Universidade do Minho - Escola de Engenharia

Perfil de Machine Learning: Fundamentos e Aplicações

Métricas em Machine Learning

Aplicação do PCA em reconhecimento de dígitos

Catarina Alexandra Carvalho de Almeida pg37015
Elisa Maria Maio Araújo pg40157
Nuno Valente a81986
Paulo Filipe Silva Ribeiro a79845
Paulo Jorge Costa da Conceição Mendes a78203
Rui Miguel Fontes pg39296

12 de Janeiro de 2020

1 Introdução

A existência de conjuntos de dados com enumeros atributos faz com que, visualiza-los, seja impossível. Para tal, é necessário reduzir o número de variáveis do mesmo. E, uma das técnicas mais utilizadas na redução de dimensionalidade é, *Principal Component Analysis, PCA*. Esta técnica consiste em realçar as semelhanças e diferenças neles existentes através da identificação de padrões. A sua identificação em dados caraterizados por grandes dimensões é difícil, uma vez que a sua representação gráfica não é viável, logo uma análise visual aos dados não é possível. Quando identificados os padrões no conjunto, o número de dimensões a analisar pode ser reduzido sem que haja uma perda significativa de informação, pois o foco recai sobre a análise das dimensões principais que caracterizam o conjunto de dados.

Para o desenvolvimento deste trabalho prático foi utilizado o *dataset MNIST*, que contém imagens manuscritas. Este está dividido num conjunto de dados de treino e de teste. O conjunto de dados de treino contém imagens 28×28 de dígitos manuscritos, e por legendas que indicam qual dos dígitos, entre 0 e 9, está a ser representado. O conjunto de dados de teste é estruturado exatamente do mesmo modo, à exceção que contém apenas 10000 registos. Estas imagens foram pré processadas e são apresentadas numa escala de cinzentos, para que sejam facilmente ajustadas nas dimensões pedidas. Em *Machine Learning* um atributo é uma característica única de uma observação que é usada para criar o modelo suposto. Neste caso, os *pixels* da imagem serão esses atributos, que ajudarão a desenvolver o modelo. De um modo geral, o objetivo é descobrir padrões entre a intensidade dos *pixels* da imagem e o dígito exibido na mesma. Em vez de pensar numa imagem 28×28 normal, podemos pensar nesta como um vetor ou uma lista de 728 colunas.

2 Definições

Seja A uma matriz $n \times n$, define-se o polinómio característico de A como

$$\Delta_A(\lambda) = det(\lambda I - A)$$

.

As raízes de $\Delta_A(\lambda)$ chamam-se **valores próprios** de A.

Se λ é valor próprio de A, então $(\lambda I - A) \cdot v = 0$ é um sistema possível indeterminado, ou seja, existe vetor $v \neq 0$ tal que $(\lambda I - A) \cdot v = 0$.

Para um valor próprio λ, existe $v \neq 0$ tal que

$$Av = \lambda v$$

.

v diz-se **vetor próprio** de A associado a λ.

Chama-se **Matriz Covariância** à matriz:

$$Cov = B^T \cdot B$$

onde B é a matriz com os dados de treino centrados na média, e B^T a sua correspondente matriz transposta.

Sejam v e w dois vetores. A **Distância Euclediana** é definida do seguinte modo:

$$d(v, w) = ||v - w||_2 = \sqrt{\langle v - w, w - v \rangle}$$

A **Distância de Mahalanobis** em $\mathbb{R}^n$ é definida do seguinte modo:

$$d_M(\begin{bmatrix} c_1 \\ \vdots \\ c_k \end{bmatrix}, \begin{bmatrix} c_i^i \\ \vdots \\ c_n^i \end{bmatrix}) = \sum_{j=1}^{k} \frac{1}{\lambda j}(c_j - c_j^i)^2$$

onde $c_1, ..., c_k$ e $c_1^i, ..., c_k^i$ são os coeficentes de projeção dos dados de treino e teste, respetivamente.

3 Desenvolvimento PCA

Para a aplicação desta técnica é necessário o executar os seguintes passos:

3.1 Centrar na média

O primeiro passo para o desenvolvimento desta técnica consiste numa normalização dos dados, isto é, subtrair a média de cada uma das dimensões que caracterizam o conjunto de dados, de modo a obter um novo conjunto, cuja a média é 0.

