Comparing Supervised Machine Learning Algorithms for Emotion Recognition Performance

Anzu Hakone

May 9, 2016

1 Introduction

As social creatures, we interact with many people on a daily basis. Recognizing another human's emotions can not only improve social interaction, but also keep one from harm, such as the case when avoiding a person in a hostile mood. Some individuals have visual or neurological conditions that make it difficult to recognize other's emotions. For example, although many visually impaired individuals can still identify emotions from verbal cues, they are still lacking visual information that could complement emotion recognition. Some individuals with autism spectrum disorder (ASP) have trouble both recognizing and expressing facial expressions [?], and therefore, having a tool that computes and reports another person's affect may aid communication.

Other motivations for emotion recognition are either research-based or involve law-enforcement [?]. Companies like Affectiva [?] have already developed real-time facial expression recognition software that uses the psychologist Dr. Paul Ekman's work on basic human emotions – surprise, contempt, fear, happy, sadness, disgust, and anger [?]. These software can be used by other companies for marketing purposes, such as measuring the consumers' reaction to product advertisements. On law-enforcement side, emotion detection can be used for counter-terrorism measures, which may warn illicit activities or uncover inconsistencies in suspects' accounts during interrogation.

Due to the large impact of emotion recognition tools, it is vital that these tools are indeed accurate at detecting emotions. Accuracy of these tools are dependent on the training sample size, machine learning classification algorithms, and generalizability to various cultures and environments. Given the scope of this project, we will use existing facial expression dataset to compare the accuracies of various supervised machine learning algorithms in classifying facial expression data.

2 Related Works

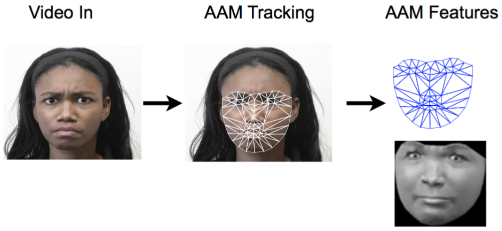

Figure 1: Diagram of how AAMs are extracted from videos as features [?].

2.1 Facial Expression Recognition

As mentioned earlier, there are already commercial interest in studying facial expression recognition, but there are also research done in academia, focusing mostly on comparing machine learning algorithms for speed and performance (leave-one-subject-out cross validation). Bartlett et al. [?] used the Cohn-Kanade dataset (see section ??) to classify six emotions with several machine learning algorithms. On classifying emotion from the PNG images (preprocessed using Gabor filters as edge detection), they found that support vector machines (SVMs) in conjunction with AdaBoost for feature selection resulted in the best speed and accuracy. They were also able to generalize the accuracy to other dataset such as the Ekman-Hager database.

Another study that examined facial expression recognition was by Michel & Kaliouby in 2003 [?]. In addition to using the Cohn-Kanade dataset, they also used real-time video feed that tracked 22 specific face locations (called Active Appearance Models, or AAMs) and used the locations as features that was fed into SVMs with various kernel types (linear, polynomial, radial basis function, and sigmoid). They found that the radial basis function kernels performed best (highest accuracy), with 86.0% accuracy for the Cohn-Kanade dataset and 71.8% for the real-

Table 1: Machine learning algorithms used and their respective libraries [**?**].

Algorithm	Libraries
Linear Discriminant Analysis (LDA)	`discriminant_analysis.LinearDiscriminantAnalysis()`
Support Vector Machine (SVM)	`svm.SVC()`
Stochastic Gradient Descent (SGD)	`linear_model.SGDClassifier()`
K-Nearest Neighbors $k = 5$ (KNN5)	`neighbors.KNeighborsClassifier(n_neighbors=5)`
K-Nearest Neighbors $k = 10$ (KNN10)	`neighbors.KNeighborsClassifier(n_neighbors=10)`
K-Nearest Neighbors $k = 15$ (KNN15)	`neighbors.KNeighborsClassifier(n_neighbors=15)`
Naïve Bayes (NB)	`naive_bayes.GaussianNB()`
Decision Tree (DT)	`tree.DecisionTreeClassifier()`
AdaBoost (AB)	`ensemble.AdaBoostClassifier()`
Label propagation (LP)	`semi_supervised.LabelPropagation()`

time video. However, unlike the present study, Michel and Kaliouby did not test their data using other classification algorithms.

