Aprendizagem Automática

Trabalho Laboratorial – grupos de 3 alunos

Classificação de Faces

1º Semestre de 2017/2018

Objetivos do trabalho: Este trabalho consiste na classificação/identificação de faces. Para este efeito, será usada a base de dados "labeled faces in the wild". O trabalho está dividido em várias etapas, as quais se descrevem resumidamente de seguida.

1. Projetar e avaliar o desempenho do classificador dos k vizinhos mais próximos (kNN).
2. Estimar qual o número ótimo de vizinhos.
3. Repetir a avaliação do desempenho usando faces processadas com PCA
4. Estimar qual o número ótimo de componentes principais.
5. Verificar se normalizar a variância dos dados transformados é benéfico.
6. Testar e avaliar outro classificador além do kNN, nos dados transformados e comparar os resultados com os do kNN.
7. Projetar e avaliar o desempenho de um classificador à sua escolha para o problema de identificação/verificação de uma das seguintes personalidades: George W. Bush ou Colin Powell ou Tony Blair. De notar que este é um problema de classificação binária e deverá ter isso em conta na avaliação.

Dados: As imagens de faces pertencem à base de dados "labeled faces in the wild". Esta base de dados consiste em imagens de faces de celebridades transferidas da Internet, e inclui faces de políticos, atletas, atores e cantores do início do último milénio. Esta base de dados pode ser descarregada através do módulo de Python, scikit-learn usando os seguintes comandos:

```
>>> from sklearn.datasets import fetch_lfw_people
>>> carasBD=fetch_lfw_people(min_faces_per_person=30,resize=0.75)
```

O último comando seleciona faces com, no mínimo, 30 exemplos, e reduz o seu tamanho para 75% do original, sendo o tamanho final das imagens de 93×70 pixeis. A variável `carasBD` é um dicionário composto por 2370 imagens de faces de 34 personalidades[1]. A seguinte tabela contém o número associado ao individuo\classe (de 0 a 33), o seu nome, e o número de imagens suas no dicionário.

0.	Alejandro Toledo	39	12.	Hans Blix	39	24.	Megawati Sukarnoputri	33
1.	Alvaro Uribe	35	13.	Hugo Chavez	71	25.	Nestor Kirchner	37
2.	Andre Agassi	36	14.	Jacques Chirac	52	26.	Recep Tayyip Erdogan	30
3.	Ariel Sharon	77	15.	Jean Chretien	55	27.	Roh Moo-hyun	32
4.	Arnold Schwarzenegger	42	16.	Jennifer Capriati	42	28.	Serena Williams	52
5.	Colin Powell	**236**	17.	John Ashcroft	53	29.	Silvio Berlusconi	33
6.	David Beckham	31	18.	John Negroponte	31	30.	Tom Ridge	33
7.	Donald Rumsfeld	**121**	19.	Junichiro Koizumi	60	31.	Tony Blair	**144**
8.	George W Bush	**530**	20.	Kofi Annan	32	32.	Vicente Fox	32
9.	Gerhard Schroeder	**109**	21.	Laura Bush	41	33.	Vladimir Putin	49
10.	Gloria Macapagal Arroyo	44	22.	Lleyton Hewitt	41			
11.	Guillermo Coria	30	23.	Luiz Inacio Lula da Silva	48			

[1] Também disponibilizado no ficheiro `lfw_raw.p`

Os dados estão bastante enviesados havendo certas classes com um grande número de imagens, como é o caso de George W. Bush, de Colin Powell e outros. Este facto prejudica o treino de classificadores, porque estes ficariam a dar preferência às classes mais populosas. Para tornar os dados menos enviesados, limite o número de imagens por pessoa a cinquenta (50).

Metodologias de Teste: Descreva detalhadamente a(s) estratégia(s) usadas para testar e avaliar os classificadores. De notar que, caso utilize métodos de validação cruzada ou afins (ex: "shuffle split") haverá várias estimativas da probabilidade total de erro e da matriz de confusão. Dado as restrições de espaço na elaboração do relatório (ver mais à frente), expresse estas métricas de desempenho em termos da média dos resultados.

