sciendo

Research Article

Comparative Study of K-NN, Naive Bayes and SVM for Face Expression Classification Techniques

Viola Bakiasi (Shtino)
"Aleksander Moisiu", University of Durres, Albania

Markela Muça
University of Tirana, Albania

DOI: https://doi.org/10.2478/bjir-2023-0015

Abstract

Classification is a data mining technique used to predict group membership for data instances within a given dataset. Classification is considered as an example of supervised learning as training data associated with class labels is given as input. This comparative study focuses on the application of K-Nearest Neighbors (K-NN), Naive Bayes, and Support Vector Machines (SVM) classification techniques in the context of face expression recognition. The study aims to evaluate and compare the performance of these techniques in accurately classifying facial expressions, their advantages and disadvantages.

Facial expression recognition is an important area in computer vision and emotion analysis. It involves identifying and classifying different facial expressions such as happiness, sadness, anger, fear, surprise and disgust. Accurate classification of facial expressions is crucial for various applications, including human-computer interaction, affective computing and psychological research.

K-NN is a non-parametric algorithm that classifies data based on the majority class of its K-nearest neighbors. Naive Bayes is a probabilistic algorithm that makes classification decisions based on the Bayes theorem and assumes independence between features. SVM is a machine learning algorithm that constructs a hyperplane to separate different classes of data.

Dataset such as Affectnet is used to conduct the study including labeled facial expression images. Three algorithms are tested on this dataset to evaluate their performance in correctly classifying facial expressions. Performance metrics such as accuracy, precision, recall, and F1-score are used to assess the classification performance of each technique. Software used for conducting the experiments is Python.

Keywords: Classification techniques, K-NN (K-Nearest Neighbors), Naive Bayes, Facial expression, SVM (Support Vector Machines), Machine learning algorithms.

1. Introduction

This study compares the performance of different classification algorithms for face expression recognition. Data classification is the process of organizing data into categories in such a way that data objects of same group are more similar and data objects from different groups are very dissimilar. In the field of machine learning, classification is a fundamental task that involves categorizing data into different classes or categories based on their features. Classification process is divided into two main steps. The first is the training step where the classification model is built. The second is the classification itself, in which the trained model is applied to assign unknown data object to one out of a given set of class label. To build an effective classification model, it is important to select appropriate features and perform feature selection to reduce dimensionality and improve model performance.

This work naturally can be seen as a continuation of another published paper,[1] where we build a Variational Autoencoder to classify face expression emotions. Since we are dealing with an (image, label) dataset (AffectNet dataset), we cannot use the classical machine learning classification algorithms like kNN, Naive Bayes and SVM, because these algorithms require the dataset to be of the form (features, label). So, the first step would be to extract a set of features as a representative for each image and then use these features to create a new dataset to be fed to the classification algorithms. There are a lot of algorithms to extract features from a given image (e.g., OpenCV library has SIPT and ORB), but we will use the latent space representation of the image as corresponding features. Each image is fed to the encoder network, which maps the image to a vector representation. Since the Variational Autoencoder is trained to reconstruct the image from the latent vector, then this vector holds important and representative information for the image, so it can be used as a feature vector.

Performance metrics such as accuracy, precision, recall, and F1 score were calculated to assess the models' ability to correctly classify facial expressions.

Classification algorithms can be applied to various domains and problems, such as image recognition, spam detection, sentiment analysis, and medical diagnosis. The choice of algorithm depends on the specific problem and the characteristics of the data. The results were compared, and the best-performing algorithm was selected.

2. Classification algorithms

Classification algorithms are machine learning algorithms that are used to categorize or classify data into different classes or categories based on their features. These algorithms are trained on labeled data, where each data point is

[1] Bakiasi (Shtino), V. Muça, M. (2023). Variational Autoencoder for Face Expression Classification. https://iipc cl.org/wp-content/uploads/2023/11/ICSNS-XXX.pdf

associated with a known class label. Once trained, the algorithms can be used to predict the class labels of new, unseen data. Some commonly used classification algorithms include: Support Vector Machines (SVM), K-Nearest Neighbors (KNN), Naive Bayes, Decision Trees, Random Forest, Neural Networks, and Logistic Regression.

