A Survey on Sentiment Classification in Face Recognition

Jingyu Qian

Engineering School, University of Michigan, Michigan MI 48105, US
*egbert_qian@163.com

Abstract. Face recognition has been an important topic for both industry and academia for a long time. K-means clustering, autoencoder, and convolutional neural network, each representing a design idea for face recognition method, are three popular algorithms to deal with face recognition problems. It is worthwhile to summarize and compare these three different algorithms. This paper will focus on one specific face recognition problem—sentiment classification from images. Three different algorithms for sentiment classification problems will be summarized, including k-means clustering, autoencoder, and convolutional neural network. An experiment with the application of these algorithms on a specific dataset of human faces will be conducted to illustrate how these algorithms are applied and their accuracy. Finally, the three algorithms are compared based on the accuracy result.

1. Introduction

Face recognition has become an important problem for a long time. Commercial and law enforcement applications make it critical for both business industry and public security. In addition, researchers from different fields including image processing and computer visions are continuously applying different algorithms for face recognition problems [1].

Till now, typical methods for face recognition include holistic methods, feature-based methods, and hybrid methods [1]. Holistic methods include algorithms such as principal component analysis (PCA) and its derivation two-dimensional principal component analysis (2DPCA) [1], [2]. Feature-based methods include algorithms such as hidden Markov model (HMM) [1], [3]. Hybrid methods include algorithms such as modular eigenfaces [1], [4]. In addition, k-means clustering and its derivation, due to their computational efficiency, are also used in face recognition. Autoencoders, including deep autoencoder [5], are widely used in complicated face recognition problems, too. Despite its long training time, convolutional neural network has also become a widely used algorithm to do image processing work including face recognition problems due to its consideration of spatial characteristics.

In this paper, we will focus on three different algorithms to deal with a specific problem of face recognition - sentiment classification from still images of facial expressions. The three algorithms include k-means clustering, autoencoder, and convolutional neural network. We will first summarize these three algorithms in Section 2 by showing details and mathematical backgrounds of them. Then, we will generally compare these three algorithms in Section 3. Next, we will practically apply these three algorithms on a specific labelled dataset of still images of facial expressions to train models and compare their accuracy in Section 4. Section 5 will conclude the paper.
2. Three ways of sentiment classification from images of facial expressions

2.1 K-means clustering

K-means clustering [6] is an unsupervised learning algorithm in machine learning, meaning that it can be used to conduct classification without labels. The goal of this algorithm is to classify all the samples into different clusters so that in the same cluster, the sum of squares is minimized [7]. Suppose there are n samples defined by feature vectors \((x_1, x_2, \ldots, x_n)\) to be classified into k clusters \((C_1, C_2, \ldots, C_k)\). In this case, we need to find k centroids \((c_1, c_2, \ldots, c_k)\) with respect to k clusters, and the assignment of each sample to one cluster such that it minimizes:

\[
\sum_{j=1}^{k} \sum_{x_i \in C_j} \|x_i - c_j\|_2^2
\]  

(1)

Since the centroids and assignment are unknown initially, traditional k-means clustering initializes randomly k centroids [8]. With the k centroids, find the assignment for each feature vector \(x_i\) to one centroid \(c_j\) that minimizes \(\|x_i - c_j\|_2\). After that, recalculate each centroid with the means of the samples assigned to it. Then find the assignment again and iterate until the centroid does not change after recalculation. The iteration is guaranteed to end.

There are several extensions of k-means. K-means++ is an algorithm of carefully choosing initial centroids [9]. It first chooses one sample randomly from n samples as a centroid. Then, with previously chosen centroids, assign a probability to each sample that has not been chosen such that the higher the shortest distance of that sample to the previously chosen centroids is, the higher the probability is. Then, choose next centroid according to the probability. The initialization ends when all the k centroids are chosen. Other extensions include penalized and weighted k-means, which uses the weighted distances and penalty term in the loss function [10].

