

Japanese Handwritten Character Recognition

Hans Livingstone, Yuta Fujiwara, Nei Kato

Fig. 1 - Euclidian distance between the averages of samples あ and お

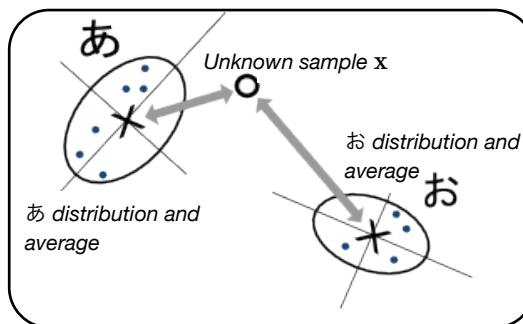


Fig. 2 - kNN with k = 3. In this case あ will be chosen instead of お. い will not be considered.

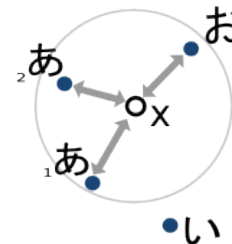


Fig. 3 - kNN Percent Error for Varying k

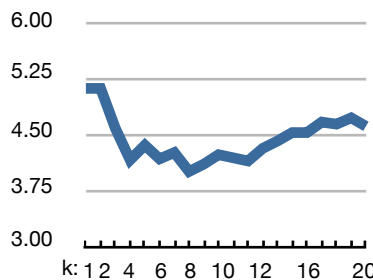


Fig. 4 - Experiment Percent Error

Euclidean	4.817
Manhattan	7.141
Deg. of Sim.	6.338
kNN (k=8)	4.012

Introduction

Demand for automated handwritten text recognition services is growing. For these systems to be valid in Japan, a robust handwritten Kanji recognition system must be implemented. My research focuses on pattern recognition and machine learning methods applicable to this field.

This paper outlines several key algorithms and useful heuristics, along with the experiment results.

Preprocessing

Pattern recognition algorithms expect input in a precise format, therefore all image data must first be preprocessed to remove noise, cropped, normalized, and have the important features extracted. In the case of handwritten Japanese characters, this results in a 196 component feature vector that can uniquely identify each character.

Machine Learning

Effective pattern recognition requires a healthy amount of sample data to learn from. This experiment uses the EL9B handwritten Kanji database as the source for training data. For simplicity Hiragana was used, resulting in 71 characters, each with 100 different learning samples.

After the training data is preprocessed, a dictionary is created. This dictionary is used to classify new characters based on the definitions from the training data.

Classification by Distance from Class Average

In the simplest case, the average vector for each type is found and saved in the dictionary along with its type. In this experiment, dictionary **D** contains 71 average vectors, one for each Hiragana.

Generally, the classification process is about discovering the distance between the input character **x** and the vectors $\mathbf{p} = \mathbf{D}_m$ in the dictionary. When the closest vector **p** is found, **x** is classified as the type of **p**.

Several different definitions of distance exist. To evaluate these distance methods, 7100 additional samples were tested against the database **D** and their error rates calculated. The results are in Fig. 4.

Euclidian Distance

The typical definition of geometric distance comes from the Pythagorean theorem. Euclidian distance is the natural extension of the Pythagorean theorem into higher dimensions and is the most commonly used distance metric in pattern recognition. The distance between an input vector **x** and a training sample **p** (from the dictionary **D**) can easily be calculated by the following (Fig. 1),

$$f(x, p) = \sqrt{\sum_{i=1}^N (x_i - p_i)^2}$$

Through experimentation, it was found that Euclidian distance is the most accurate distance metric for this data-set with an error rate of 4.817%.

Manhattan Distance

Although, Euclidean distance is regarded as the canonical distance measurement, with some data distributions other metrics prove useful. Manhattan distance is one such metric. Its name comes from the fact that inside a city the shortest

distance between two points is not a straight line, but rather a zig-zag between the city blocks.

$$f(x, p) = \sum_{i=1}^N |x_i - p_i|$$

In this case it was found that the Manhattan distance was the least accurate, with 7.141% error.

Degree of Similarity

Angular distance, also known as the Degree of Similarity, results from the dot product between vectors **x** and **p**

$$f(x, p) = \frac{x \cdot p}{|x||p|} = \frac{\sum_{i=1}^N x_i p_i}{\sqrt{\sum_{i=1}^N x_i^2} \sqrt{\sum_{i=1}^N p_i^2}} = \cos \theta$$

Because two very different vectors can still share a similar angle, except for special cases, Euclidian distance is most commonly used. Again, with a 6.338% error rate, the experiment results verify this.

k-Nearest Neighbor (kNN)

Instead of using a dictionary containing only the class average vectors, the kNN algorithm (Fig. 2) uses a dictionary complete with all the training data. In this experiment the dictionary β contained 7100 items (71 Hiragana with 100 samples each.)

1. Select the closest k characters from β by Euclidian distance.
2. From the k selected characters, find the most often selected type. If there is a tie, select the type with the smallest total distance.
3. Classify **x** as this type of character.

Other distance methods were tried with kNN but yielded poor results. A k value of 8 was found to give the lowest error rate for this distribution of training data (Fig. 3). Because kNN does not rely on fuzzy averages, but instead uses the entire set of training data, it should be the most accurate method. With a 4.012% error rate, the experiment results verify this.