Letter Recognition

Bautista, Dianne Carrol
Chen, Hongshu
Paul, Rajib
Jun 8th, 2004

Outline

- Introduction
- Preliminary analysis
- Results & Discussions
  - 1-NN
  - LDA
  - SVM
- Conclusions

Introduction

- Multi-Class Recognition Problem
- Objective:

to classify each of a large number of black and white rectangular pixel displays as one of the 26 capital letters of the English alphabet.

Data Description

- Source: David Slate (Jan 1991)

  Odesta Corporation, Evanston, IL 60201
- Size: **20,000** (obs) X **17** (variables)
- Details:

  -based on 20 different fonts

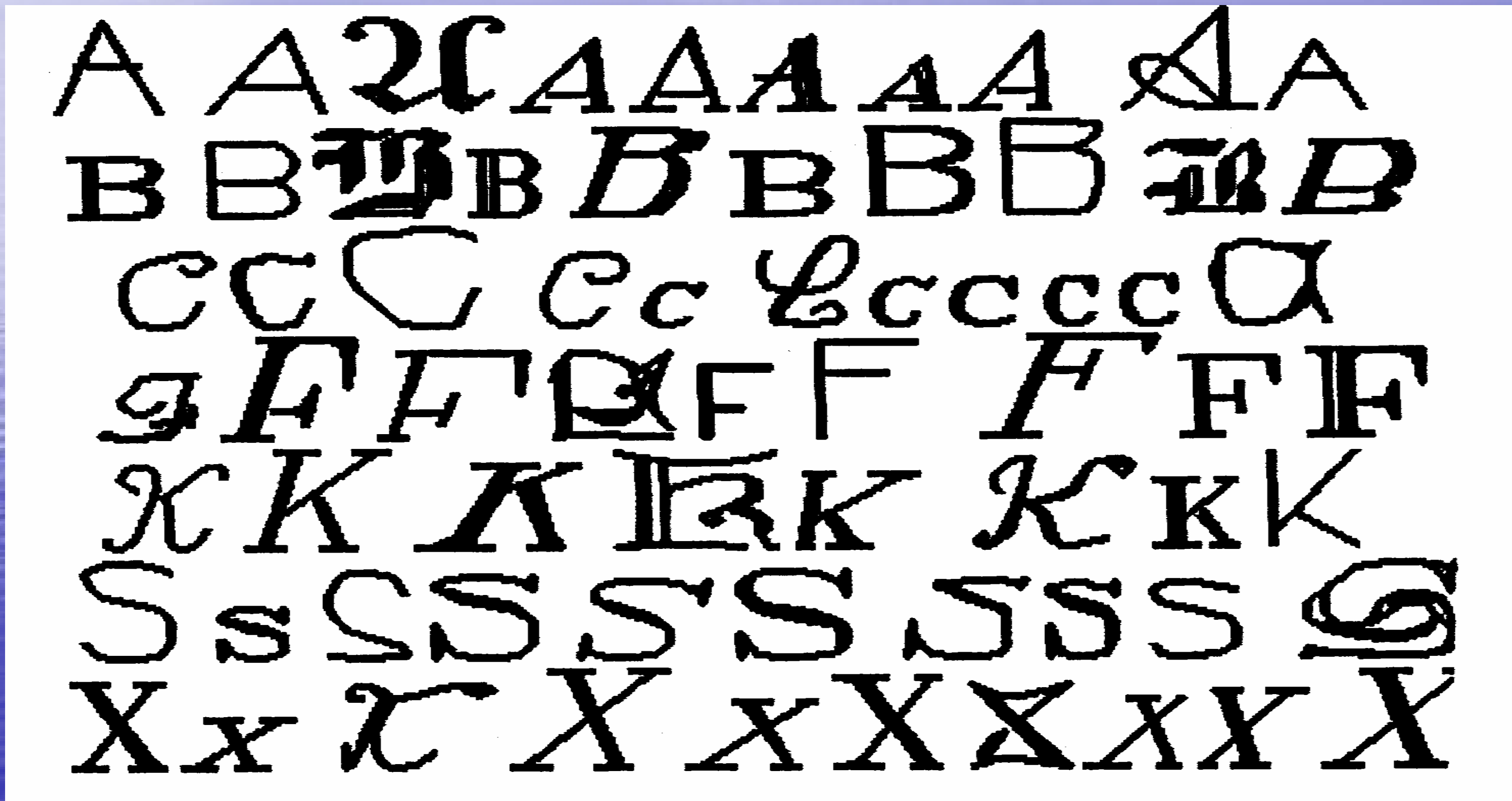
  -representing five different stroke styles--simplex, duplex, triplex, complex, and Gothic