2.2 Extended Cohn-Kanade Dataset

The Extended Cohn-Kanade Dataset (CK+) [**?**, **?**] is a database developed by Carnegie Mellon University and the University of Pittsburgh, and it contains facial expression data that can be used to study automatic facial expression detection. Data were taken from 123 adults of various age, sex, and race, performing specific instructed facial displays – neutral, surprise, anger, contempt, disgust, fear, happiness, sadness, and surprise.

The dataset includes the following data:

- 10,709 PNG images (640x490 pixels) of the subjects demonstrating the instructed facial expressions in sequences.
- 593 Facial Action Coding System (FACS) Action Unit (AU) Labels for the peak sequences (last sequence of each facial expression). AUs are specific facial muscle movements that are used to describe and taxonomize facial expressions (e.g., AU1=Inner Brow Raiser, AU12=Lip Corner Puller). The AUs for the CK+ dataset were coded by two certified FACS coders.
- 10,709 Active Appearance Models (AAMs) Landmarks (593 peak sequences), which are the x and y positions of the 68 facial landmark points on the image screen (Figure **??**). AAMs landmarks are used to track and locate specific features of the face in computer vision (e.g., corners of the eye, mouth).
- 327 Emotion Labels for the sequences. Only 327 of the 593 sequences have emotion sequences since some sequences did not correspond with the typical depiction of the emotions. The labeled emotions were coded as 0=neutral, 1=anger, 2=contempt, 3=disgust, 4=fear, 5=happy, 6=sadness, and 7=surprise.

3 Methods

This present study used Scikit-learn's machine learning algorithm libraries and the CK+ dataset to classify 1) emotion from the AAM landmarks, 2) emotion from the AUs, and 3) AUs from the AAM landmarks.

The images were not used to classify emotion or AUs because images and AAMs are both continuous measures of facial positions. Additionally, images have more noise than the AAMs because we would have had to run edge detection on the images, which may include features (e.g., face shape, age-related wrinkles) that are not indicative of facial expressions.

3.1 Dataset Parsing

All training and testing data were from the Extended Cohn-Kanade dataset [**?**, **?**]. The neutral and peak expression AAMs landmarks were parsed into two separate two-dimensional lists ((2 x 68) x 593), `aam0` and `aam` respectively. The machine learning algorithms were performed on a two-dimensional list ((2 x 68) x 593) of the difference between the peak expression and neutral landmarks, representing the AAM landmarks as vectors `aamvec`. In order to discretize the continuous features, all features were converted to floats with one decimal point precision.

The AUs were converted into a two-dimensional list (64 x 593) with 1 representing that the feature (i.e., AU) is present, and 0 if it is absent. When using AUs as the output labels, each algorithm was run 39 times, one for each AUs present in the dataset (some AUs were not present in any of the sequences).

The emotion data were parsed in two ways. The first round of testing converted the emotion data into a one-dimensional list (`len`=593) with the same emotion labels as the CK+ dataset (see section **??** for the

Table 2: Mean accuracies, z scores, and p values for the first round of classifications, using one-dimensional list of emotions.

Algorithm	AAM → Emotion	z (p)	AAM → AU	z (p)	AU → Emotion	z (p)
LDA	0.172	-0.13 *(0.45)*	0.876	0.22*(0.41)*	0.895	0.60*(0.27)*
SVM	0.255	**1.96 *(0.02)***	0.907	0.65*(0.26)*	0.862	0.25*(0.40)*
SDG	0.178	0.02 *(0.49)*	0.777	-1.12*(0.13)*	0.892	0.57*(0.28)*
KNN5	0.172	-0.13 *(0.45)*	0.899	0.54*(0.29)*	0.886	0.51*(0.31)*
KNN10	0.138	-0.98 *(0.16)*	0.905	0.62*(0.27)*	0.889	0.54*(0.29)*
KNN15	0.194	0.41 *(0.34)*	0.905	0.62*(0.27)*	0.868	0.32*(0.38)*
NB	0.148	-0.75*(0.23)*	0.670	**-2.58*(0.005)***	0.760	-0.80*(0.21)*
DT	0.138	-0.98*(0.16)*	0.859	-0.005*(0.50)*	0.855	0.19*(0.43)*
AB	0.240	1.57*(0.06)*	0.893	0.45*(0.32)*	0.572	**-2.75*(0.002)***
LP	0.138	-0.98*(0.16)*	0.904	0.61*(0.27)*	0.892	0.57*(0.28)*

labels). The second round used a two-dimensional list (7 x 593) with 1 representing that the feature (i.e., specific emotion) is present, and 0 if it is absent. When using the two-dimensional list of emotions as the output labels, each algorithm was run 7 times, one for each emotion. Because not all of the landmarks had corresponding emotion labels, sequences without emotion labels were omitted from the training and testing data.