2.1. Methodology for using classification algorithms (KNN, Naive Bayes, and SVM) for face expression classification

1. Data Collection: Collect a dataset of face images labeled with their corresponding expressions (e.g., happy, sad, angry, etc.);
2. Data Preprocessing: Preprocess the face images to remove noise, normalize the intensity, and extract relevant features. This may involve techniques such as face detection, alignment, and feature extraction;
3. Feature Extraction: Extract meaningful features from the preprocessed face images;
4. Data Split: Split the dataset into training and testing subsets. The training set is used to train the classification models, while the testing set is used to evaluate their performance;
5. Model Training: Train the KNN, Naive Bayes, and SVM models using the training data. This involves fitting the models to the extracted features and their corresponding expression labels;
6. Model Evaluation: Evaluate the trained models using the testing data. Measure the performance of each model using appropriate evaluation metrics such as accuracy, precision, recall and F1-score;
7. Model Selection: Compare the performance of the different models (KNN, Naive Bayes, and SVM) and select the one that performs best for face expression classification;
8. Model Deployment: Once a model is selected, it can be deployed to classify the expressions of new, unseen face images. This involves applying the chosen model to the features extracted from the new images and predicting their expressions;

2.2. *K-Nearest Neighbors Classification*

K - Nearest Neighbors (K-NN) is one of the simple algorithms in the learning algorithm to predict a class in a dataset. Classification of classes on K-NN based on the closest neighbor's distance using Euclidean distance, Minkowski distance, Manhattan distance, Cosine distance, Hamming distance. The distance is so important for the KNN technique. The most often used distance, is the Minkowski distance of order q where q is an integer. Between two data points $U = (U_1,...,U_n)$ and $V = (V_1,...,V_n)$, this distance is defined by the following formula :

$$d_q(U,V) = (\sum_{i=1}^{n} |U_i - V_i|^q)^{\frac{1}{q}} \quad (1)$$

It is a generalization of both the Manhattan distance (order q = 1) and the Euclidean distance
(Order q = 2).
The KNN algorithm is widely used to determine to which class a new data belongs to. As shown in Figure 1, the predicted class is typically the class that is the most voted in the k nearest neighbors of this data (majority vote of its neighbors).

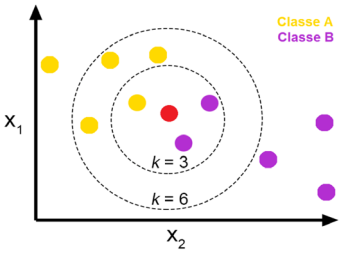

Figure 1. K-Nearest Neighbors Algorithm.

- The K-NN classifier works as follows:

1. Initialize value of K.
2. Calculate distance between input sample and training samples.
3. Sort the distances.
4. Take top K- nearest neighbors.
5. Apply simple majority.
6. Predict class label with more neighbors for input sample.

2.3. Naive Bayes Classification

A Naive Bayes classifier is a probabilistic machine learning model that is used for classification task. The core of the classifier is based on the Bayes theorem:

$$P(y|X) = \frac{P(X|y) * P(y)}{P(X)} \quad (2)$$

Bayes theorem is used for calculating the posterior probability P($y|X$), from P(y), P(X), and P($X|y$). Where: *P($y|X$) is the posterior probability of target class ; P(y) is called the prior probability of class ; P($X|y$) is the likelihood which is the probability of predictor of given class ; P(X) is the prior probability of predictor of class.*

Here X represent the features and y is the class to be predicted. The assumption

made here is that the features are independent of each other, which means that the presence of one particular feature does not affect the other as is not usually the case. Hence it is called naive. The features are represented as a vector $X = (x_1, x_2, ..., x_n)$, and the algorithm takes the form:

$$P(y|X) = \frac{P(x_1, x_2, ..., x_n|y) * P(y)}{P(X)} = \frac{P(x_1|y) * P(x_2|y) * \ldots * P(x_n|y) * P(y)}{P(x_1) * P(x_2) * \ldots * P(x_n)} \quad (3)$$

We calculate this posterior probability for each class y and then classify the sample to the class with maximal posterior probability.

2.4 *Support Vector Machine Classification*

Support Vector Machine is a learning machine algorithm that works on the principle of Structural Risk Minimization (SRM) to find the best hyperplane (Figure 2) that separates two classes in the input space.

