Emotional Expression Recognition using Support Vector Machines

Melanie Dumas
Department of Computer Science
University of California, San Diego
La Jolla, CA 92193-0114
mdumas@cs.ucsd.edu

Abstract

The objective of this paper is to apply Support Vector Machines to the problem of classifying emotion on images of human faces. This well-defined problem is complicated by the natural variation in people's faces, requiring the classification algorithm to distinguish the small number of relevant features from the large pool of input features. Recent experimentation using neural networks has achieved over 85% classification accuracy. These experiments provide a metric for evaluation of the Support Vector Machine technique, which was shown to have equivalent performance to neural networks.

1 Introduction

Classification of emotion on faces of actors has traditionally been perceived as a task only for humans. Given a set of photos, humans will classify emotion consistently 91.7% [Ekman, 1976] of the time. However, recent work with neural networks have determined that computers are similarly capable, achieving an accuracy of 85.9% [Dailey *et al.*, 2000]. The objective of this paper is to apply a new technique, Support Vector Machines, to the problem of emotion classification in an attempt to increase accuracy.

Support Vector Machines (SVMs) view the classification problem as a quadratic optimization problem. The technique has successfully been applied to standard classification tasks, such as text classification [Joachims, 1998a] [Joachims, 1998b] and medical diagnosis [Morik *et al.*, 1999]. SVMs avoid the "curse of dimensionality" by placing an upper bound on the margin between the different classes, making it a practical tool for large, dynamic datasets. The feature space may even be reduced further by selecting the most distinguishing features through minimization of the feature set size [Fung and Mangasarian, 2000].

SVMs plot the training vectors in high-dimensional feature space, and label each vector with its class. A hyperplane is drawn between the training vectors that maximizes the distance between the different classes. The hyperplane is determined through a kernel function, which is given as input to the classification software. The kernel function may

be linear, polynomial, radial basis, or sigmoid. The shape of the hyperplane is generated by the kernel function, though many experiments select the polynomial kernel as optimal [Morik *et al.*, 1999][Joachims, 1998b].

This paper will introduce the details of SVMs in Section 2. The software package used to run the analysis is described in section 3. Section 4 describes the dataset, which is used in the experiments detailed in Section 5. Concluding remarks and observations complete the paper in Section 6.

2 Support Vector Machine Details

Support Vector Machines classify data through determination of a set of support vectors, through minimization of the Structural Risk. The support vectors are members of the set of training inputs that outline a hyperplane in feature space. This *l*-dimensional hyperplane, where *l* is the number of features of the input vectors, defines the boundary between the different classes. The classification task is simply to determinate which side of the hyperplane the testing vectors reside in. Minimizing the structural risk reduces the average error of the inputs and their target vectors. In the description that follows, training data are classified into binary classes.

The support vector algorithm approximately performs Structural Risk Minimization. Given a set of training examples $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), ..., (\mathbf{x}_l, y_l)$, if there is a hyperplane that separates the positive and negative examples, than the points $\mathbf{x}$ which lie on the hyperplane satisfy $(\mathbf{w} \cdot \mathbf{x_i}) + b = 0$, where $\mathbf{w}$ is normal to the hyperplane and b is the distance from the origin. The margin of a separating hyperplane is defined as the shortest distance to the closest positive or negative example. The support vector algorithm looks for the separating hyperplane with the largest margin.

SVMs provide a generic mechanism to fit the surface of the hyperplane to the data through the use of a kernel function. The user may provide a function, such as a line, polynomial, or sigmoid curve, to the SVM, which selects support vectors along the surface of this function. This capability allows a broader range of problems to be classified, since the user may input any function, customized to a specific dataset. In the case of linearly inseparable datasets, the cost of misclassification is accepted through the use of 'slack variables'.

An exciting property of SVMs is how the "curse of dimensionality" is avoided by the upper bound on the VC-dimension. The VC (Vapnik-Chervonkis)-dimension measures the capacity of the machine (i.e. the ability to learn any training set without error). This bound does not depend on the dimensionality, but on the separation margin between the classes [Joachims, 1998b].

3 LIBSVM

The SVM package used for experimentation is LIBSVM. This package is under active development and has several advantages over other packages. LIBSVM is developed by Chih-Chung Chang and Chih-Jen Lin [Chang and Lin, 2001] and its features include parameterized kernel functions, different SVM formulations (variable optimization algorithms), and multi-class classification.