2.2 Autoencoder

Autoencoder is an unsupervised learning algorithm using neural networks [11]. It is used to learn the structure of the input feature vector instead of predicting the output label. An autoencoder consists of two parts: encoder and decoder. Suppose F is the input space, and \(F \in \mathbb{R}^k\) is a subspace of the k-dimensional real space. The encoder and decoder is defined as follows.

encoder: \(F \rightarrow \mathbb{R}^d\) \hspace{1cm} (2)

decoder: \(\mathbb{R}^d \rightarrow F\) \hspace{1cm} (3)

As we do not know the labels for unsupervised learning, we learn an function defined by decoder \(\circ\) encoder, whose domain and image are both in F. This function first encodes the input and then reconstructs the input feature vector by decoding the compressed-dimension feature vector. Typically the encoder decreases the dimensionality of the original input feature vector, i.e. \(k < d\) [11]. One way of decreasing the dimensionality is to use pooling layers in the encoder. It is intended to throw away irrelevant dimensions of data and help reduce the likelihood of overfitting. For each input feature vector \(f \in \mathbb{F}\), the loss function is given by the minimum square error:

\[
\|f - \text{decoder} \circ \text{encoder}(f)\|_2.
\]

(4)

Both the encoder and decoder can contain single or multiple layers of neural network. To learn the function \(\text{decoder} \circ \text{encoder}\), the weights related to the neural network for the encoder and decoder are learned.

Extensions of autoencoder includes denoising autoencoder, sparse autoencoder, and so on. Denoising autoencoder is intended to recover useful information from corrupted input [12]. Sparse autoencoder constrains the number of active hidden layers so that useful information can be extracted when the number of hidden layers is large [13].

2.3 Convolutional neural network (CNN)

Convolutional neural network (CNN) is a supervised learning algorithm [14]. It consists of four parts: convolutional, pooling, activation, and fully connected layers. The convolutional layer uses different filters to produce several feature maps for the input. The feature maps have smaller dimension than the
original input. The pooling layer is intended to downsample previous feature maps and avoids overfitting [14]. For example, the max pooling will apply the max filter to the sub-region of the previous feature maps, while the average pooling will apply the average filter. In this way, the dimensionality is reduced further. Between each successive two layers, the activation layer can apply activation functions to do more complicated calculation than linear combination. Activation functions include rectifier, sigmoid, and so on. The fully connected layer is typically applied at last to produce a combination of features of the previous layer for prediction. Linear combination and activation function can be used in this layer. Figure 1 shows a simple example of a convolutional neural network. In this example, two filters are applied to the original input, creating two feature maps. After the pooling layer, the dimensionality of the two feature maps further decreases. Finally, the fully connected layer will combine the two feature maps.

![Figure 1. An Example of Convolutional Neural Network](image)

During the training process, the weights in the convolutional and fully-connected layers are learned. Different design of hidden layers can be applied to deal with different machine learning problems.

3. Advantages and disadvantages of K-means clustering, autoencoder, and convolutional neural network

As two unsupervised leaning algorithms, k-means clustering and autoencoder does not require the label for the input. Thus, no human efforts are required to manually generate the labels for a large number of input. Besides that, k-means clustering is computationally fast [15]. However, k-means clustering often produces clusters of relatively uniform size. Thus, if the input has different true cluster size, k-means clustering can fail to learn that. Moreover, the result of k-means clustering depends on initialization a lot [15]. Poor initialization can produce poor clustering result. Autoencoder can be used to compute good representation of the input data. The combination of encoder and decoder can reduce irrelevant part of the input data. However, if the input has only limited essential information, the autoencoder will fail to throw away most of the irrelevant aspects of the data. Moreover, it is computationally expensive to train an autoencoder. Convolutional neural network takes advantage of locality of input data, and works well to learning problem that exhibits locality including sentiment detection from image. However, it requires large training dataset, and the training time is long. Table 1 gives a general comparison between the three algorithms.
Table 1. Comparison of K-Means, Autoencoder, and CNN

| Method       | Label Free | Computational Efficiency | Accurate | Model-Based |
|--------------|------------|--------------------------|----------|-------------|
| K-means      | √          | √                        |          |             |
| Autoencoder  | √          |                          | √        |             |
| CNN          |            |                          |          | √           |

