K-Nearest Neighbor Based Spoken Letter Recognition

1st Annika Shankwitz
Department of Linguistics
Indiana University
Bloomington, Indiana
ashankwi@iu.edu

***Abstract*—Spoken letter recognition is the task of hearing a spoken letter and identifying the corresponding written letter, e.g. hearing the letter "B" and providing the written letter B. This project examines whether a K-Nearest Neighbor (KNN) model can complete the task of spoken letter recognition. In order to do so, a KNN model was trained on a multi-class dataset of isolated spoken letters. As said dataset is high-dimensional, Principal Component Analysis was used to reduce data dimensionality from 617 to 179, and a ball tree was used to improve the KNN model's efficiency. Ultimately, the resulting KNN performed well, achieving a macro-precision, macro-recall, and macro-F1 score of 0.93. Letters with distinct pronunciations were easiest for the model to classify – H, I, O, W, R, X, Y, while letters with pronunciations similar to other letters of the alphabet were hardest for the model to classify – B, M, N, P, V.**

***Index Terms*—Spoken Letter Recognition, K-Nearest Neighbors, Principal Component Analysis, Ball Trees, Multi-class Classification**

I. Introduction

Automatic speech recognition is a difficult task. Not only must automatic speech recognition systems contend with the acoustical complexity of human speech, they must also deal with ambiguity ("tale" versus "tail"), out-of-vocabulary words, and variability across speakers [5]. As the demand for speech technologies increases, this task becomes more and more important. The current project aims to address a simplified version of automatic speech recognition, namely spoken letter recognition.

Spoken letter recognition is a relatively straightforward task – it is the task of hearing a spoken letter of an alphabet, i.e. the sound "B" ([bi][1]), and identifying the corresponding written letter, i.e. the letter B. While this task is easy for humans, it is interesting to consider if machine learning classifiers can successfully complete this task.

A team of researchers did just that by creating the Isolated Letter Speech Recognition database (ISOLET) [1], [2], and subsequently using ISOLET to train and test a neural network. The resulting model achieved 95.5% accuracy [3]. Similar results have also been achieved on ISOLET using a Hidden Markov Model, explicitly an accuracy of 96.6% [6].

To the best of my knowledge, a K-Nearest Neighbor (KNN) based approach has only been applied to ISOLET once [4]. This is possibly due to the large dimensionally of ISOLET (7,800 instances x 617 features). In fact, reference [4] was not focused on ISOLET itself, but rather proposed the IOC (International Olympic Committee) algorithm[2], a novel KNN based algorithm designed for high-dimensional, multi-class classification tasks. Given that such an approach is outside the scope of the current project, this project aims to investigate the performance of a standard KNN model – with techniques to reduce data dimensionality (Principal Component Analysis) and increase model efficiency (KD tree or ball tree) on ISOLET. In doing so, this project seeks to answer the following questions:

Q1. Does a KNN model perform well on ISOLET?
Q2. Which spoken letters are easiest for the KNN model to classify?
Q3. Which spoken letters are hardest for the KNN mode to classify?

II. Methods

This section outlines the methodology employed by this project. The dataset, ISOLET, is discussed in Section II-A, data preprocessing is covered in Section II-B, and creation of the KNN model is outlined in Section II-C.

A. Data

The dataset used by this project is ISOLET[3] – the Isolated Letter Speech Recognition database [1], [2]. "ISOLET is a database of letters of the English alphabet spoken in isolation" [1]. ISOLET was created by having 150 speakers each produce all 26 letters of the English alphabet twice. As a result, ISOLET contains 7,800 spoken letters total. Each spoken letter in ISOLET is labeled with an integer between 1 and 26, with 1 corresponding to the letter A and 26 corresponding to the letter Z.

Every spoken letter has 617 features. Broadly, these features were computed from the audio signal. More specifically, the features can be broken down into four categories: contour features (features reflecting the phonetic category of the letter), sonorant features (features of the vowel accompanying the

[1]Phonetic transcription of the sound "B" using the International Phonetic Alphabet.

[2]The IOC algorithm mimics how cities are chosen to host the summer Olympics.

[3]https://archive.ics.uci.edu/dataset/54/isolet

letter[4]), pre-sonorant features (features of the letter preceding the vowel[5]), and post-sonorant features (features of the letter following the vowel[6]) [3]. All features are continuous.

