MINIMUM SURFACE BHATTACHARYYA FEATURE SELECTION

José Andrés González, Michael J. Mendenhall

Air Force Institute of Technology
Electrical & Computer Engineering
Wright-Patterson AFB, Ohio 45433

Erzsébet Merényi

Rice University
Electrical & Computer Engineering
Houston, TX 77251

ABSTRACT

This paper introduces a novel feature selection method called Minimum Surface Bhattacharyya (MSB). The method is applicable for multiple class problems utilizing supervised training. The minimum surface method selects features by means of inter-class separability. For the purposes of this paper, the method is applied to a hyperspectral data set with high correlations among the features. The method shows promise for hyperspectral analysis due to its speed and demonstrated capacity to improve classification performance.

Index Terms— feature selection, dimensionality reduction, Bhattacharyya coefficient, machine learning

1. INTRODUCTION

Hyperspectral images require significant amounts of memory due to the passive recording of hundreds of image bands. Pending one's application, storage, transfer and/or analysis of hyperspectral images may be problematic due to their size. In order to reduce the impact of high-dimensional data on transfer rates and processing time, feature selection may be used to determine a relevant feature set. Additionally, a small feature subset may improve classification performance by maintaining features pertinent to the classification task.

Feature selection methods typically suffer in two areas. First, many have significant runtime time complexities, often making them impractical outside of a research environment. Second, some do not handle highly correlated data effectively. A common feature selection taxonomy (i.e., wrapper, filter and embedded methods) is fitting for categorizing the observed trends in the compared methods [1]. For example, one may search the feature subset space with a wrapper method for well performing features utilizing classification performance as an evaluation measure. Subset search utilizing heuristics (best first search) or stochastic (genetic algorithm) search methods result in small subsets with good performance for a given classifier, but require significant computational resources for large feature sets [1]. Faster feature selection methods (filter methods), such as RELIEF-F and the weighted sum of Probability of Error and Average Correlation Coefficient (POEACC), analyze features individually and not as subsets [1]. Filter methods significantly reduce time complexity, but for some data sets, these methods do not adequately handle dependency among features. Embedded feature selection methods determine features as part of a classification learning paradigm. Examples of embedded methods include the C4.5 decision tree and Generalized Relevance Learning Vector Quantization Improved (GRLVQI) [1, 2]. C4.5 uses entropy for generating a decision tree classifier and an error rate estimation method for pruning branches. GRLVQI uses a prototype-based neural paradigm, while updating the relevances of features in concert with evaluating the classification performance and prototype learning.

Utschick [3] discusses feature selection based on the separability between classes for a given feature for a multiple-class classification problem. The Bhattacharyya coefficient is used as a measure of the separability of two classes, a and b, for a given feature f. The Bhattacharyya coefficient for a given feature, B_f, is calculated by

$$B_f = \sum_{i}^{k} \sqrt{p_i^a p_i^b}, \tag{1}$$

where there are k bins of the data histogram, and p_i represents the probability (or contribution) of a bin in a feature's histogram in respect to samples of a given class, a or b. The Bhattacharyya coefficient ranges from [0,1], where a value of one indicates that the two histograms are identical and zero indicates no overlap of the histograms. The mean Bhattacharyya coefficient of all class pairs has been used as the measure for comparing features among multiple classes. Sorting by the mean is analogous to sorting by the median Bhattacharyya coefficient, since both represent differing perspectives of the "center" of a sample set. Correlation is shown between the predictive accuracy of a classifier and the Bhattacharyya coefficient of the features used in training. Utschick [3] assumes the distributions of the features among the classes are Gaussian in order to utilize the Bhattacharyya coefficient. Thacker, *et al.* [4] show the Gaussian assumption need not be made; the Bhattacharyya coefficient may be used as a measure for a data set with any distribution of the features.

Utilizing the Bhattacharyya coefficient results in a O(S) runtime in respect to the number of samples (S). This is due

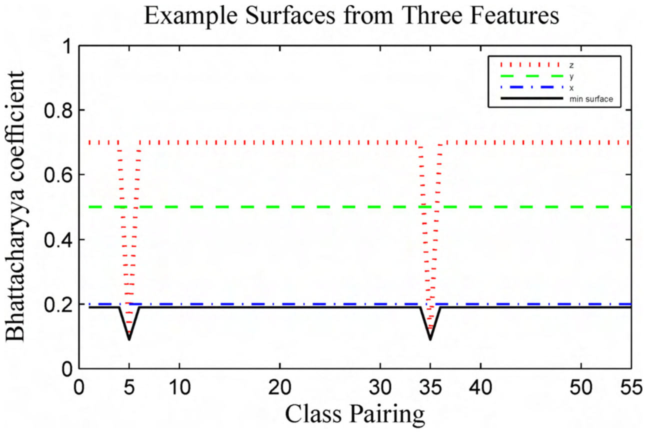


Fig. 1. The mean performance for this example is 0.2, 0.5, and 0.6782 for x, y, and z, respectively. The current minimum surface consists mostly of feature x and two class pairs of feature z (5 and 35).

to the fact that there is a constant number of passes through all the data for the sake of generating counts and binning the class histogram of the feature. The runtime in respect to the number of features (N) would still be O($NlogN$) due to the sorting of the features by Bhattacharyya value. For the purposes of this paper, this feature selection approach is referred to as the mean or median Bhattacharyya method. The runtime performance for this method is highly efficient in comparison to the other feature selection methods discussed. The sorting of features by the median Bhattacharyya coefficient does not contribute to the worst case runtime performance as long as the number of overall samples is significantly greater than the number of features.