4. Experiment

4.1 Experiment settings

Our dataset contains images of facial expressions with labels. Using parts of them as training data, we need to find a classification of the images of facial expressions into seven sentiments: Anger, Disgust, Fear, Happiness, Sadness, Surprise, and Neutral. We denote these 7 sentiments sequentially as image class 0 to 6. Three algorithms of classifications, k-means clustering, autoencoder, and convolutional neural network, are evaluated.

We first preprocess the data by splitting data samples into training, validation, and test sets. Next, we need to remove the blank image. Our definition of a blank image with feature matrix $X$ is given by Equation (5).

$$\|X - \text{mean}(X)\|_2 < 1$$  \hspace{1cm} (5)

Here $\text{mean}(\cdot)$ takes a matrix $X$ and returns a matrix of the same shape whose entries are the same and equal to the mean of the sum of entries of $X$. Finally, we normalize the feature matrix $X$ by making its mean of sum of entries zero and variance one by shifting and scaling values of each entry of $X$.

We use three different algorithms (k-means clustering, autoencoder, and convolutional neural network) to train the models and test the accuracy of each model by using cluster purity. Figure 2 shows a general procedure of the experiment.

![Figure 2. Experiment Setup](image)

4.2 Experiment description

4.2.1 K-means clustering. Instead of randomly initializing the cluster centroids, we use k-means++ initialization. The details of k-means++ is given below [9]:

1. Randomly select one point as the first centroid.
2. For each point $x$, calculate the distance to the nearest centroid that has been selected. The distance is denoted as $D_x$.
3. Select the next centroid with the probability of selecting point $x$ proportional to $D_x^2$.
4. Repeat step 2 and step 3 until $k$ centroids are chosen.
We test all the possible $k$ from $[1, 10]$ for random initialization and k-means++ initialization. The clustering performance is evaluated by using cluster purity. After the performance comparison, we will choose the best $k$ as the cluster number, and compare the performance of random initialization and k-means++ initialization.

4.2.2 Autoencoder. The autoencoder neural network consists of 6 layers. Layer 1 is an average pooling layer to reduce dimensionality. Layer 2 is a fully-connected layer to further reduce the image dimensionality to one dimension. Rectifier function $f(x) = \max(x, 0)$ is used as the activation function. Layer 1 and 2 work as the encoder. Layer 3 is a fuller-connected layer with rectifier function as activation function. Layer 4 is a convolutional layer intended to grow the dimensionality of the image. Layer 3 and 4 work as the decoder. Layer 5 is intended to crop the image to have the same dimensionality as the original input image, and layer 6 carries out color normalization, making the mean of the pixel values zero and the variance of the pixel values one for the image. Layer 5 and 6 are used to post-process. The weights are initialized by independently randomly sampling from Gaussian distribution given mean and standard deviation.

When training the data, we use root mean square error as the loss function. Then, we use logistic regression with L2-regularization and regularization strength equal to 1 to classify the sentiment of the faces by the representational vectors achieved by the autoencoder.

4.2.3 Convolutional neural network (CNN). The architecture of the convolutional neural network consists of 5 layers. The first three layers are convolutional layers with the rectifier function (ReLU) as the activation function. The fourth layer is a fully connected layer, and the fifth layer is the output layer, which outputs the label for the sentiment. We then train the model on the training data and test it on the validation data. We can then achieve the accuracy (cluster purity) of the classification.

4.3 Experiment results
For k-means clustering algorithm, Figure 3 shows a general increase trend of the cluster purity for both k-means and k-means++ clustering. The purity reaches maximum when the number of clusters $k = 10$ for both cases. When $k = 10$, the random initialization of k-means produces cluster purity $= 0.2306$. On the other hand, k-means++ initialization produces cluster purity $= 0.2358$. Both cluster purities are close to each other with k-means++ initialization a bit better than random initialization. The highest accuracy achieved is 0.2358.

