid:  $\tanh(\gamma u \cdot v + \text{coef0})$ 
```

- 3) A taxa de acerto caso utilize caso inverta os arquivos para treinar e testar (utilizar o “treino2.svm” para teste e “teste2.svm” para treino) ;
- 4) A taxa de acerto usando estes dados no k-NN com $k = 3$;
- 5) A taxa de acerto usando estes dados na Árvore de Decisão (com poda) ;