  -and six different letter styles--block, script, italic, English, German, and Italian.

Generation of Character Images

- calls made to a character-image generating program with uniformly distributed parameter values for font type, letter of the alphabet, linear magnification, aspect ratio, and horizontal & vertical warp
- character image represented in terms of the vector coordinates of the end-points of its constituent line segments
- Warping was applied to these coordinates
- line segments were then converted to raster format forming a rectangular array of pixels, each of which was “on” or “off”
- The totality of “on” pixels represented the image of the desired character
- average dimension of the arrays was 45 pixels high by 45 pixels wide

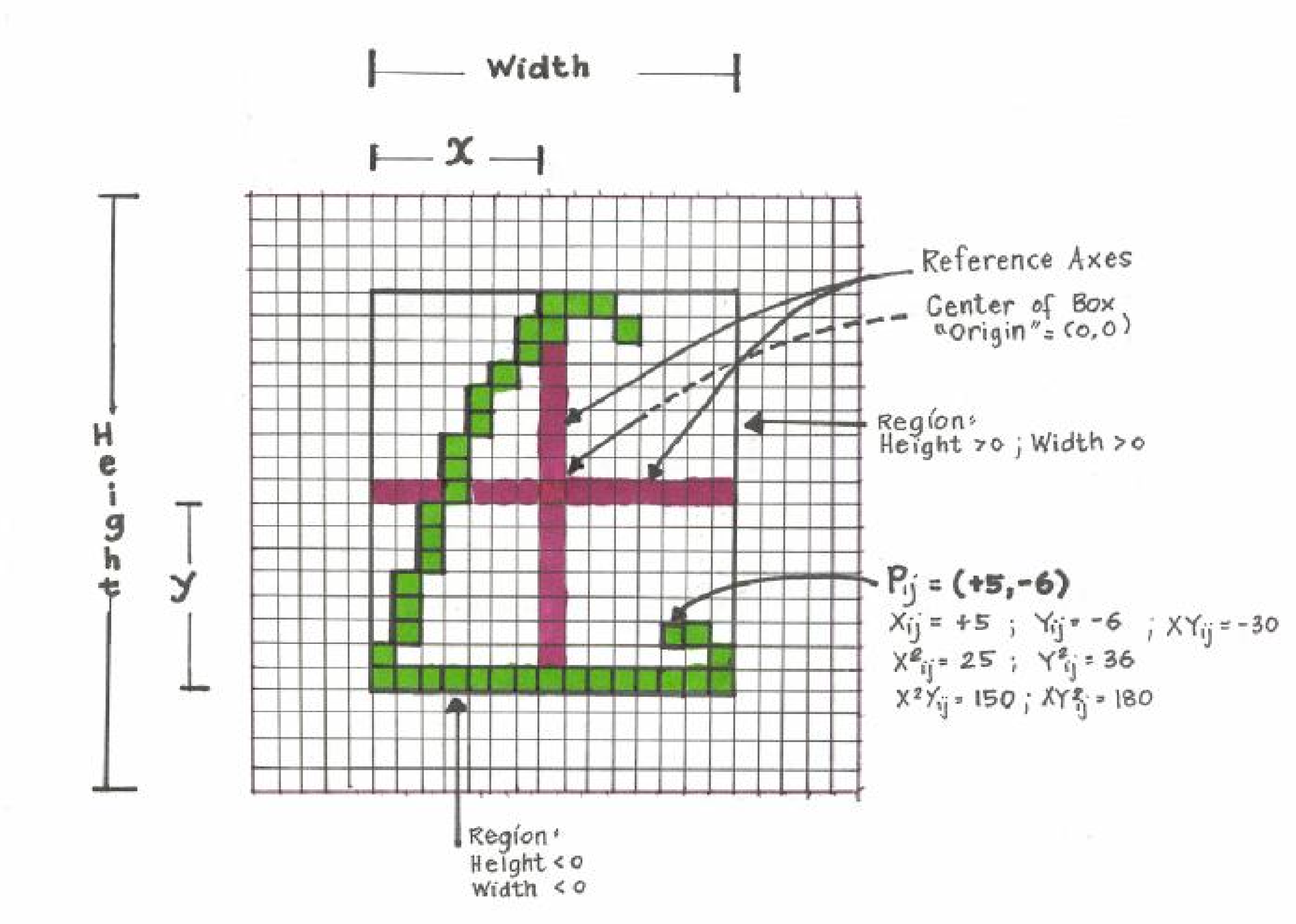
Sample Images



Attribute Information

- Each image associated with a vector of 16 numerical attributes
- numerical attributes scaled to fit into a range of integer values from 0-15
- attributes represent primitive statistical features of pixel distribution

Sample Image and Pixel Distribution



Frequency Distribution of Letters

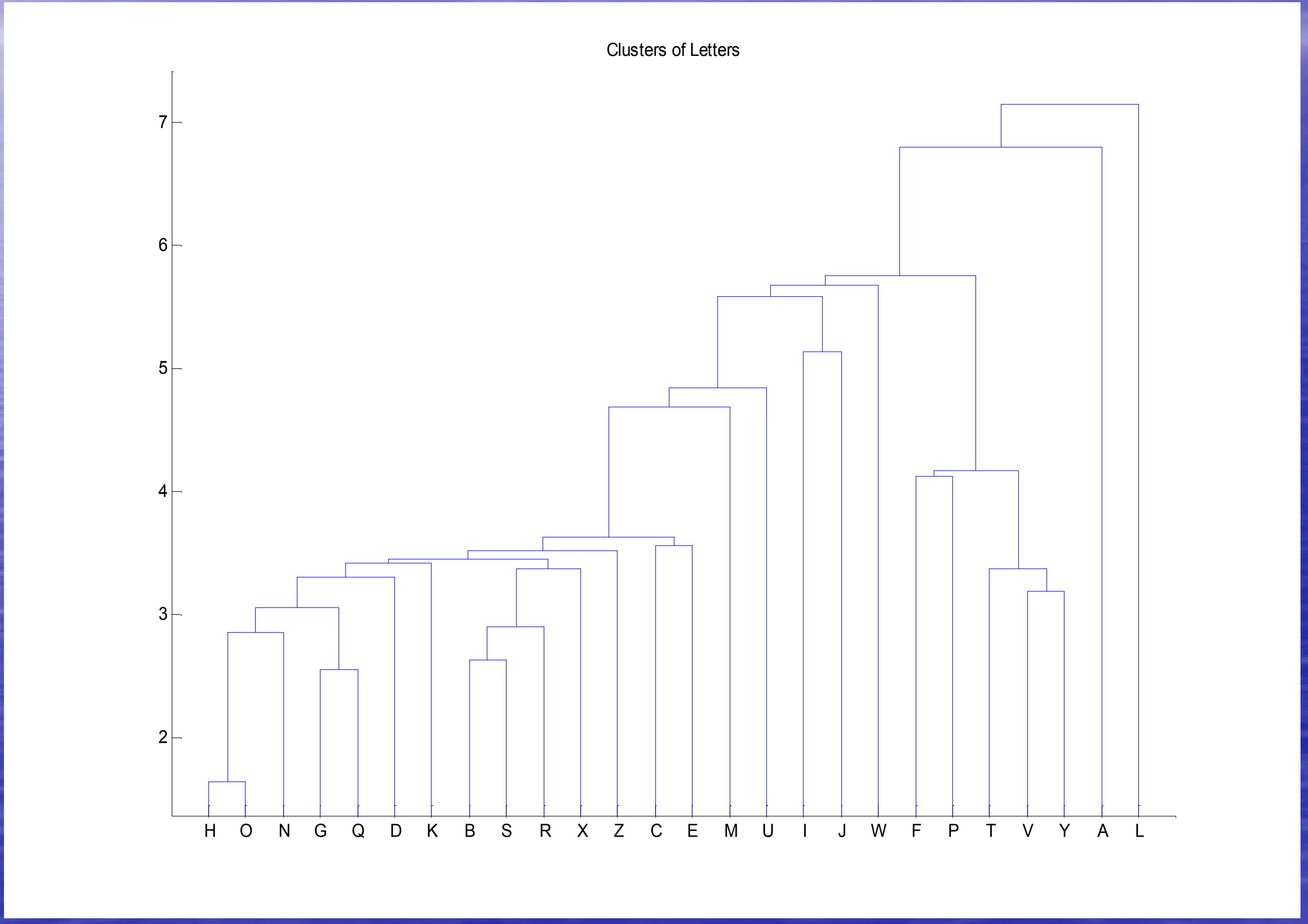
A – 789	F – 775	K – 739	P – 803	U – 813
B – 766	G – 773	L – 761	Q – 783	V - 764
C – 736	H – 734	M –792	R – 758	W - 752
D – 805	I – 755	N – 783	S – 748	X – 787
E – 768	J – 747	O – 753	T – 796	Y – 786
				Z - 734