3.2 Cálculo da matriz covariância

A variância e a covariância são duas medidas utilizadas em estatística. A variância indica quão dispersos estão os dados de uma dimensão em relação à média, ignorando a existência de outras dimensões. A covariância, por sua vez, aplica-se a duas dimensões, permitindo assim perceber como ambas estão relacionadas entre si. Esta medida indica quanto as dimensões variam em relação à média, tendo em conta a relação que existe entre elas. De modo a percebermos como é que todas as dimensões do conjunto de dados se relacionam entre si, é necessário calcular as covariâncias entre todas as dimensões do conjunto. Uma forma útil de obter todos os valores de covariância possíveis entre todas as diferentes dimensões do conjunto de dados consiste em calculá-las e escrevê-las sob a forma de uma matriz. Para isso é usada a fórmula que calcula a matriz de covariância, indicada na Secção 2.

Na Figura 1, pode-se observar a matriz covariância para o *dataset* de treino.

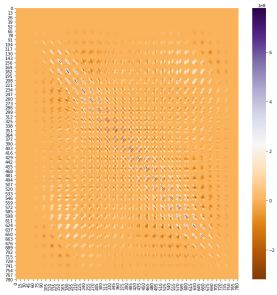


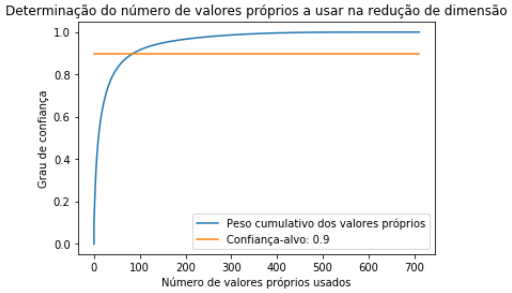
Figura 1: Matriz Covariância

3.3 Cálculo dos valores e vetores próprios

Os vetores próprios e os valores próprios representam as características principais de uma matriz. Serão calculados os valores e vetores próprios da matriz de covariância determinada anteriormente. Para o seu calculo são usadas as fórmulas indicadas na Secção 2.

3.4 Cálculo do valor da redução

Com o intuito de reduzir a dimensão dos dados, determinou-se o número de vetores próprios mais relevantes,K. O que indica o número de direcções a usar nas projeções dos dados. Para tal, partiu-se de uma confiança-alvo de 90% e determinou-se que o número de vetores próprios a utilizar é de $K = 87$.



Consultando os pesos relativos de cada valor próprio através da aplicação do método do cotovelo, valida-se a opção inicial. $K = 87$ corresponde ao fim do cotovelo, zona onde a adição consecutiva de valores próprios à analise tem pouca influência da exatidão do modelo.

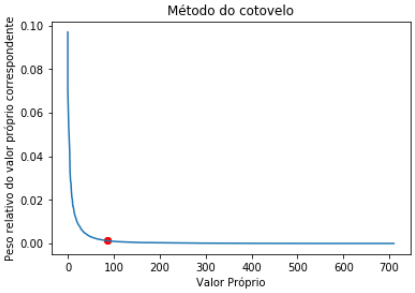


Figura 2: Método do Cotovelo

3.5 Projeção dos dados

Após os valores e vetores próprios calculados, bem como o valor de K, é feita a projeção dos dados. Esta é feita através da multiplicação da matriz transposta que contém os vetores próprios resultantes da redução de dimensão, isto é, os mais relevantes, com a matriz transposta que contém os dados de treino centrados na média, ou seja, $V^T \cdot B^T$, sendo V a matriz dos vetores próprios resultantes da redução de dimensão e B a matriz com os dados de treino centrados na média.

4 Reconhecimento de dígitos

O reconhecimento de dígitos é feito através do cálculo de distâncias entre os dados de treino projetados e os dados de teste projetados. E, o digito é reconhecido no passo seguinte. Isto é, após o cálculo da distância entre o digito que se pretende calcular e cada um dos digitos do *dataset* de treino, determina-se qual o elemento do *dataset* de treino com a distância mínima ao digito. Esse elemento é considerado igual ao digito que se pretende identificar e pela consulta do nome desse digito, obtêm-se o nome do digito a identificar. Foram utilizadas para o calculo das distâncias, a distância Euclideana e a distância de *Mahalanobis*. As fórmulas usadas para este cálculo encontram-se representadas na Secção 2. São esperados melhores resultados com a distância de *Mahalanobis*, pois intuitivamente como pretendemos minimizar a distância, dá-se mais importância ás entradas correspondentes aos maiores valores próprios.