3.1 Kernel Functions

The LIBSVM package provides four different standard kernels, which the user defines during training. The definitions of the kernel functions that follow include the use of parameters such as γ, c, and *degree* that are defined by the user during training. **u** is the testing vector, and **v** is the support vector.

Kernel	Formula
Linear	$\mathbf{uv}$
Polynomial	$(\gamma\mathbf{uv}+c)^{degree}$
Radial Basis Function	$exp^{(\gamma\lvert\mathbf{uv}\rvert^2)}$
Sigmoid	$tanh(\gamma\mathbf{uv}+c)$

3.2 SVM Formulations

LIBSVM allows customization of the formulations, or the decision functions used to classify the data. Determining the appropriate decision function increases accuracy specific to the dataset.

Formulation	Features
C-Support Vector Classification (C-SVC)	C provides an upper bound on the VC-dimension
nu-Support Vector Classification (nu-SVC)	nu bounds the fraction of training errors and support vectors

The LIBSVM package also provides a One-class SVM and ε- and μ-regression formations. For classification of emotions on faces, One-class SVM is not applicable. This formulation is used to estimate the support vectors in a dataset with high dimensionality, and is not used for classification. In addition, the ε-regression and μ-regression formulations (which bound the error and number of support vectors, respectively) use regression to progressively decrease the error. This method works well for determining probabilities of class membership, and is not applicable to recognizing emotion classes.

3.3 Multi-class classification

LIBSVM runs a "one-against-one" classification for each of the k classes. $\frac{k(k-1)}{2}$ classifiers are actually generated to train the data, where each training vector is compared against two different classes and the error (between the separating hyperplane margin) is minimized. The classification of the testing data is accomplished by a voting strategy, where the winner of each binary comparison increments a counter. The class with the highest counter value after all classes have been compared is selected.

4 Dataset

The Pictures of Facial Affect (POFA) dataset [Ekman, 1976] was selected for experimentation due to its large number of features and high inter-subject classification accuracy. The dataset provides a six-way classification (Happy, Sad, Afraid, Angry, Surprised, and Disgusted) of static human facial photographs. The pictures were digitized, cropped, and scaled, leaving only the central facial features.

The dataset contained 95 labeled examples from 13 subjects, where each example had 40,600 features. The large number of features is the result of Gabor filtering applied to each image. Gabor filtering selects a region around each pixel and applies a series of masks used to determine the contours of that region. This dataset was generated by Matt Dailey and the process is detailed in [Dailey *et al.*, 2000].

Recent experimentation determined that not all features have an equal effect on the emotion classification [Padgett and Cottrell, 1998]. Examples include the whites around the eyes are more representative of fear, and wrinkles around the mouth represent happiness. SVMs do not provide a mechanism for feature selection, and the all features are weighted equally.

5 Experimental Results

The following experiments are designed to answer the following questions:

- What is the highest accuracy attainable by SVMs on the POFA dataset?
- Which kernel functions were used to achieve this accuracy?
- Which parameters were most influential in the performance of the SVM?

The POFA dataset was divided into twelve subjects for training and one subject for testing. For each experiment, thirteen tests were performed by varying which individual was used as the test subject, then the mean accuracy was recorded.

5.1 Binary and Multi-class comparison

The dataset was adjusted for binary classification. One class was selected, for example happiness. All happy faces were classified as +1, and all non-happy faces were classified as -1. The resulting comparison is simply binary. The next class is selected and the positive and negative examples are reclassified for training, and the mean score of all six binary classifications is recorded. Binary classification outperformed the multi-class classifier provided by the LIBSVM package in all cases, as shown in Figure 1.

The multi-class comparison generates a high number of classifiers, reducing the precision of the classification, since many irrelevant classifiers are compared and their results are contribute to the overall classification. Using a binary, single classifier isolates the single class from all other classes, and is shown to have higher performance.

5.2 Kernel and its Formulations

Referring again to Figure 1, the linear kernel for binary classification achieved slightly higher accuracy than the rest of the kernels. This result indicates that the separating plane between all of the different classes is simply linear. The polynomial, radial basis, and sigmoid kernels all classified the same examples correctly using binary classification, throughout the cross validation experiments. This result is interesting, since each function is attempting to draw basically a line between the classes, and they accomplish this with similar efficiency.