There are no missing data

Previous Work

Method	Accuracy	Author, Year
Holland Style Classifier	82.7	Frey, Slate (1991)
First-NN	95.67	Aha et al (1991)
Alloc80 LVQ	93.6 92	Taylor (1994)
C4.5 + CART + ECOC	90	Dietterich and Bakiri (1995)
SVM	97.98	Hsu and Lin (2002)
1-NN+Adaboost	96	Athitsos (2004)

Exploratory Cluster



Performance Measures

- Sensitivity (*S1*)

  $S1 = P\{ X = \text{'A'} \mid Y = \text{'A'} \}$

  $= P\{X=\text{'A'} \cap Y=\text{'A'}\} / P\{Y=\text{'A'}\}$

- Specificity (*S2*)

  $S2 = P\{ Y = \text{'A'} \mid X = \text{'A'} \}$

  $= P\{X=\text{'A'} \cap Y=\text{'A'}\} / P\{X=\text{'A'}\}$

1 Nearest Neighbor

- Entails retaining all elements of the training set in memory and using them to classify each member of the testing set
- To determine the class of a member in the testing set, its Euclidean distance from each member in the memory is calculated.
- It is then assigned the same classification as the classification of the member it is nearest to

1 Nearest Neighbor

- Use the randperm function in matlab to create training and testing sets
- Traing -16000

  Testing-4000
- Code the 1-NN algorithm in matlab
- Run 12 experiments

1 Nearest Neighbor

- Average success rate: 95.81%
- Consistency (standard deviation): 0.13%
- Acceptable low error rates
- High sensitivity
- High specificity

1 Nearest Neighbor

Worst Error Rate Statistics

Letter	Mean	STD	Min	Max
H	10.43	3.15	5.67	16.54
K	8.73	2.94	3.97	14.46
B	7.47	2.53	4.83	13.84
R	7.03	2.60	3.57	11.25
E	6.45	1.67	4.29	11.03
F	6.12	1.93	2.84	9.35

1 Nearest Neighbor

Best Error Rate Statistics

Letter	Mean	STD	Min	Max
A	0.79	0.84	0.00	2.12
Z	1.47	0.98	0.00	3.10
Y	2.17	0.91	0.69	3.68
M	2.18	1.13	1.16	4.90
S	2.20	0.87	0.71	3.59
Q	2.84	0.87	1.27	4.09
V	3.26	1.26	0.63	4.97

1 Nearest Neighbor

- We also investigated what is the nature of misclassification,
- For example, the misclassification rate of "B" as "R" is 2.22%
- And the misclassification rate of "F" as "P" is 2.75% etc

