

國立臺灣大學資訊管理學系研究所
資料探勘
小組作業一

**基於PCA、NMF、LDA降維
之KNN、SVM、DecisionTree、
RandomForest分類模型準確度比較**

第二組

蘇達立、蕭溥辰、王辰豪

中華民國 109 年 03 月 25 日

目錄

目錄	1
壹、作業概述	2
貳、研究動機與目的	2
參、資料集	2
肆、實驗方法	3
伍、實驗結果與分析	4
陸、結論	11
柒、附錄	14

壹、作業概述

在本次作業中我們採用PCA、NMF和LDA對MNIST datasets做降維的處理，並針對每種降維方法分別做KNN、SVM、DecisionTree、RandomForest的分類，比較三者套用不同分類方法後的Accuracy、Precision和Recall。

貳、研究動機與目的

基於課堂上資料降維度的理論所學，我們想針對該理論進程式碼的實作，了解如何將資料降低維度，並了解資料降維後對原有資料的視覺影響為何，最後根據實作結果知道各種降低維度方法的效能以及在不同分類器下，準確率之優劣。因為手寫數字辨識在分類問題上，是相當知名的問題，且具有充足和乾淨的資料集，因此，在本次作業中，我們以手寫數字辨識來當作我們本次作業的題目。

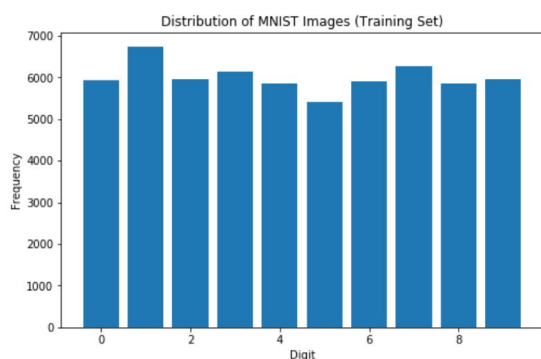
參、資料集

我們從Kaggle網站中獲取MNIST資料集作為我們的datasets，資料集中共有60000筆資料作為訓練資料集，10000筆資料作為測試資料集，資料內容情形如圖一所示。每一橫列的一筆資料即代表一張圖片，每張圖片為28*28像素的灰階照片，每一張灰階照片代表了手寫符號的一個數字(0~9)，資料欄位中的label說明該數字為何。其他欄則為像素資料，顯示其組成灰階圖片的像素結構，以空格區分出784個（28*28個像素）坐落在0至255之間，顯示圖片自左上到右下各像素的灰階色彩。

	label	1x1	1x2	1x3	1x4	1x5	1x6	1x7	1x8	1x9	...	28x19	28x20	28x21	28x22	28x23	28x24	28x25	28x26	28x27	28x28
0	7	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1	2	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
2	1	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
4	4	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0

(圖一) datasets 內容

此外MNIST的手寫數字辨識datasets資料分布情形，如圖二圖三所示



(圖二) 訓練datasets 分布情形



(圖三) 測試datasets 分布情形

我們可以看出訓練和測試的資料集，數字的分布比例十分接近。

肆、實驗方法

1. 使用套件：

我們使用sklearn中的PCA、NMF和LDA三種降維度套件和KNN、SVM、DecisionTree、RandomForest四種分類套件作為本次作業的數據分析工具，並且使用matplotlib來呈現datasets視覺化效果。

2. 作法採用：

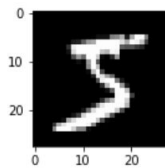
首先，我們先將28*28的灰階圖像，經過三種不同的降維方法，分別是PCA、NMF及LDA，再將降維過後的圖像，經過不同的分類方法進行分類，在此作業中，我們選用了四種不同的分類模型，包含KNN、SVM、DecisionTree和RandomForest，並且針對各種分類模型使用Accuracy、Precision和Recall來檢視成果。

伍、實驗結果與分析

(1) PCA:

- 原始影像與經過PCA降維影像之比較(以數字5為例)

原始影像:



PCA不同維度降維之影像:

	PCA(1)	PCA(5)	PCA(10)	PCA(30)
原始影像與經過PCA降維影像之比較				

	PCA(50)	PCA(100)	PCA(200)	PCA(500)
原始影像與經過PCA降維影像之比較				

- 透過PCA降至1維，經過四種分類器(KNN、SVM、DecisionTree、RandomForest)之結果

PCA(1)	SVM	KNN	DecisionTree	RandomForest
Accuracy	0.31510	0.27380	0.24870	0.24880
Precision	0.20797	0.23564	0.23945	0.23955
Recall	0.30372	0.26384	0.24015	0.24026

- 透過PCA降至5維，經過四種分類器(KNN、SVM、DecisionTree、RandomForest)之結果

PCA(5)	SVM	KNN	DecisionTree	RandomForest
Accuracy	0.77280	0.74750	0.66930	0.76280
Precision	0.77361	0.74491	0.66513	0.76162
Recall	0.77083	0.74453	0.66499	0.76029

- 透過PCA降至10維，經過四種分類器(KNN、SVM、DecisionTree、RandomForest)之結果

PCA(10)	SVM	KNN	DecisionTree	RandomForest
Accuracy	0.93640	0.92760	0.82380	0.91390
Precision	0.93549	0.92678	0.82163	0.91301
Recall	0.93556	0.92666	0.82130	0.91270

- 透過PCA降至30維，經過四種分類器(KNN、SVM、DecisionTree、RandomForest)之結果

PCA(30)	SVM	KNN	DecisionTree	RandomForest
Accuracy	0.98030	0.97540	0.84900	0.95380
Precision	0.98031	0.97543	0.84674	0.95319
Recall	0.98014	0.97522	0.84685	0.95341

- 透過PCA降至50維，經過四種分類器(KNN、SVM、DecisionTree、RandomForest)之結果

PCA(50)	SVM	KNN	DecisionTree	RandomForest
Accuracy	0.98330	0.97490	0.84340	0.95540
Precision	0.98329	0.97498	0.84108	0.95496
Recall	0.98320	0.97468	0.84111	0.95500

- 透過PCA降至100維，經過四種分類器(KNN、SVM、DecisionTree、RandomForest)之結果

PCA(100)	SVM	KNN	DecisionTree	RandomForest
Accuracy	0.98410	0.97270	0.84030	0.95150
Precision	0.98406	0.97285	0.83788	0.95109
Recall	0.98403	0.97248	0.83771	0.95116

- 透過PCA降至200維，經過四種分類器(KNN、SVM、DecisionTree、RandomForest)之結果

PCA(200)	SVM	KNN	DecisionTree	RandomForest
Accuracy	0.98290	0.96980	0.83020	0.94620
Precision	0.98291	0.97021	0.82806	0.94565
Recall	0.98281	0.96948	0.82770	0.94561

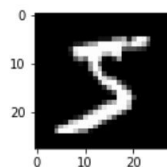
- 透過PCA降至500維，經過四種分類器(KNN、SVM、DecisionTree、RandomForest)之結果

PCA(500)	SVM	KNN	DecisionTree	RandomForest
Accuracy	0.98240	0.96870	0.82000	0.92300
Precision	0.98242	0.96917	0.81752	0.92233
Recall	0.98233	0.96837	0.81726	0.92183

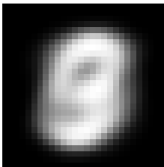
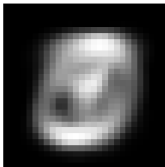
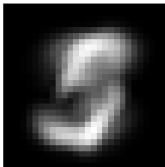
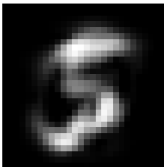


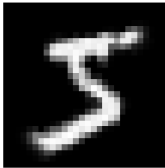
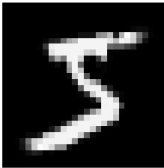
(2) NMF:

原始影像與經過NMF降維影像之比較(以數字5為例)

原始影像:



NMF不同維度降維之影像:

	NMF(1)	NMF(5)	NMF(10)	NMF(30)
原始影像與降維後影像比較				
	NMF(50)	NMF(100)	NMF(200)	NMF(300)
原始影像與降維後影像比較				

NMF(1)	SVM	KNN	DecisionTree	RandomForest
Accuracy	0.20456	0.1648	0.147	0.1471
Precision	0.12574	0.14344	0.14469	0.14477
Recall	0.19622	0.15803	0.14325	0.14335

NMF(5)	SVM	KNN	DecisionTree	RandomForest
Accuracy	0.7054	0.667	0.6116	0.7039
Precision	0.69935	0.65992	0.60954	0.70086
Recall	0.70283	0.66336	0.60770	0.70121

NMF(10)	SVM	KNN	DecisionTree	RandomForest
Accuracy	0.8815	0.8647	0.7993	0.8872
Precision	0.87984	0.86226	0.79706	0.88551
Recall	0.87974	0.86246	0.79634	0.88556

NMF(30)	SVM	KNN	DecisionTree	RandomForest
Accuracy	0.9715	0.9558	0.8811	0.9596
Precision	0.97139	0.95561	0.87952	0.95934
Recall	0.97125	0.95529	0.87940	0.95924

NMF(50)	SVM	KNN	DecisionTree	RandomForest
Accuracy	0.9781	0.9508	0.8697	0.9655
Precision	0.97800	0.95108	0.86892	0.96533
Recall	0.97804	0.94998	0.86795	0.96521

NMF(100)	SVM	KNN	DecisionTree	RandomForest
Accuracy	0.9765	0.9508	0.8663	0.9654
Precision	0.97649	0.95108	0.86436	0.96513
Recall	0.97641	0.94997	0.86438	0.96509

NMF(200)	SVM	KNN	DecisionTree	RandomForest
Accuracy	0.9739	0.9396	0.8676	0.9644
Precision	0.97372	0.93995	0.86617	0.96421
Recall	0.97366	0.93834	0.86599	0.96404

NMF(500)	SVM	KNN	DecisionTree	RandomForest
Accuracy	0.967	0.9091	0.8425	0.9624
Precision	0.96678	0.91142	0.84042	0.96209
Recall	0.96679	0.90778	0.84055	0.96221

(3) LDA

由於LDA中需要計算類別間散度矩陣 S_b 和類別內散度矩陣 S_w 。當原有資料降維到矩陣 W ，有 n 行和 d 列，而矩陣 W 的列是 $S_w^{-1}S_b$ 的特徵向量。而 S_b 的行最大為 $k-1$ ，所以最多有 $k-1$ 個特徵向量。所以矩陣 W 最多只有 $k-1$ 列。因此當原有資料有 C 個類別時，LDA至多可生成 $C-1$ 維空間，因此LDA降維後的維度區間介於 $[1, C-1]$ 。由於在MNIST datasets中資料的label介於 $[0, 9]$ ，所以在LDA中只能降維度到 $[1, 9]$ 。

LDA(1)	SVM	KNN	DecisionTree	RandomForest
Accuracy	0.42428	0.54613	0.99942	0.99908
Precision	0.42522	0.23564	0.99815	0.99911
Recall	0.43018	0.26384	0.99815	0.99907

LDA(2)	SVM	KNN	DecisionTree	RandomForest
Accuracy	0.58840	0.66622	1.0	0.99997
Precision	0.55013	0.23564	1.0	0.99981
Recall	0.53647	0.26384	1.0	0.99982

LDA(3)	SVM	KNN	DecisionTree	RandomForest
Accuracy	0.75473	0.79293	1.0	0.99995
Precision	0.75618	0.23564	1.0	0.99983
Recall	0.75471	0.26384	1.0	0.99983