4.1 Aplicação das distâncias

Para as duas métricas usadas na avaliação da exatidão do modelo, a distância Euclidiana e a distância de *Mahalanobis*, verifica-se que para um grau de confiança de 90%, a exatidão do modelo encontra-se perto do máximo absoluto. Fez-se uma comparação entre o cálculo da exatidão do modelo para o *dataset* de teste completo, 10000 unidades, e uma amostra aleatória de 500 uni-

dades, a diferença era inferior a 1%. Como tal, para a análise seguinte usou-se a abordagem das 500 unidades para reduzir o esforço computacional.

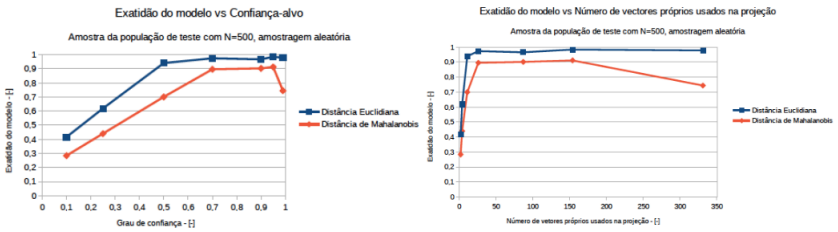


Figura 3

Grau de confiança - [-]	Número de valores próprios	Exatidão		Tempo de execução - [s]
		Distância Euclidiana	Distância de Mahalanobis	
0,99	331	0,978	0,744	44,60
0,95	154	0,984	0,912	17,12
0,9	87	0,966	0,902	9,35
0,7	26	0,974	0,896	3,99
0,5	11	0,94	0,7	2,98
0,25	4	0,618	0,44	2,50
0,1	2	0,416	0,284	2,58

Figura 4

Os dados da análise mostram que aumentar o número de vetores próprios usados na análise melhora a exatidão do modelo. O máximo de exatidão estará algures entre os 0.9 de confiança e os 0.99, o máximo relativo foi obtido para os 0.95. Neste caso o número de valores próprios óptimo usados na análise (métrica análoga ao nível de confiança) depende de um balanço entre a exatidão pretendida e o esforço computacional permitido. O esforço computacional aumenta com o aumento do número de vetores próprios usados. Tentou-se quantificar esse esforço computacional através do tempo de execução do código para a análise de uma imagem. A análise mostra um aumento exponencial do tempo com o aumento do número de dimensões. Entre 0,9 e 0,95 de confiança, o tempo de execução praticamente duplica (1,8). Como se pode observar na Figura 4, a distância *Eucledíana* foi a que obteve um maior número de reconhecimentos, ao contrário daquilo que se estava a espera.

4.2 Exemplo de Reconhecimento do dígito

Na Figura 5 podemos observar um caso onde o programa reconhece os dígitos 2 e 9.

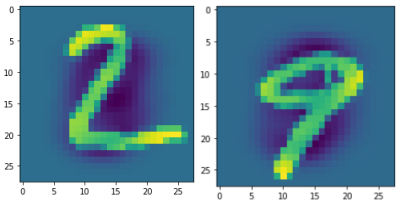


Figura 5: Exemplos em que o programa reconheceu o dígito

4.3 Exemplos de não reconhecimento

Na Figura 6 o modelo desenvolvido reconhece a imagem como sendo um 9, quando na realidade é um 4.

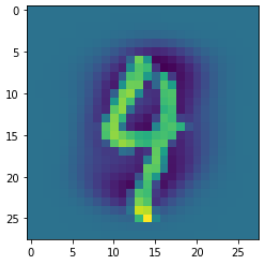


Figura 6: Número 4

Na Figura 7 o modelo desenvolvido reconhece a imagem como sendo um 7, quando na realidade é um 8.

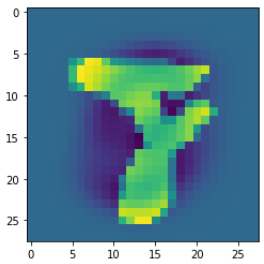


Figura 7: Número 8