1 Nearest Neighbor

- As attributes are quantized from 0~15, so we have situation of ties
- In the case of tie, we choose the first class

Percentage of Error Due to Ties

Expt #:	1	2	3	4	5	6	7	8	9	10	11	12
% of error	27.1	26.4	22.4	28.1	26.8	22.4	29.1	20.6	23.3	21.5	28.1	20.7

Linear Discriminant Analysis

- Let Pr(G|X) be the posterior of classification given attribute X
- $f_k(X)$=conditional density of X given G=k
- π_k =prior probability of class k
- By Bayes theorem

$$Pr(G=k \mid X=x) = f_k(x).\, \pi_k \;/\; \Sigma_l \; f_l(x)\, .\pi_l$$

Linear Discriminant Analysis

- LDA assumes

$$f_k(x) = (1 / (2\pi)^{p/2} |\boldsymbol{\Sigma}_k|^{1/2}) \exp\{-½ (\mathbf{x} - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1} (\mathbf{x} - \boldsymbol{\mu}_k)\}$$

- So the logarithm of the posterior probability is

$$\log\{\Pr(G=k \mid X=x)\} = -.5 * \log\{(1 / (2\pi)^p |\Sigma_k|)\} - ½ (x - \mu_k)^T \Sigma_k^{-1} (x - \mu_k) + \log(\pi_k)$$

- LDA classifies based on the logarithm of the posterior probability

Linear Discriminant Analysis

- We coded the LDA in R
- Average success rate: 69.88%
- Consistency (standard deviation): 0.21%

Worst Classified

Letters	*Average accuracy*
E	44.3
G	45.6
H	46.8
S	46.9
Y	50.1

Linear Discriminant Analysis

Best Classified

Letters	*Average accuracy*
A	85.8
M	88.6
V	85.4
W	85.9

Linear Discriminant Analysis

- Nature of misclassification
- The misclassification rate of “E” as “G” is 10.9%
- The misclassification rate of “G” as “C” is 18%
- The misclassification rate of “H” as “K” is 8.1%

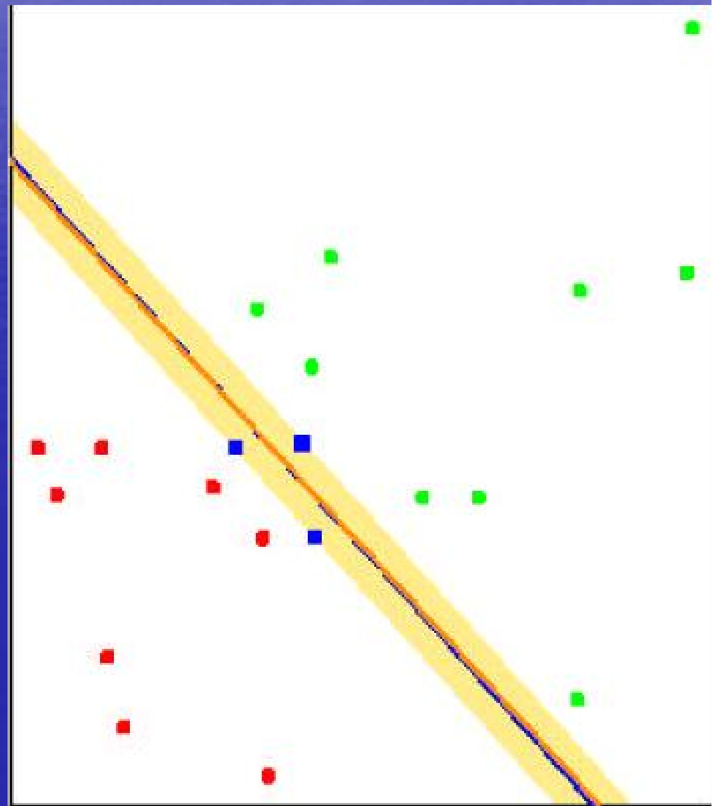
Support Vector Machines

- SVM is a classification method to maximize the margin between two classes {-1, 1}