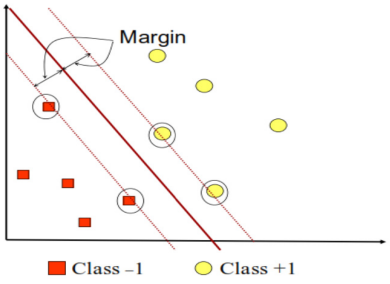

Figure 2. Hyperplane support vector machine

The objective of the Support Vector Machine (SVM) algorithm is to find a hyperplane in an N-dimensional space that distinctly classifies the data points. Here for demonstration purposes, we assume the binary classification case, later we develop ideas for the case with more than 2 classes. The basic version is the Maximal Margin Classifier which supposes that datapoints belonging to different classes can be perfectly separated by linear hyperplanes. Given this assumption, the algorithm aims to maximize the margin the separates datapoints of different classes.

The generalization of the Maximal Margin Classifier to the non-separable case is known as the Support Vector Classifier (SVC). If ($B_0, B_1, \ldots, B_p$) are the coefficients of the separating hyperplane, then a test observation x^* is classified based on the sign of:

$$f(x^*) = B_0 + B_1 + B_2 + \ldots + B_p \quad (4)$$

We construct the Support Vector Classifier based on a set of n training observations

$x_1, x_2, \ldots, x_n \in R^p$ and associated class labels $y_1, y_2, \ldots, y_n$. The SVC hyperplane is the solution of the following optimization problem:

$$maximize_{B_0,B_1,\ldots,B_p,e_0,e_1,\ldots,e_n} \quad M \tag{5}$$

$$s.t. \quad \sum_{j=1}^{p} B_j^2 = 1 \tag{6}$$

$$y_i * (B_0 + B_1 * x_{i1} + B_1 * x_{i2} + \ldots + B_p * x_{ip}) \geq M * (1 - e_i) \quad \forall i \in 1, 2, \ldots, n \tag{7}$$

$$e_i \geq 0 \quad \sum_{i=1}^{n} e_i \leq C \tag{8}$$

Here e_1 , e_2 , . . . , e_n are slack variables that allow individual observations to be on the wrong side of the margin or the hyperplane. C bounds the sum of the ei's, and so it determines the number and severity of the violations to the margin (and to the hyperplane) that we will tolerate. The support vector machine (SVM) is an extension of the support vector classifier that results from enlarging the feature space in a specific way, using kernels. The non-linear nature introduced in the kernel enables the use of non-linear class boundaries, so we can cope with complex non-linear distributions of the datapoints.

The technical details of the SVM models discussed so far assume the binary case, where we have only two classes. In our dataset we have 8 different classes so some modification of the original algorithm should be used. There are two approaches to deal with the multiclass classification.

- **One-to-One approach**, which breaks down the multiclass problem in to multiple binary classification problems. A binary classifier per each pair of classes is used so in total we end up with binary classifiers. In other words, we need a hyperplane to separate between every two classes, neglecting the points of the third class. This means the separation takes into account only the points of the two classes in the current split.
- **One-to-Rest approach** where the breakdown is set to a binary classifier per each class, so the classifier can use k-binary classifiers. In both cases k is the number of classes. In the One-to-Rest approach, we need a hyperplane to separate between a class and all others at once. This means the separation takes all points into account, dividing them into two groups; a group for the class points and a group for all other points.

The SVC algorithm from the library uses the One-to-One approach to implement the multiclass classification.

2.5. Advantages and Disadvantages of K-NN, Naive Nayes, and SVM classification techniques

It is important to note that the choice of classification algorithm depends on the specific problem and the characteristics of the data. K-NN is suitable for simple

implementations and robustness to noise. Naive Bayes is efficient with high-dimensional data and small training datasets, while SVM is effective in high-dimensional spaces and can handle large datasets.

Advantages K-NN	Disadvantages K-NN
Simple implementation	Computationally expensive during testing
No assumptions about data distribution	Sensitive to the choice of k and distance metric
Handles multi-class classification	Requires large memory
Captures complex decision boundaries	Not suitable for high-dimensional data
Robust to noisy data	Can be affected by imbalanced datasets

Table 1. Advantages and Disadvantages of K-NN

Advantages Naive Bayes	Disadvantages Naive Bayes
Simple and computationally efficient	Assumes independence between features
Works well with high-dimensional data	May struggle with imbalanced datasets
Handles both continuous and categorical features	Limited expressive power
Performs well with small training datasets	Sensitive to irrelevant features