3.2 Procedure

The machine learning algorithms and the libraries used, as well as the specific parameters for the functions are represented in Table **??**. Performances of the machine learning algorithms were evaluated by leave-one-out cross-validation. K-folds ($K = n = 593$) were generated by Scikit-learn's `sklearn.cross_validation.KFold` library. The training data of size $k - 1 = 592$ was used in the `fit(train_input, train_output)` function for all algorithms. The accuracy of each algorithm was derived from taking the mean accuracy of the testing data using each algorithm's `score(test_input, test_output)` function. The Z-scores and their respective p-values were then calculated by `scipy.stats.mstats.zscore()` and `scipy.stats.norm.sf()`.

Both rounds of testing classified emotion from AAMs (AAM → Emotion), AUs from AAMs (AAM → AU), and emotion from AUs (AU → Emotion). In the first round of testing, we used the one-dimensional list for emotion, and so there were seven classes. In the second round of testing, we used the two-dimensional list for emotion, which meant that there were two classes (e.g., anger versus not anger).

4 Results

The mean accuracies of each of the algorithms on the three classifications of the first round of testing (i.e., using the one-dimensional emotion outputs) are summarized in Table **??**. The mean accuracies of the second round of testing (i.e., using the two-dimensional emotion outputs) are summarized in Table **??**. The classifications that performed significantly above or below the average are bolded. The averages of the mean accuracies of each classification were as follows: AAM→emotion (one-dimensional) = 0.178, AAM→AU = .860, AU→emotion (one-dimensional) = 0.837, AAM→emotion (two-dimensional) = 0.796, and AU→emotion (two-dimensional) = 0.936.

5 Discussion

This study examined the performance of various machine learning algorithms on classifying facial expression data. Overall, we were able to classify both emotion and AUs from the facial expression data in the CK+ dataset, with most average accuracies being in the 85% range. Below, we discuss the specifics of the classification results.

5.1 Classifying Emotion from AAMs

In the first round of testing with the one-dimensional list of emotions as outputs, SVM had the highest accuracy, and the difference between SVM and the other algorithms were significant, but the mean accuracy of classifying emotion from AAMs was the lowest of all the classifications. There are two main reasons that may contribute to such a low mean performance. First, the number of samples were small since only 327 out of the 593 sequences had emotion labels. Second, there were seven possible classes (seven emotions), which makes the sample size per emotion even smaller.

Table 3: Mean accuracies, z scores, and p values for the classifications of round two, using two-dimensional list of emotions.

Algorithm	AAM → Emotion	z (p)	AU → Emotion	z (p)
LDA	0.749	-0.52 *(0.30)*	0.969	0.40*(0.35)*
SVM	0.860	0.73*(0.23)*	0.946	0.12*(0.45)*
SDG	0.703	1.05*(0.15)*	0.956	0.25*(0.40)*
KNN5	0.853	0.65*(0.26)*	0.972	0.44*(0.33)*
KNN10	0.860	0.73*(0.23)*	0.970	0.42*(0.34)*
KNN15	0.856	0.69*(0.25)*	0.965	0.35*(0.36)*
NB	0.580	**-2.44*(0.01)***	0.693	**-2.99*(0.001)***
DT	0.799	0.03*(0.49)*	0.959	0.28*(0.39)*
AB	0.839	0.49*(0.31)*	0.963	0.33*(0.37)*
LP	0.858	0.70*(0.24)*	0.969	0.40*(0.34)*

In the second round of testing with the two-dimensional list of emotions, the mean accuracy was higher than that of using the one-dimensional list, most likely because there were more samples per emotion. As with the first round of testing, SVM performed the best, along with KNN of $k = 10$. Our results confirm Michel and Kaliouby's accuracy results of 86% for SVMs [?]. Naïve Bayes performed significantly worse than the other algorithms, probably because Naïve Bayes assumes conditional independence between features [?], and the AAMs were given as x and y coordinates of the landmarks, therefore half of the 136 features were dependent on the other half.