The dataset is broken into five parts: ISOLET1-5. Each part contains 30 speakers' productions of the English alphabet, with 15 speakers being female and 15 speakers being male. In order to prevent individual speakers from appearing in both training and testing datasets, the authors suggest using ISOLET1-4 for training and ISOLET5 for testing. This results in an 80%/20% training and testing split.

B. Data Preprocessing

This project employs two data preprocessing techniques. First, to assure that all features have equal importance in predicting class labels, data standardization was performed. The standardization technique used in this project was mix-max scaling.

The next technique employed by this project was Principal Component Analysis (PCA). ISOLET has 617 features, making it high-dimensional. As such, PCA was used to reduce the dimensionality of the data (while capturing 95% of ISOLET's variance).

C. K-Nearest Neighbor Model

This project implements a KNN model, specifically scikit-learn's KNeighborsClassifier() [7]. The model was trained on ISOLET1-4 (80% of ISOLET). Additionally, five-fold cross validation was used and gridSearchCV() was performed. Grid search targeted the number of neighbors (k=1,3,5,7,9,11,13,15,17,19,21), the weights (uniform versus distance), and the algorithm (ball tree versus KD tree). Finally, the model was tested on ISOLET5 (20% of ISOLET). Given the structure of ISOLET, namely that 150 speakers each produced every letter of the English alphabet twice, this is a balanced classification task.

III. Results

This section details the project's results. The results of PCA can be seen in Section III-A, the final KNN model is described in Section III-B, and the performance of said model is summarized in Section III-C.

A. PCA

PCA revealed that 179 principal components capture 95% of ISOLET's variance. ISOLET's first two principal components can be seen graphed against each other in Figure 1. Some classes form distinct clusters, e.g. H, R, and O; however, the majority of classes do not form distinct clusters and instead overlap with each other.

[4]Spoken letters of the English alphabet are accompanied by vowels; the letter B is pronounced like the word "bee".

[5]Some letters of the English alphabet are produced preceding a vowel: B, C, D, etc.

[6]Some letters of the English alphabet are produced following a vowel: F, S, X, etc.

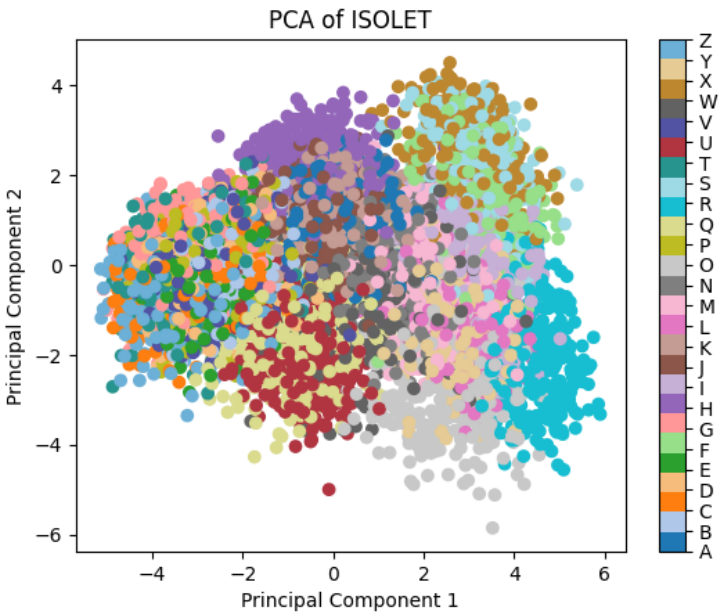

Fig. 1. PCA of ISOLET

B. The KNN Model

The best parameters found by grid search were as follows:

- number of neighbors = 9
- weights = distance
- algorithm = ball tree

In prose, these parameters indicate that the final KNN model considers the 9-nearest neighbors during classification, and that the votes of said 9 neighbors are weighted by their distance from new instances being classified. Additionally, the model organizes its data with a ball tree.

C. Model Performance

Turning now to performance, the KNN model's macro-precision, macro-recall, macro-F1 score, and accuracy can be seen in Table I, and the precision, recall, and F1 scores of individual classes can be seen in Table II. Additionally, the KNN model's confusion matrix can be seen in Figure 2.