![Figure 3. Cluster Purity vs. Number of Clusters](image-url)
After classification using autoencoder, the cluster purity is achieved and equal to 0.268. Figure 4 gives an example of the original face and the reconstructed face after autoencoder. The autoencoder keeps track of the outline of the original face, and also some critical facial characteristics such as eyes and nose position.

![Figure 4](image)

**Figure 4.** The original face and the reconstructed face after autoencoder

Classification using CNN produces the cluster purity as 0.444. Figure 5 shows that classification using autoencoder have a slightly higher accuracy than classification using k-means++. Classification using convolutional neural network will produce the highest accuracy.

![Figure 5](image)

**Figure 5.** Accuracy of Three Models

5 Conclusions

By comparing the classification using three different algorithms, we find out that classification using convolutional neural network can produce the highest accuracy. In case of requirement of fast training speed with limited dataset, we might choose to use k-means clustering, which will sacrifice accuracy; otherwise, convolutional neural network is a nice algorithm to use. In addition, autoencoder can be combined with convolutional neural network to improve deep neural network [16]. Thus, to pursue a high accuracy when classifying sentiment from human face images, the combination of autoencoder and convolutional neural network is better than k-means clustering.

References

[1] Zhao W, Chellappa R, Phillips P. J and Rosenfeld A 2003 Face recognition: a literature survey *ACM. Compu. Surveys (CSUR)* **35** 399-458
Yang J, Zhang D, Frangi A. F and Yang J. Y 2004 Two-dimensional PCA: a new approach to appearance-based face representation and recognition IEEE Transactions on Pattern Analysis and Machine Intelligence 26.1 131-137

Nefian A. V and Hayes M. H 1999 An embedded HMM-based approach for face detection and recognition Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing 6 3553-56

Pentland A, Moghaddam B and Starner T 1994 View-based and modular eigenspaces for face recognition Computer Vision and Pattern Recognition 94 84-91

Le QV 2013 Building high-level features using large scale unsupervised learning IEEE International Conference on Acoustics, Speech and Signal Processing 8595-98

MacQueen J 1967 Some methods for classification and analysis of multivariate observations Proc. of the fifth Berkeley symposium on mathematical statistics and probability 1 281-297

Hartigan J. A and Wong M. A 1979 Algorithm AS 136: A K-means clustering algorithm J. of the Royal Statistical Society. Series C (Applied Statistics) 28 100-108

Khan SS and Amir A 2004 Cluster center initialization algorithm for K-means clustering Pattern recognition letters 25.11 1293-1302

Arthur D and Vassilvitskii S 2007 K-means++: The advantages of careful seeding Proc. of the eighteenth annual ACM-SIAM symposium on discrete algorithms 1027-35

Tseng G. C 2007 Penalized and weighted K-means for clustering with scattered objects and prior information in high-throughput biological data Bioinformatics 23.17 2247-55

Hinton G. E and Salakhutdinov R. R 2006 Reducing the dimensionality of data with neural networks Science 313.5786 504-507

Vincent P, Larochelle H, Bengio Y and Manzagol P. A 2008 Extracting and composing robust features with denoising autoencoders Proc. of the 25th Int. Conf. on Machine learning 1096-1103

Ng A 2011 Sparse autoencoder CS294A Lecture notes 72.2011 1-19

Kryzhevsky A, Sutskever I and Hinton G. E 2012 Imagenet classification with deep convolutional neural networks Advances in neural information processing systems 1097-1105

Singh K, Malik D and Sharma N 2011 Evolving limitations in K-means algorithm in data mining and their removal Int. J. of Computational Engineering & Management 12 105-109.

Le QV 2015 A Tutorial on Deep Learning Part 2: Autoencoders, Convolutional Neural Networks and Recurrent Neural Networks Google Brain