Exploring kNN, MLE and SVM Classification Techniques

Applied to Letter Image Recognition Data

CS559 – Machine Learning: Fundamentals and Applications (Dr. Mordohai)

Matt Hall

Overview

- This project use 3 classification methods to classify a set of randomly distorted letters.
- Prior to classification Primary Component Analysis will be performed. This will allow to hopefully reduce dimensionality of the data.
- Maximum Likelihood Estimation, k-Nearest Neighbor (multiple k-values) and Support Vector Machine classification techniques will be used.
- The outcome will be the accuracy of the classification.

Dataset

- The data samples were taken from the UCI Machine Learning Repository - https://archive.ics.uci.edu/ml/datasets/Letter+Recognition
- The full set consists of 20,000 records
- Total of 26 classes, representing an English letter of the alphabet
- The character images were based on 20 different fonts and each letter within this font was randomly distorted to produce the 20,000 unique stimuli
- Each stimulus was converted into the 16 primitive numerical attributes, which were then scaled to fit integers ranging from 0 to 15

Attribute List

lettr	capital letter
x.box	horizontal position of box
y.box	vertical position of box
width	width of box
high	height of box
onpix	total number of on pixels
x.bar	mean x of on pixels in box
y.bar	mean y of on pixels in box
x2bar	mean x variance
y2bar	mean y variance
xybar	mean x y correlation
x2ybr	mean of *x^2 y*
xy2br	mean of *x y^2*
x.ege	mean edge count left to right
xegvy	correlation of x.ege with y
y.ege	mean edge count bottom to top
yegvx	correlation of y.ege with x

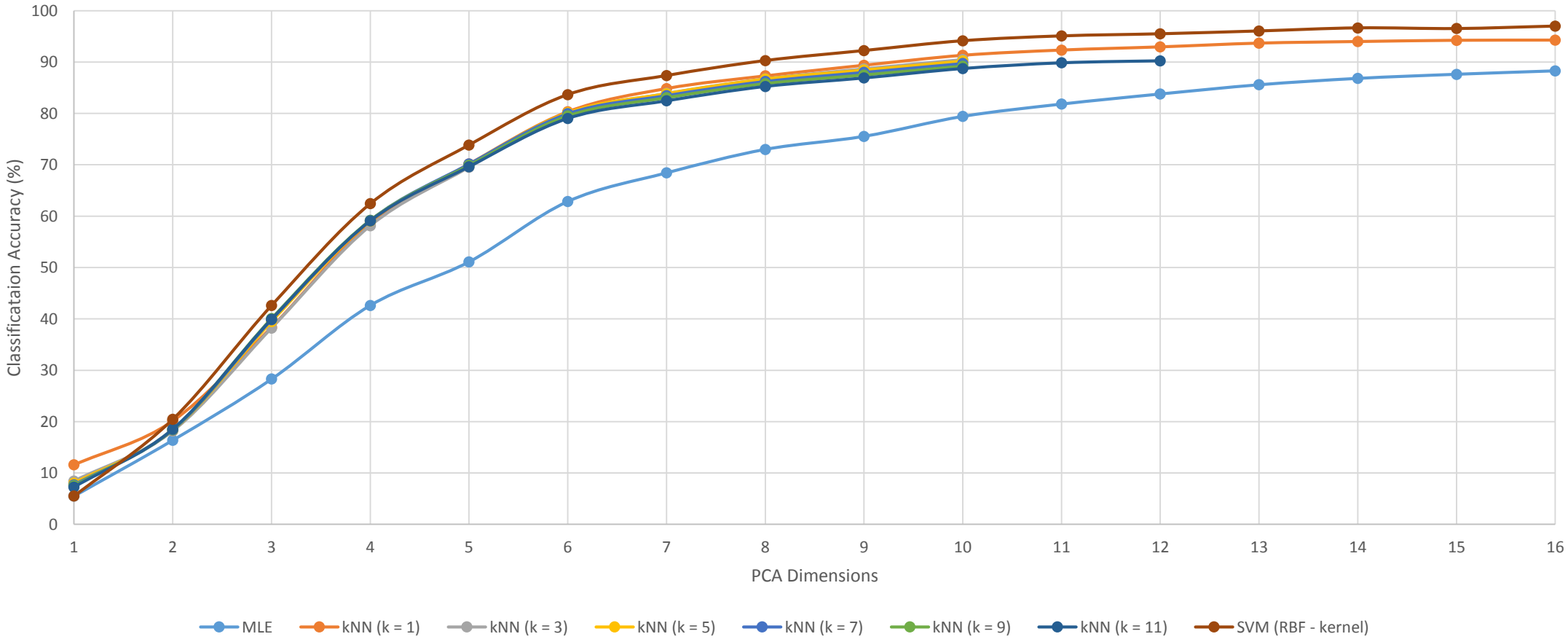
Tests

- Data was split in half with training being done on the first half and testing on the second half.
- Primary component analysis was used on the native dataset and prioritized, then tests were run on only the first component, then the first two components, and so on, with the last test having all 16 primary components included.
- k-Nearest Neighbor was performed with k values of 1, 3, 5, 7, 9 and 11.
- The accuracy reported is an average over 10 runs with each run being a different randomized set of the native data.
- In each of the 10 runs, each classification technique was performed using the exact same set of training and testing data.

Results

	MLE	kNN (k = 1)	kNN (k = 3)	kNN (k = 5)	kNN (k = 7)	kNN (k = 9)	kNN (k = 11)	SVM (RBF - kernel)
PCA Dimension(s)	Accuracy	Accuracy						Accuracy (%)
1	5.5	11.59	8.43	8.03	7.69	7.49	7.25	5.51
2	16.38	20.16	18.19	18.32	18.38	18.48	18.54	20.47
3	28.28	38.39	38.2	39.47	39.83	40.05	39.92	42.61
4	42.62	58.74	58.13	59.05	59.18	59.17	59.07	62.42
5	51.1	70.14	69.52	70.05	70.14	69.88	69.62	73.85
6	62.85	80.32	79.95	80.13	79.94	79.49	79.01	83.66
7	68.42	84.84	83.89	83.84	83.47	83.04	82.44	87.37
8	72.99	87.3	86.74	86.69	86.22	85.78	85.24	90.28
9	75.53	89.37	88.69	88.44	87.98	87.47	86.89	92.23
10	79.41	91.31	90.49	90.2	89.71	89.19	88.73	94.14
11	81.81	92.33					89.84	95.08
12	83.77	92.95					90.23	95.49
13	85.57	93.68						96.04
14	86.81	93.99						96.64
15	87.61	94.23						96.51
16	88.28	94.27						96.99

Overall Plot



Conclusions

- MLE performed reasonably well, however, the results are not interesting when compared against kNN and SVM, except for MLE performed much faster than kNN.
- kNN was the slowest by far, but performed well and slightly behind SVM. kNN showed some strange results in the k = 1 on average performed better than higher k-values. This should be investigated further, with one way being to try kNN without PCA.
- SVM performed the best and was much faster than kNN. Some further improvements should be investigated.
- All classification methods stopped seeing much improvement after the inclusion of the first 10 or 11 PCA dimensions.