Etapas do Trabalho: Para diferentes etapas do trabalho enumeradas no início do enunciado, tenha em conta o seguinte:

- As imagens estão guardadas no dicionário acedendo à chave `'images'` (array de $2370 \times 93 \times 70$ dimensões), mas também podem ser acedidas pela chave `'data'` (array de 6510×2370). Ao limitar o número de exemplos por classe a 50 imagens, o número total de imagens passa a ser 1410.

1.+2. Nos testes relativos a estes pontos, normalize os dados de modo a terem média nula e variância unitária.

3.+4. Estime a transformação PCA com todos os dados. Antes de projetar as imagens nas suas componentes principais, remova a média dos dados. Não é necessário repetir os testes para estimar o número ótimo de vizinhos mais próximos – use o valor estimado anteriormente com as imagens não processadas.

5. Usar os dados projetados com o número ópimo de componentes principais, e normalizar a variância de cada dimensão de moda a esta ser igual a 1. Verificar se os resultados do kNN melhoram com este passo de processamento.

6. Testar outro classificador do módulo `scikit-learn`. De seguida estão os comandos necessários para carregar os classificadores disponíveis:

```
from sklearn.neural_network import MLPClassifier
from sklearn.svm import SVC
from sklearn.gaussian_process import GaussianProcessClassifier
from sklearn.gaussian_process.kernels import RBF
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier, AdaBoostClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.discriminant_analysis import QuadraticDiscriminantAnalysis
```

Teste um destes classificadores e compare com os resultados obtidos com o kNN. Deve usar dados processados com PCA com o número ótimo de componentes principais. O classificador kNN deve igualmente usar o número óptimo de vizinhos. Verificar também

se processar os dados de modo a terem variância unitária (ponto anterior) é benéfico para este novo classificador. No relatório descreva o funcionamento do classificador.

7. Este ponte é referente a um problema de classificação binária: uma dada imagem é ou não da personalidade escolhida. A base de dados está agora dividida em duas classes, sendo a classe dos positivos as imagens da personalidade escolhida e a classe dos negativos todas as outras imagens. Deverá ter igualmente em conta os dois tipos de erros cometidos, e como minorar um ou outro tipo de erros.

Elaboração do Relatório:

- O relatório terá no máximo 8 páginas e deverá ser formatado de acordo com o modelo "ACM_Large" da revista ACM - Association for Computer Machinery (ver página http://www.acm.org/publications/authors/submissions). Os membros do grupo devem estar claramente identificados com número e nome no campo referente aos autores, e o relatório deverá ser bem estruturado: ter uma introdução, uma conclusão, descrição dos métodos usados, das experiências efetuadas, resultados obtidos, referências usadas, etc.
- A descrição das experiências feitas deve seguir a ordem dada neste enunciado. Deve brevemente descrever cada testes (dados usados, se foram processados com PCA, o classificador usado, etc), reportar detalhadamente os resultados obtidos.
- Tendo em conta o espaço limitado do relatório, é preferível sempre que possível, descrever os resultados através de gráficos. Por exemplo, para determinar o valor óptimo do número de vizinhos, pode apresentar um gráfico em que as abcissas são o número de vizinhos e as ordenadas é a média de acertos (ou erros) para esse número de vizinhos. O mesmo se aplica ao número óptimo de componentes principais.
- Matrizes de confusão e probabilidades totais de erro (ou acerto) devem ser expressas em termos da média dos resultados obtidos. Adicionalmente, visto haver 34 classes, não é prático expressar as matrizes de confusão numa tabela; utilize antes uma imagem de 34×34 pixeis.
- Não inclua no relatório o código implementado.
- Deve comentar os resultados obtidos e as possíveis causas para os (bons/maus) desempenhos, e quando achar pertinente, complementar o seu raciocínio com gráficos ou imagens.
- Inclua na bibliografia todo o material consultado para elaborar o relatório.
- IMPORTANTE: Entregar unicamente o ficheiro do relatório. Terá que ser um ficheiro ".pdf" com o nome: `TP1_Axxxx_Axxxx_Axxxx.pdf`
  onde os "`Axxxx`" correspondem aos números de aluno dos 3 membros do grupo. Colocar os número em ordem crescente.