To summarize, the KNN model achieved a macro-precision, macro-recall, macro-F1 score, and accuracy of 0.93. The letters easiest for the model to classify, based on F1 score, were H (1.0), I (0.99), O (0.99), R (0.99), W (0.99), X (0.99), and Y (1.0). The letters hardest for the model to classify, based on F1 score, were B (0.79), M (0.84), N (0.84), P (0.84), and V (0.85).

TABLE I
KNN Model Metrics

Macro-Precision	Macro-Recall	Macro-F1 score	Accuracy
0.929524	0.926836	0.926585	0.926876

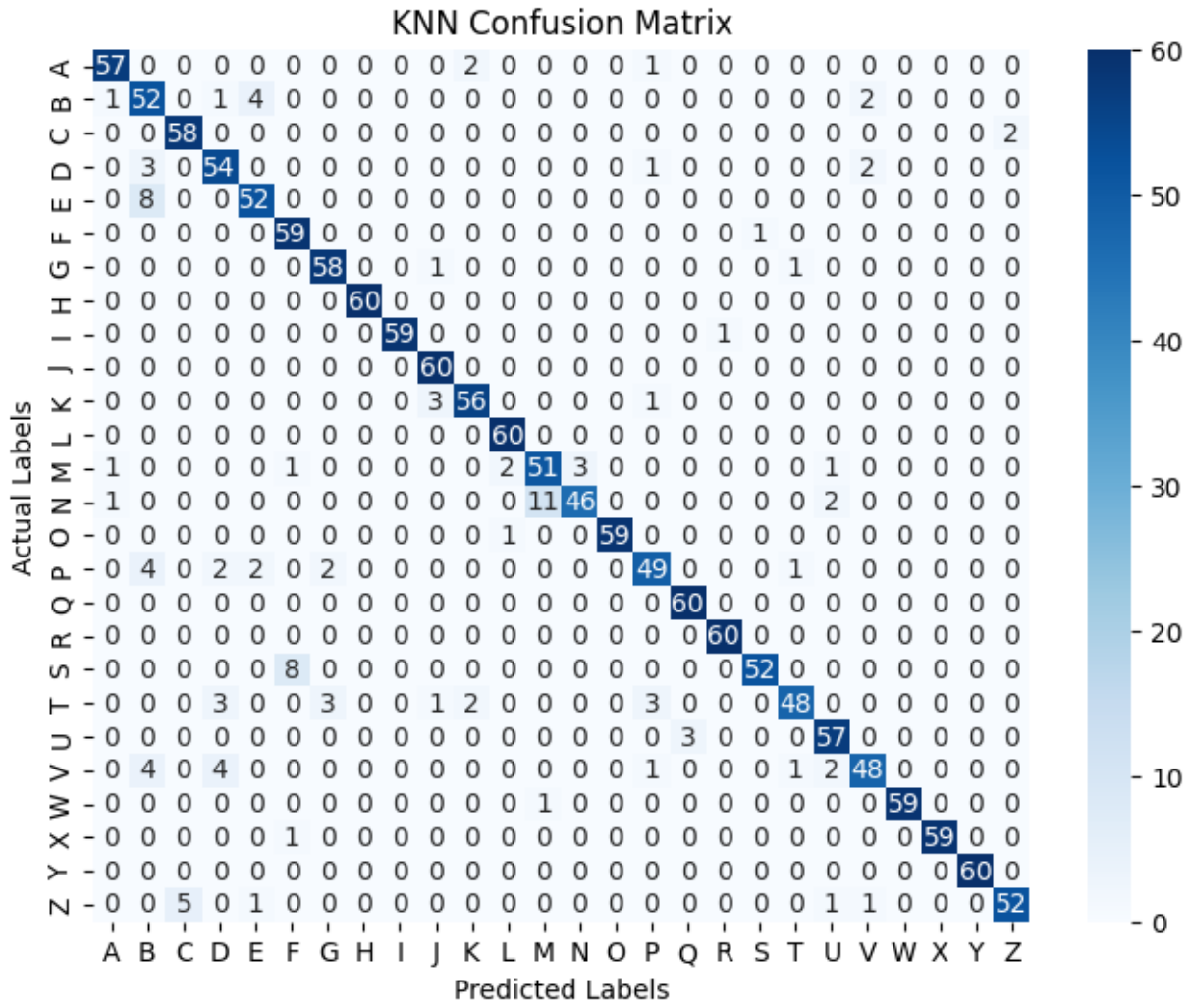

Fig. 2. KNN Model's Confusion Matrix

IV. Discussion

This section discusses the project's research questions. Q1 is considered in Section IV-A, Q2 in Section IV-B, and Q3 in Section IV-C.