5.2 Classifying AUs from AAMs

The classification of AUs from AAM features were overall very good, with an average mean accuracy of 86%, and as high as 90.7% with SVMs. Only the Naïve Bayes algorithm had significantly lower accuracy than the other algorithms, which was consistent with the AAM→emotion classification, most likely due to the nature of the AAM features as explained above. Although not explored in this study, the extracted AUs can then be further used to classify emotion, since classifying emotion from AU had greater accuracy than from AAMs.

5.3 Classifying Emotion from AUs

The performance for classification of emotion from AUs were overall very high. This was expected, since there are already AU criteria for emotions, such as AU23 and AU24 must be present for anger [?]. It was also expected that the second round of testing would result in higher performance, since there would be more samples per emotion. The average performance in classifying AUs from AAMs were higher than that of classifying emotion from AUs (first round), but this may be due to the smaller sample size for the AU→emotion classification ($n = 327$) compared to the AAM→AU classification ($n = 593$). This reasoning is supported by the higher performance in AU→emotion compared to AAM→AU in the second round of testing.

It is unclear why AdaBoost had a significantly lower performance than the rest in round one of testing, especially since the Scikit-learn's AdaBoost function defaults to using decision trees as weak learners, and decision trees performed slightly above average. Future work should involve testing various weak learner algorithms for AdaBoost. In the second round of testing, Naïve Bayes performed significantly worse than the rest, most likely since, like AAMs, AUs are not independent features for classifying emotion.

6 Limitations and Future Work

Although this study gave insight into which Scikit-learn algorithms had better accuracy than others for facial expression recognition on the CK+ database, certain factors need to be addressed before generating an exhaustive list of well-performing algorithms or generalizing the algorithms' performance to all facial expression recognition. For one, this study did not test various parameters for the algorithm functions, nor did it explore feature selection. Previous studies like Bartlett et al. [?] has indicated that feature selection using AdaBoost and then performing SVM has better accuracies than running SVM or AdaBoost alone.

Another limitation was the possibility to overfitting to the CK+ dataset. Therefore, in order to make a general statement of classifying emotion, the machine learning algorithms should be tested on different AAM, AU, and emotion-labeled datasets. Lastly, if this work were to be used towards developing a real-time facial expression detection tool, computation speed and generalizability to different subject environment (e.g., race, dark room) must also be taken into consideration.

References

[1] Affectiva. (2015, May). *How to Teach a Machine to Recognize Emotions*. Retrieved from http://www.affectiva.com/blog/how-to-teach-a-machine-to-recognize-emotions-part-1/

[2] American Psychiatric Association. (2013). Diagnostic and statistical manual of mental disorders: DSM-5. Washington, D.C: *American Psychiatric Association.*

[3] Bartlett, M. S., Littlewort, G., Frank, M., Lainscsek, C., Fasel, I., & Movellan, J. (2005). "Recognizing Facial Expression: Machine Learning and Application to Spontaneous Behavior". *Computer Vision and Pattern Recognition, 2*, 568-573.

[4] Dwoskin, E., & Rusli, E. M. (2015, January 28). The Technology that Unmasks Your Hidden Emotions. *The Wall Street Journal*. Retrieved from http://www.wsj.com/articles/startups-see-your-face-unmask-your-emotions-1422472398

[5] Ekman, P. (1999). Basic Emotions. In Dalgleish, T. & Power, M. J. (Eds.), *Handbook of Cognition and Emotion* (pp. 45-60). New York, NY: John Wiley & Sons Ltd.

[6] Kanade, T., Cohn, J. F., & Tian, Y. (2000). "Comprehensive database for facial expression analysis". *Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition (FG'00)*, Grenoble, France, 46-53.

[7] Lucey, P., Cohn, J. F., Kanade, T., Sraragih, J., Ambadar, Z., & Matthews, I. (2010). "The Extended Cohn-Kanade Dataset (CK+): A complete expression dataset for action unit and emotion-specific expression". *Proceedings of the Third International Workshop on CVPR for Human Communicative Behavior Analysis (CVPR4HB 2010)*, San Francisco, USA, 94-101.

[8] Michel, P., & Kaliouby, R. E. (2003). "Real Time Facial Expression Recognition in Video using Support Vector Machines". *Proceedings of the 5th international conference on Multimodal interfaces*, Vancouver, British Columbia, Canada, 258-264.

[9] Mitchell, T. (1997). Machine Learning. Portland, OR: McGraw-Hill.

[10] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. & Duchesnay, E. (2011). "Scikit-learn: Machine Learning in Python". *JMLR, 12*, 2825-2830.