$$\min_{\beta,\beta_0\xi_i} \frac{1}{2}\|\beta\|^2 + \gamma\sum_{i=1}^{N}\xi_i$$

$$s.t.$$

$$\xi_i \geq 0, y_i(\beta \cdot \phi(x_i) + \beta_0) \geq 1 - \xi_i \forall i$$



- Decision rule

$$\hat{y}_i = \mathrm{sgn}(\beta \cdot \phi(x_i) + \beta_0)$$

Support Vector Machines

- For Multi Classes ($k=1, 2 \ldots n$)
  - Solve one optimization problem
  - Combining several SVMs for binary classifications
    - One-against-all: *n* SVMs
    - One-against-one: $n(n-1)/2$ SVMs
    - DAG

Support Vector Machines

- LIBSVM: one-against-one
  - Success rate: 97.98%
- OSU SVM: matlab toolbox for LIBSVM
  - Does not work for our problem: 16%
- SVM toolbox by Dr. Schwaighofer
  - Error-Correcting Output Codes

Support Vector Machines

- ECOC

Class	Code Word														
	f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}
0	1	1	0	0	0	0	1	0	1	0	0	1	1	0	1
1	0	0	1	1	1	1	0	1	0	1	1	0	0	1	0
2	1	0	0	1	0	0	0	1	1	1	1	0	1	0	1
3	0	0	1	1	0	1	1	1	0	0	0	0	1	0	1
4	1	1	1	0	1	0	1	1	0	0	1	0	0	0	1
5	0	1	0	0	1	1	0	1	1	1	0	0	0	0	1
6	1	0	1	1	1	0	0	0	0	1	0	1	0	0	1
7	0	0	0	1	1	1	1	0	1	0	1	1	0	0	1
8	1	1	0	1	0	1	1	0	0	1	0	0	0	1	1
9	0	1	1	1	0	0	0	0	1	0	1	0	0	1	1

- Hamming distance: number of bits differ

Support Vector Machines

- Gaussian RBF kernel
- ECOC table with string length of 15
- Run 12 experiments
- Average success rate: 96.96%
- Consistency (standard deviation): 0.26%

Support Vector Machines

Comparison of SVM Methods

Method	Number of SVMs	Success rate (%)	Tuning of Parameters
one-against-all	26	97.88	Yes
one-agianst-one	325	97.98	Yes
DAG	325	97.98	Yes
ECOC	15	96.96	No