A. Q1: Does a KNN model perform well on ISOLET?

The above results indicate that yes, a KNN model does perform well on ISOLET. First, the model's macro-precision, macro-recall, macro-F1 score, and accuracy are high in their own right. Second, an accuracy of 93% makes the current KNN model only 2.5% less accurate than a neural network [3] and only 3.6% less accurate than a Hidden Markov Model [6], both of which are more complicated models.

Figure 2 also provides a way to gauge the KNN model's performance. The deep blue diagonal present in the confusion matrix indicates that the majority of the KNN model's predictions were correct.

In sum, a KNN model does perform well on ISOLET. This conclusion is supported by both the current model's metrics and its confusion matrix.

B. Q2: Which spoken letters are easiest for the KNN model to classify?

Based on F1 scores, the letters easiest for the model to classify were H, I, O, R, W, X, and Y. Of the letters with an F1 score of 0.99, I, O, W, and X all had a precision of 1, meaning that these labels were not wrongly applied to other classes. Additionally, it is interesting to note that three of these letters, H, O, and R, formed distinct clusters in Figure 1

A possible explanation for the high performance of these letters can be found in their pronunciations, which are provided in Table III[7]. Some of these letters contain unique, two sound sequences: H is the only letter to contain the sequence [tʃ], O is the only letter to contain the sequence [ow], and X is the only letter to contain the sequence [ks]. Some of these

[7]All letters were transcribed by the author.

TABLE II
CLASS METRICS

Class	Precision	Recall	F1 score
A	0.950000	0.950000	0.950000
B	0.732394	0.866667	0.793893
C	0.920635	0.966667	0.943089
D	0.843750	0.900000	0.870968
E	0.881356	0.866667	0.873950
F	0.855072	0.983333	0.914729
G	0.920635	0.966667	0.943089
H	1.000000	1.000000	1.000000
I	1.000000	0.983333	0.991597
J	0.923077	1.000000	0.960000
K	0.933333	0.933333	0.933333
L	0.952381	1.000000	0.975610
M	0.809524	0.864407	0.836066
N	0.938776	0.766667	0.844037
O	1.000000	0.983333	0.991597
P	0.875000	0.816667	0.844828
Q	0.952381	1.000000	0.975610
R	0.983607	1.000000	0.991736
S	0.981132	0.866667	0.920354
T	0.941176	0.800000	0.864865
U	0.904762	0.950000	0.926829
V	0.905660	0.800000	0.849558
W	1.000000	0.983333	0.991597
X	1.000000	0.983333	0.991597
Y	1.000000	1.000000	1.000000
Z	0.962963	0.866667	0.912281

letters have a unique structure: R is the only rhotic[8] letter of the English alphabet, W is the only letter with a multi-syllable pronunciation, and Y is the only letter to feature a glide,[9] [w], followed by a diphthong,[10] [aɪ]. Finally, while I doesn't have a unique structure, its structure appears only in the letter Y [waɪ], which is itself a distinct letter.

So, it seems that letters which have distinct pronunciations were easiest for the model to classify.

TABLE III
IPA TRANSCRIPTIONS OF THE ENGLISH ALPHABET

Letter	IPA Transcription	Letter	IPA Transcription
A	[eɪ]	N	[ɛn]
B	[bi]	O	[ow]
C	[si]	P	[pi]
D	[di]	Q	[kju]
E	[i]	R	[aɹ]
F	[ɛf]	S	[ɛs]
G	[dʒi]	T	[ti]
H	[eɪtʃ]	U	[ju]
I	[aɪ]	V	[vi]
J	[dʒeɪ]	W	[də.bl̩.ju]
K	[keɪ]	X	[ɛks]
L	[ɛl]	Y	[waɪ]
M	[ɛm]	Z	[zi]

C. Q3: Which spoken letters are hardest for the KNN model to classify?

Based on F1 scores, the letters hardest for the model to classify were B, M, N, P, and V. Once again, their pronunciations appear in Table III. Like in Section IV-B, a possible explanation for the lower performance of these letters can also be found in their pronunciations.