Table 2. Advantages and Disadvantages of Naive Bayes

Advantages SVM	Disadvantages SVM
Effective in high-dimensional spaces	Computationally expensive during training
Handles large feature spaces and datasets	Requires careful selection of kernel function and hyperparameter tuning
Robust against overfitting	Difficult to interpret results
Captures complex decision boundaries using different kernel functions	Sensitive to noisy or mislabeled data
Works well with both linearly separable and non-linearly separable data	May struggle with imbalanced datasets

Table 3. Advantages and Disadvantages of SVM

3. Results

We fit the KNN, Naive Bayes and SVM models to the Affectnet dataset in order to evaluate their performance and make a comparative study of their results. The K- Nearest Neighbors is fit with k=10, that is 10 nearest neighbors are taken into account when evaluating the label for a new test observation x_0. Regarding the Naive Bayes algorithm, classifiers differ mainly by the assumptions they make regarding the distribution of $P(x_i|y)$. Naive Bayes learners and classifiers can be extremely fast compared to more sophisticated methods. The decoupling of the class conditional feature distributions means that each distribution can be independently estimated as a one-dimensional distribution. We implement the Gaussian Naive Bayes classifier which assumes the likelihood of features to be Gaussian:

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} exp(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}) \tag{9}$$

Also, we fit the default Support Vector Classifier from the Python library. Next, we show the main classification metrics for each of the algorithms based on the dataset derived from the original Affectnet dataset as explained above.

Class 0 Anger
Class 1 Contempt
Class 2 Disgust
Class 3 Fear
Class 4 Happy
Class 5 Neutral
Class 6 Sad
Class 7 Suprize

	precision	recall	f1-score	support
0	0.48	0.54	0.51	652
1	0.59	0.65	0.62	567
2	0.43	0.34	0.38	488
3	0.53	0.46	0.49	641
4	0.91	0.91	0.91	996
5	0.87	0.88	0.87	1045
6	0.44	0.48	0.46	613
7	0.54	0.52	0.53	807
accuracy			0.64	5809
macro avg	0.60	0.60	0.59	5809
weighted avg	0.64	0.64	0.64	5809

Figure 3. KNN classification metrics

	precision	recall	f1-score	support
0	0.50	0.49	0.49	652
1	0.58	0.63	0.60	567
2	0.38	0.37	0.38	488
3	0.53	0.46	0.50	641
4	0.92	0.90	0.91	996
5	0.87	0.86	0.87	1045
6	0.45	0.49	0.47	613
7	0.53	0.55	0.54	807
accuracy			0.64	5809
macro avg	0.60	0.59	0.59	5809
weighted avg	0.64	0.64	0.64	5809

Figure 4. Naive Bayes classification metrics

Class 0 Anger
Class 1 Contempt
Class 2 Disgust
Class 3 Fear
Class 4 Happy
Class 5 Neutral
Class 6 Sad
Class 7 Suprize

	precision	recall	f1-score	support
0	0.48	0.52	0.50	652
1	0.60	0.60	0.60	567
2	0.42	0.37	0.39	488
3	0.53	0.47	0.50	641
4	0.91	0.92	0.92	996
5	0.88	0.87	0.88	1045
6	0.47	0.48	0.47	613
7	0.53	0.58	0.56	807
accuracy			0.65	5809
macro avg	0.60	0.60	0.60	5809
weighted avg	0.65	0.65	0.64	5809