LDA(4)	SVM	KNN	DecisionTree	RandomForest
Accuracy	0.83844	0.86293	1.0	0.99997
Precision	0.83721	0.23564	1.0	0.99972
Recall	0.83167	0.26384	1.0	0.99973

LDA(5)	SVM	KNN	DecisionTree	RandomForest
Accuracy	0.85435	0.8784	1.0	0.99997
Precision	0.85316	0.23564	1.0	0.99972
Recall	0.85432	0.26384	1.0	0.99971

LDA(6)	SVM	KNN	DecisionTree	RandomForest
Accuracy	0.87757	0.89695	1.0	1.0
Precision	0.85631	0.23564	1.0	1.0
Recall	0.87163	0.26384	1.0	1.0

LDA(7)	SVM	KNN	DecisionTree	RandomForest
Accuracy	0.90168	0.91895	1.0	1.0
Precision	0.90631	0.23564	1.0	1.0
Recall	0.90364	0.26384	1.0	1.0

LDA(8)	SVM	KNN	DecisionTree	RandomForest
Accuracy	0.90526	0.93342	1.0	1.0
Precision	0.90521	0.23564	1.0	1.0
Recall	0.90427	0.26384	1.0	1.0

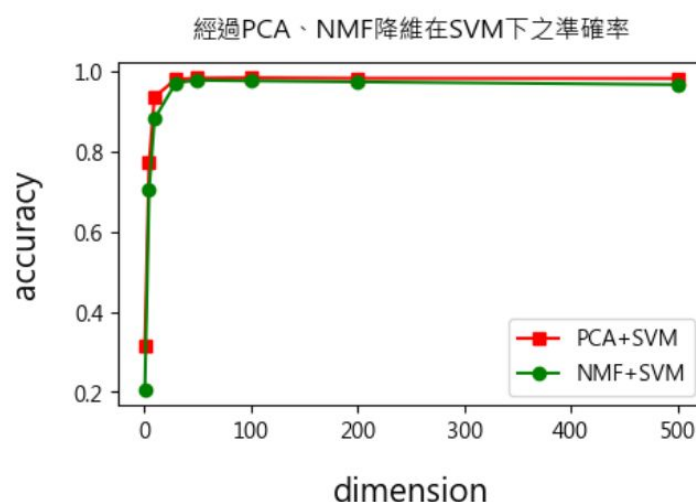
LDA(9)	SVM	KNN	DecisionTree	RandomForest
Accuracy	0.92631	0.93873	1.0	1.0
Precision	0.92314	0.23564	1.0	1.0
Recall	0.93011	0.26384	1.0	1.0

陸、 結論

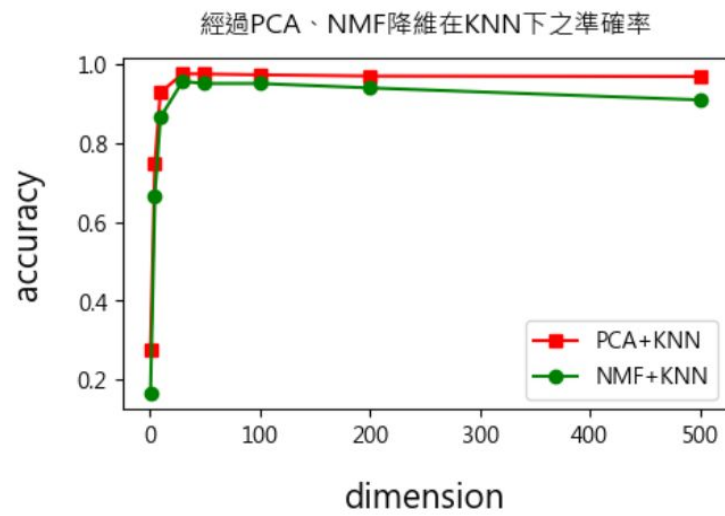
PCA降維的實驗結果，我們可以明顯看出，不管是在accuracy、precision還是recall，SVM的分類結果都比KNN、DecisionTree、RandomForest還要突出。而在降低維度的部分，也可以清楚看出，從一開始一維的準確率(0.31510)，隨著維度增加，準確率也不斷上升，在一百維時，準確率來到(0.98410)，但在這之後，維度增加卻導致準確率下降，在五百維時，準確度降到0.98240，因此，我們可以得知，降維不但可以去除資料中的雜質，降低資料的大小，還可以提升準確率。