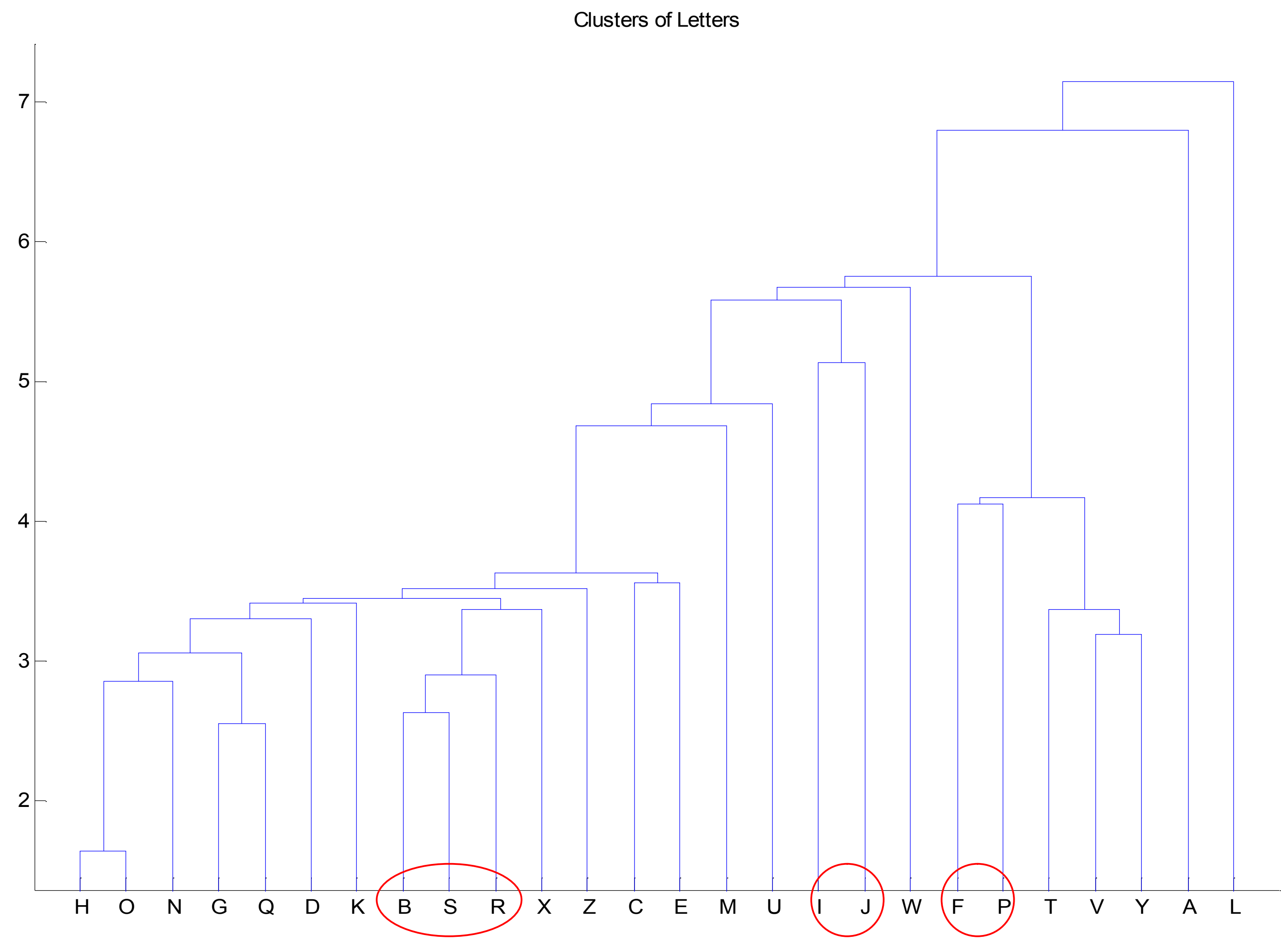
Comparison of ECOC Methods

Method	Success rate (%)	Length of codes
Tree-based (C4.5, CART)	Above 90	62
SVM	96.96	15

Support Vector Machines

- Best Classified: *A, S, U, Z, T*
- Worst Classified: *D, B, K, R, H*
- Some misclassification patterns in confusion matrix: *P & F; I & J; B & S & R*

Support Vector Machines



Findings

- 1. 1-NN and SVM gave higher sensitivity and specificity, compared with the LDA. The difference in both measures is at least 15%
- 2. 1-NN and SVM showed faster learning rates compared with LDA. We note however that LDA's accuracy (70%) did not significantly change when the training set was reduced from 16,000 to 1,600.

Findings

- 3. In terms of classification errors, 1-NN and SVM algorithms produce similar misclassification patterns in their respective confusion matrices.
- 4. The most computing intensive method is SVM, and least is LDA.
- 5. The OSU-SVM Toolbox (Ahalt, Ma,&, Zhao,2002) may need code modification as it did not work for this particular data set.

Limitation

-methods treated all sixteen attributes equally.

-relationships between the features to determine plausibility of dimension reduction was not fully explored.

(e.g., elimination, linear or non-linear combinations of some features)

Next Steps

- a. Consider different types of boosting to improve the performance of 1-Nearest Neighbor.
- b. Distance metrics, other than Euclidean, may be explored for the nearest-neighbor algorithm.

Next Steps:

- c. Optimize the parameters of the SVM ECOC.
- Also try using different ECOC tables.

- d. In view of recent developments, explore hybrid methods which combine the advantages of statistical and non-statistical algorithms. For example, doing a tree-based method and a multiple logistic regression.

End of Presentation

Acknowledgement

- Prof. Prem Goel
- Prof. Joseph Verducci
- Prof. Yoonkyung Lee
- Prof. Stanley Ahalt
- Dr. Junshui Ma