Figure 5. SVC classification metrics

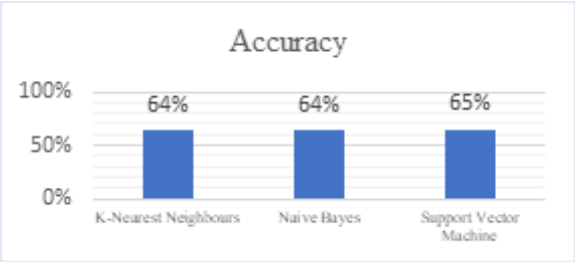

Figure 6. Result accuracy of some classifier on Affectnet database

4. Conclusions

In conclusion, classification algorithms are powerful tools in machine learning for categorizing data into different classes or categories based on their features. This study aims to compare the performance of K Nearest Neighbors, Naive Bayes and Support Vector Machines in the Affectnet dataset, which is a face expression classification dataset. As we said, initially we need to extract some features for each image in the dataset before we use these classification algorithms. To achieve that, we use the latent space vector representation of each image as features and the class label to construct the final dataset. Our latent dimension is 128, so we have projected each image to a 128 feature vector. Almost all three algorithms have the same accuracy. It should be mentioned that we have 8 classes, as shown in the classification metrics above (numbered 0 to 7). In this case the random accuracy would be 100% / 8 = 12.5%, so the accuracy improvement is huge. Also, the state of the art algorithms perform in this accuracy region for this dataset. Since the accuracy is almost identical, 64% for the KNN and Naive Bayes model and 65% for the SVC model, we can observe the differences in the classification metrics within each class. The Support Vector Classifier model has almost the best f1-score for each class, which translates in a slightly better overall accuracy. F1-score is a metric computed from precision and recall as, so it is a more complete indicator of accuracy. It is important to note that the choice of classification algorithm depends on the specific problem and the characteristics of the data. K-NN is suitable for simple implementations and robustness to noise, Naive Bayes is efficient with high-dimensional data and small training datasets, while SVM is effective in high-dimensional spaces and can handle large datasets. Different algorithms have different assumptions and approaches, and their performance can vary depending on the dataset and the task at hand.By understanding their principles, strengths, and limitations, we can effectively apply them to solve real-world problems and make accurate predictions.

References

[1] Shan, C., Gong, S., & McOwan, P. W. (2009). Facial expression recognition based on local binary patterns: a comprehensive study. Image and Vision Computing, 27(6), 803-816.
[2] V. Bakiasi and M. Muça, "Variational Autoencoder for Face Expression Classification", in Thirtieth International Conference On: "Social And Natural Sciences – Global Challenge 2023", pp. 101-110.
[3] I. T. Meftah, N. Le Thanh, and C. B. Amar, "Emotion recognition using KNN classification for user modeling and sharing of affect states," in International Conference on Neural Information Processing, 2012, pp. 234-242.
[4] M. Abdulrahman and A. Eleyan, "Facial expression recognition using support vector machines," in 2015 23nd Signal Processing and Communications Applications Conference (SIU), 2015, pp. 276-279.
[5] Martinez, A. M. (2004). Recognizing imprecisely localized, partially occluded, and expression variant faces from a single sample per class. IEEE Transactions on pattern analysis and machine intelligence, 26(5), 673-685.
[6] Bartlett, M. S., Littlewort, G., Frank, M., Lainscsek, C., Fasel, I., & Movellan, J. (2005). Recognizing facial expression: Machine learning and application to spontaneous behavior. In IEEE International Conference on Systems, Man and Cybernetics (Vol. 4, pp. 3464-3469).
[7] Li, S. Z., & Jain, A. K. (2011). Handbook of face recognition. Springer Science & Business Media.
[8] Hastie, T., Tibshirani, R., & Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer.
[9] Rish, I. (2001). An empirical study of the naive Bayes classifier. In IJCAI 2001 workshop on empirical methods in artificial intelligence (Vol. 3, No. 22, pp. 41-46).
[10] Duda, R. O., Hart, P. E., & Stork, D. G. (2001). Pattern classification (2nd ed.). Wiley-Interscience.
[11] H. Bhavsar and A. Ganatra, "A Comparative Study of Training Algorithms for Supervised Machine Learning", International Journal of Soft Computing and Engineering (IJSCE), Vol. 2, Issue. 4, September 2012.
[12] Vapnik, V. (1995). The nature of statistical learning theory. Springer Science & Business Media.
[13] S.Archana and Dr. K.Elangovan, "Survey of Classification Techniques in Data Mining", International Journal of Computer Science and Mobile Applications, Vol. 2 Issue. 2, February 2014.
[14] Sagar S. Nikam, "A Comparative Study of Classification Techniques in Data Mining Algorithms", Oriental Journal of Computer Science & Technology, Vol. 8, April 2015.
[15] C. Y. Yong, R. Sudirman, and K. M. Chew, "Facial expression monitoring system using pca-bayes classifier," in Future Computer Sciences and Application (ICFCSA), 2011 International Conference on, 2011, pp. 187-191.
[16] I. M. Revina and W. S. Emmanuel, "A survey on human face expression recognition techniques," Journal of King Saud University-Computer and Information Sciences, 2018.
[17] https://www.ijsr.net/archive/v5i1/NOV153131.pdf
[18] https://american-cse.org/csci2015/data/9795a175.pdf
[19]https://www.researchgate.net/publication/333502821_Facial_Expression_Classification_Based_on_SVM_KNN_and_MLP_Classifiers