NMF降維的實驗結果，可以看出，在accuracy、precision、recall的表現上面，使用SVM相較於我們選擇的其他三種分類方法表現得較為突出，從低維度到50維的時候，準確率明顯的提升許多，但是隨著維度增加，可以觀察到準確度甚至出現了些微的下降。因此我們可以得知，透過NMF降維，隨著維度的增加，可以有效地提升分類的準確率。

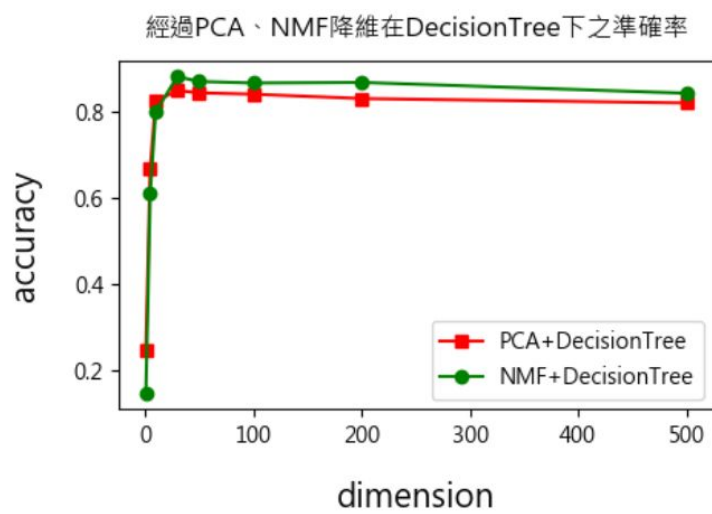
LDA降維的實驗結果，我們可以看出DecisionTree和RandomForest的分類結果比SVM和KNN還要突出。而在降低維度的部分，如圖七到圖十所示，我們可以清楚地看出，以SVM為例，從一開始一維的準確率(0.42428)，隨著維度增加，準確率也不斷上升，在最後九維時，準確率來到(0.92631)，在KNN、DecisionTree和RandomForest的趨勢亦同，因此，我們可以得知，透過LDA降維隨著降低的維度越高，可以有效提升分類準確率。