These letters form two categories based on acoustic similarity, the first of which consists of M and N. The pronunciations of M and N are very similar: [ɛm] and [ɛn]. Not only do they share the vowel [ɛ], but the sounds [m] and [n] are themselves very similar to one another acoustically and articulatorily. This explanation is supported by the confusion matrix, which shows that M and N are often mistaken for one another.

The second group consists of B, P, and V. The pronunciations of these letters are very similar to each other: [bi], [pi], and [vi], but also to the pronunciations of D [di], E [i], and T [ti]. Once again, the confusion matrix reflects these similarities. B was wrongly classified as A (n=1), D (n=1), E (n=4), and V (n=2), and was wrongly predicted as the label for D (n=3), E (n=8), P (n=4), and V (n=4). P was wrongly classified as B (n=4), D (n=2), E (n=2), G (n=2), and T (n=1), and was wrongly predicted as the label for A (n=1), D (n=1), K (n=1), T (n=3), and V (n=1). Finally, V was wrongly classified as B (n=4), D (n=4), P (n=1), T (n=1), and U (n=2), and was wrongly predicted as the label for B (n=2), D (n=2), and Z (n=1). Additionally, the next hardest letters for the model to predict were in fact T (F1 score=0.86), D (F1 score=0.87), and E (F1 score=0.87).

These patterns suggest that letters with pronunciations similar to other letters of the English alphabet were the most difficult for the model to classify.

V. CONCLUSION

This project used a KNN model for the task of spoken letter recognition, namely to classify spoken letters from the Isolated Letter Speech Recognition database. To do so, ISOLET was standardized with min-max scaling, and the dimensionality of ISOLET was reduced from 617 to 179 with PCA. Performing grid search and five-fold cross validation resulted in a KNN model which organized its data with a ball tree, and considered the distance-weighted votes of the 9 nearest neighbors during classification. Ultimately, said model performed well, achieving a macro-precision, macro-recall, macro-F1 score, and accuracy of 0.93 (Q1).

Based on F1 scores, the letters easiest for the model to classify were H, I, O, R, W, X, and Y (Q2) and the letters hardest for the model to classify were B, M, N, P, and V (Q3). A possible explanation for these groups can be found in their pronunciations. Specifically, letters with pronunciations distinct from other letters of the alphabet were easy to classify, while letters with pronunciations similar to other letters of the alphabet were difficult to classify.

A limitation of the present project is that a baseline KNN model wasn't created. As such, it is unclear if the techniques used to reduce data dimensionality and increase model efficiency were necessary, or if a baseline model, i.e. a brute-force KNN model trained and evaluated on all 617 of ISOLET's features would show equal performance. Thus, future work should make this comparison.

[8] An R-like sound.
[9] A sound like the W in "water" or the Y in "yes".
[10] A sound consisting of two vowels.

ACKNOWLEDGMENT

All work was completed solely by the author.

REFERENCES

[1] R. Cole, Y. Muthusamy, and M. Fanty. “The ISOLET spoken letter database.” 1996.

[2] M. Fanty and R. Cole. “ISOLET”. UCI Machine Learning Repository. Available: https://archive.ics.uci.edu/dataset/54/isolet.

[3] M. Fanty and R. Cole. “Spoken letter recognition”. Advances in Neural Information Processing Systems (NIPS Conference). 1990. pp. 220-226.

[4] T. Liu, K. Yang, and A. Moore. “The IOC algorithm: efficient many-class non-parametric classification for high-dimensional data”. Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2004.

[5] F. Markus. “Why is speech recognition difficult?”. Semantic Scholar. 2003.

[6] H. Nock and S. Young. “Loosely coupled HMMs for ASR”. Sixth International Conference on Spoken Language Processing (INTERSPEECH). 2000.

[7] Pedregosa et al., “Scikit-Learn: machine learning in Python”, Journal of Machine Learning Research, vol. 12. 2011. pp. 2825-2830.