For the multi-class comparison, one surprising result is the kernel function that generated the highest accuracy is the Sigmoid kernel. There was a high degree of variation using the different kernels with multi-class classification.

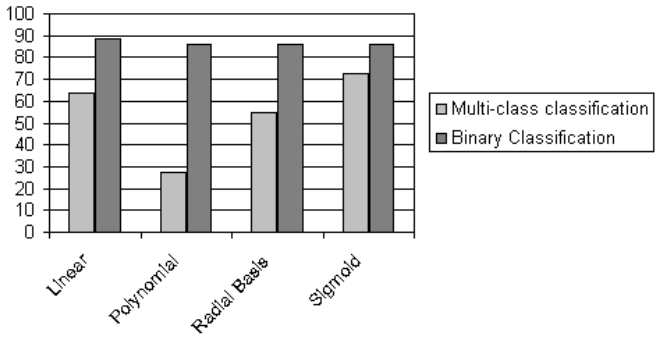

Figure 1: Binary vs multiclass comparison, over all kernels

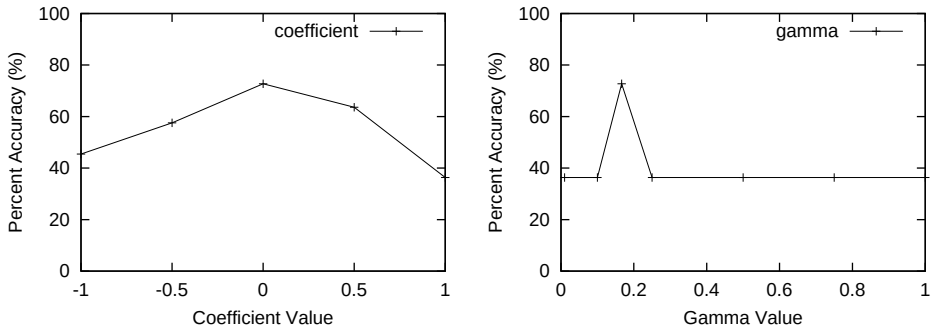

Figure 2: Accuracy of Sigmoid kernel adjusting coefficient and gamma

For the multi-class classifications, the C-SVC and nu-SVC formulations were compared, and C-SVC consistently outperformed the nu-SVC formulation. One reason C-SVC performed so well is due to the penalty acquired when an example is misclassified. The default penalty of 1 achieved the highest accuracy. The nu-SVC formulation controls the number of support vectors and bounds the error. Without the additional penalty of misclassification, the decision values obtained by nu-SVC were not as accurate. This indicates that the bounds on number of support vectors and the error do not impact the overall accuracy of the decision functions. Only the C-SVC formulation worked with the binary classification.

5.3 Optimizing the Sigmoid Function: $tanh(\gamma \mathbf{uv} + c)$

Attempts to optimize the Sigmoid kernel using the multi-class classifier included determining the center of the tanh function. The coefficient parameter, *c*, in the kernel function explained above, was varied in this test. The coefficient parameter *c* adjusts the center of the tanh function along the x axis. The coefficient has no affect on scaling the sigmoid or adjusting its y-axis elevation. The default was set to 0 and the default was determined to be the optimal setting for this experiment. See Figure 2.

The gamma parameter in the Sigmoid kernel scales the width of the sigmoid itself. The Sigmoid is wider with smaller gamma values, and narrow with larger gamma values. Figure 2 clearly shows a peak at 0.1667, or 1/6. This is 1 over the number of classes. This

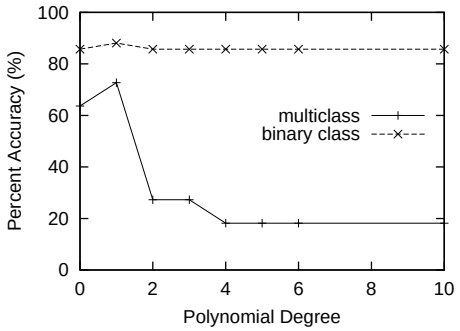

Figure 3: Accuracy by varying degree of Polynomial kernel

value essentially normalizes the support and testing vectors to a restricted range to optimize performance.