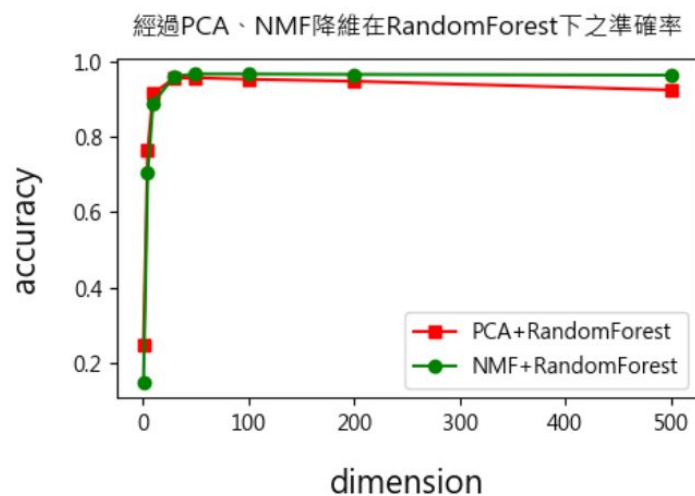
(圖四)Accuracy for PCA+SVM、NMF+SVM



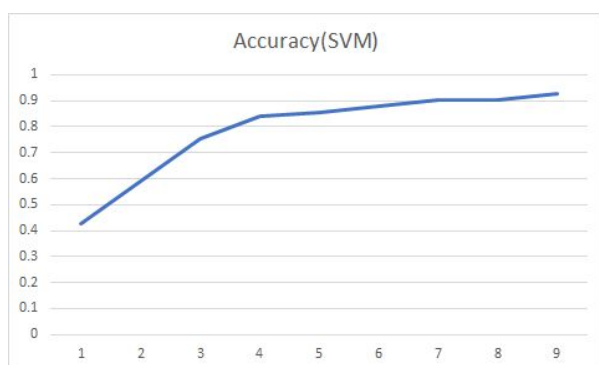
(圖五) Accuracy for PCA+KNN、NMF+KNN



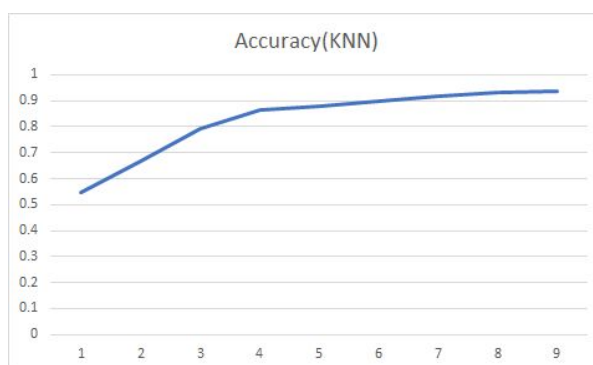
(圖六) Accuracy for PCA+DecisionTree、NMF+DecisionTree



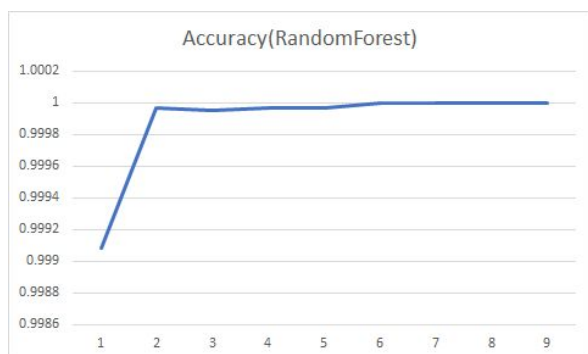
(圖七) Accuracy for PCA+RandomForest、NMF+RandomForest



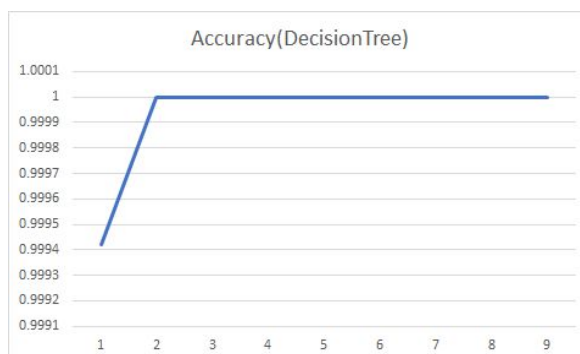
(圖八) Accuracy for LDA+SVM



(圖九) Accuracy for LDA+KNN



(圖十) Accuracy for LDA+RandomForest



(圖十一) Accuracy for LDA+DecisionTree

柒、 附錄

1. Kaggle MNIST datasets

https://www.kaggle.com/oddrational/mnist-in-csv?fbclid=IwAR1PtcRFYIpFjGg3MvhOLSEB-W8cW6bU_xlbNIT9Uv624b8TrlhOShemrjg

2. 實作程式碼

(1). LDA

https://github.com/cfs9805804/DataMining/blob/master/DM_HW1_LDA.ipynb

(2). PCA

https://github.com/jonathanaa/data_mining_class/blob/master/hw1.ipynb

(3). NMF

<https://gist.github.com/44d0285ddf74f60a7d44fdef0ff1405e.git>