

## Parte I - Máquinas de Vetores de Suporte

Este projeto verificará o funcionamento de máquinas de vetores de suporte, tanto com *kernel* linear quanto Gaussiano.

- O arquivo *dado1.mat* contém um conjunto de exemplos, com 2 atributos, e linearmente separáveis. Quando este dado é carregado, terá acesso às variáveis  $X$  (atributos) e  $y$  (classes). Teste a implementação de SVM do *sklearn* como visto no código da aula (ou alguma outra implementação pronta no caso de usar Matlab/Octave). Use  $C = 1$ ,  $C = 50$  e  $C = 100$  e mostre a fronteira de decisão sobre o *scatterplot* em cada caso. Note que o dado tem um *outlier* que pode ser identificado visualmente. Discuta no relatório o resultado obtido, o papel de  $C$  e como o *outlier* influenciou no processo.
- Já o arquivo *dado2.mat* contém dados que não são linearmente separáveis e por isso então deverá fazer uso de SVM com *kernels*. Agora, além das variáveis  $X$  e  $y$ , terá acesso também a  $X_{val}$  e  $y_{val}$ , que serão usados para validação. Use a mesma implementação pronta do item anterior para treinar um modelo de SVM com *kernel* Gaussiano.
- Use o conjunto de validação para determinar  $C$  e  $\sigma$  (do kernel Gaussiano) ideais. Sugere-se que teste os valores  $\{0.01, 0.03, 0.1, 0.3, 1, 3, 10, 30\}$  para ambos os hiperparâmetros. Mostre a fronteira de decisão sobre o *scatterplot* para o  $C$  e  $\sigma$  ótimos obtidos.

## Parte II - Análise de Componentes Principais

Uma das aplicações mais interessantes e surpreendentes de análise de componentes principais é no reconhecimento facial (*eigenfaces*). Nesta parte, você implementará este algoritmo para um pequeno conjunto de imagens de rostos.

- Carregue os dados do arquivo *dado3.mat*. A variável  $X$  contém uma matriz em que cada linha corresponde a imagens de faces em tons de cinza, com tamanho original  $32 \times 32$ , mas convertidas para um vetor de 1024 componentes. Mostre 100 imagens quaisquer desta base (lembre-se de reconverter o vetor linha para uma imagem). Implemente o algoritmo de PCA como visto em aula e rode sobre esta base de faces.
- Os componentes PCA (autovetores) principais neste tipo de aplicação são o que se chama de *eigenfaces* e eles podem ser visualizados se

forem reformatados para uma imagem  $32 \times 32$ . Mostre em seu código as *eigenfaces* correspondentes aos 36 primeiros componentes principais. Comente no relatório sobre o que observa nestas imagens.

- Projete a matriz  $X$  sobre os 100 componentes principais e reconstrua o dado original sobre esta projeção. Mostre as imagens originais de 100 faces ao lado das imagens reconstruídas. Comente no relatório sobre o que achou da reconstrução e implicações práticas deste processo.