5.4 Polynomial Degree

Previous papers [Morik *et al.*, 1999][Joachims, 1998b] showed optimal performance with the polynomial kernel, which lead to this experiment adjusting the degree of the polynomial to try to increase accuracy. For the binary classification case, the optimal hyperplane was a line, independent of the polynomial degree. This is indicated by its steady accuracy shown in Figure 3, with an insignificant peak at degree = 1. For the multi-class classification, the graph shows a peak and steady decline of accuracy at degree = 1.

5.5 Neural Net Comparison

Determining the overall performance of the SVM requires comparison against a proven technique, such as neural networks. In a recent experiment [Dailey *et al.*, 2000], a single layer neural network classified the POFA dataset with 85.9% accuracy. This network contained 216 input units, 6 output units, and no hidden units. The time for training was 210 seconds and the time for classification was 0.26 seconds/pattern. The peak accuracy for SVMs is 85.7%. Time for training is comparable with 223 seconds, and each pattern was classified on average in 62 seconds.

Compared to the SVM techniques, neural nets achieve similar accuracy in less time. This difference in time can be immediately explained by the number of features which were selected for training and testing. The neural net in the paper above selected 216 features using Principal Components Analysis. The smaller input size significantly reduced the classification time.

6 Concluding Observations

The objective of this paper is to determine the highest possible accuracy attainable with SVMs classifying the POFA dataset, and compare its performance against recent experimentation. The large number of features and well-defined classifications seemed to be a rich dataset for SVMs. Experimental results determined that for all binary classifications, SVMs achieved comparable performance to single-layer neural networks. The highest ac-

curacy value obtained by the SVMs was using the Linear kernel with the C-SVC formulation, generating a mean accuracy of 88.1%. The standard deviation of 2% is comparable to the neural net's performance of 85.9% accuracy. We believe that higher accuracy may be obtained through several optimizations.

One possible optimization would be designing a kernel function that will specifically handle a six-way classification. This was suggested by Olvi Mangasarian in a personal correspondence.

This dataset was not optimized for this implementation of Support Vector Machines. Reduction of the number of features may lead to higher performance, and certainly reduce the classification time. Many features in the POFA are inherently noisy, due to the natural variations in human faces. This would require a feature selection technique, similar to a minimum set of features suggested by [Fung and Mangasarian, 2000].

Acknowledgements

Special thanks are extended to Matt Dailey for his patient answers to many questions, Gary Cottrell, Olvi Mangasarian, and Joe Drish.

References

[Chang and Lin, 2001] Chih-Chung Chang and Chih-Jen Lin. Libsvm: a library for support vector machines (version 2.3). 2001.

[Dailey *et al.*, 2000] Matt Dailey, Garrison Cottrell, and Ralph Adolphs. A six-unit network is all you need to discover happiness. In *Proceedings of the Twenty-Second Annual Conference of the Cognitive Science Society*. Mahwah, NJ. Erlbaum, 2000.

[Ekman, 1976] Paul Ekman. Pictures of facial affect. 1976.

[Fung and Mangasarian, 2000] Glenn Fung and O. L. Mangasarian. Data selection for support vector machine classifiers. In *Knowledge Discovery and Data Mining*, pages 64–70, 2000.

[Joachims, 1998a] Thorsten Joachims. Making large-scale SVM learning practical. Cambridge, MA: MIT Press, 1998.

[Joachims, 1998b] Thorsten Joachims. Text categorization with support vector machines: learning with many relevant features. In Claire Nédellec and Céline Rouveirol, editors, *Proceedings of ECML-98, 10th European Conference on Machine Learning*, pages 137–142, Chemnitz, DE, 1998. Springer Verlag, Heidelberg, DE.

[Morik *et al.*, 1999] Katharina Morik, Peter Brockhausen, and Thorsten Joachims. Combining statistical learning with a knowledge-based approach - a case study in intensive care monitoring. In *Proc. 16th International Conf. on Machine Learning*, pages 268–277. Morgan Kaufmann, San Francisco, CA, 1999.

[Padgett and Cottrell, 1998] C. Padgett and G. Cottrell. A simple neural network models categorical perception of facial expressions. In *In Proc. 20th Cognitive Science Conference*, pages 806–807. Mahwah, NJ. Erlbaum, 1998.