2. FEATURE SELECTION METHODOLOGY

The feature selection methods discussed (genetic algorithm, best first search, RELIEF-F, POEACC, C4.5, GRLVQI and median Bhattacharyya) are compared to a novel feature selection method to determine their performance on a hyperspectral data set. The feature selection methods are performed on the data set using the training sets from a three-fold stratified cross validation to provide unbiased feature evaluations. The classification performance of ordered subsets from the feature selection methods are used to compare performance. The ordered subsets for each feature selection method are generated using the feature rankings created by a given feature selection method. For the feature methods that generate subsets and not inherent rankings (or weightings) the features are ranked based on the number of folds that selects the feature. The classification performance of a subset is determined with a minimum euclidean distance (MED) classifier [2]. The Lunar Crater Volcanic Field (LCVF) scene by AVIRIS is analyzed for selecting relevant image bands and provides 23 and 35-class problems [5]. This section continues by presenting the novel minimum surface Bhattacharyya (MSB) method.

Algorithm 1 Minimum Surface Bhattacharyya Pseudocode

```
Require: Array $f[0, number\_of\_features-1]$ of $value$ arrays {a $value$ array per feature}
Require: Arrays $value[0, number\_of\_class\_pairs - 1]$ of real numbers {a Bhattacharyya coefficient per class pair}
  $sorted\_list \leftarrow <empty\ list>$
  $open\_list \leftarrow [1, number\_of\_features]$ {feature indices}
  while $open\_list$ is not empty do
    $min\_value \leftarrow \min(f, open\_list)$ {minimum Bhattacharyya coefficients for each class pair in $open\_list$}
    $min\_surface \leftarrow <empty\ list>$
    for $i = 0$ to $number\_of\_class\_pairs$ do {find the features that generate the current minimum surface}
      $x \leftarrow -1$
      for each feature index, $j$, in $open\_list$ do
        if $f[j].value[i] = min\_value[i]$ then
          if $x$ = -1 then
            $x \leftarrow j$
          else {multiple features have minimum value: select features by median and skewness}
            if median($f[j].value$) < median($f[x].value$) then
              if skew($f[j].value$) > skew($f[x].value$) then
                $x \leftarrow j$
              end if
            end if
          end if
        end if
      end for
      $min\_surface \leftarrow$ union($min\_surface, x$)
    end for
    sort $min\_surface$ from smallest to largest median Bhattacharyya value and secondarily skewness
    append $min\_surface$ to end of $sorted\_list$
    $open\_list \leftarrow$ set_difference($open\_list, min\_surface$) {remove features in $min\_surface$ from $open\_list$}
  end while
  return $sorted\_list$ {sorted list of feature indices}
```

2.1. Minimum Surface Bhattacharyya

The mean or class-weighted Bhattacharyya methods for feature selection is lacking in finding the smallest set of features for the multiple-class classification problem. The novel approach treats the Bhattacharyya coefficient for the $C(C-1)/2$ pairs of C classes as a surface and sorts the features by iteratively selecting members consisting of the minimum surface. For example, Fig. 1 depicts the surfaces of three features. The mean of the features are 0.2, 0.5 and 0.6782 for features x, y, and z, respectively. The mean of the three surfaces obfuscates features with the best separability for a given class pair. That is, the mean (or median) Bhattacharyya represents average (or median) performance and has no intelligent way of adapting to local "best performance" in its ordering process. By selecting the features creating the minimum surface (e.g., features x and z in the first iteration), the separability for each class pair is optimized in the feature ordering. Hence, features are greedily selected in a manner that preserves the best separability without searching through all combinations of features. The order of the features utilizing the minimum surface are x, z, and then y. When all features lie on the initial minimum surface, the minimum surface sort becomes a sort of median values.

Algorithm 1 details the pseudocode for implementing the sorting method based on the minimum surface. The MSB method for sorting the features maintains an open list of all features. It selects all features that have a minimum value for a given class pair across the entire combinatorial space of pairings. If there is a tie in the minimum value for a given class pair, the feature with the smallest median Bhattacharyya value and secondarily, the largest skewness is selected. The selected features (composing the minimum surface) may then be sorted in a similar fashion, added to the end of the sorted list, and then removed from the open list. The method is performed iteratively until all of the features have been removed from the open list. The result of the method is an ordered list of features, where a subset may be selected based on a threshold or approach outlined in [6].

By sorting the features based on their minimum Bhattacharyya coefficient, one will guarantee the features that provide the best separability for each class will be ordered first. Assuming the correlation of separation and performance is maintained for various classification problems, this method should result in a smaller subset providing comparable performance to that of the mean or median Bhattacharyya method. Fundamentally, the Bhattacharyya coefficient bases a given feature distribution on a sampling of data, not a model of the data. The method assumes the sampling accurately represents the data population in selecting features. Redundant features will impact the capacity in obtaining the minimal set. Although they will not lie on the same minimum surface, the redundant features will be sorted in successive surfaces. Hence, the method does not have a mechanism for handling highly correlated data. For features that are highly correlated, one of the features may be removed from the open list in advance of the sort to minimize the feature set.

3. RESULTS & CONCLUSIONS

Fig. 2 (a) and (b) illustrates the classification performance of subsets generated by a group of feature selection methods. The classification performance is denoted by the equal weighted accuracy (EWA), which is the average of the accuracies of each class. In general, the embedded and wrapper methods provide the smallest subsets with the best performance. Most of the filter methods provide inferior classification performance for comparable sized subsets with less than a hundred features. The MSB method is the only filter method to provide comparable classification performance with its selected subsets to those of the embedded and wrapper methods. The MSB method operates at a significantly faster runtime than the genetic and best first searches, and GRLVQI.

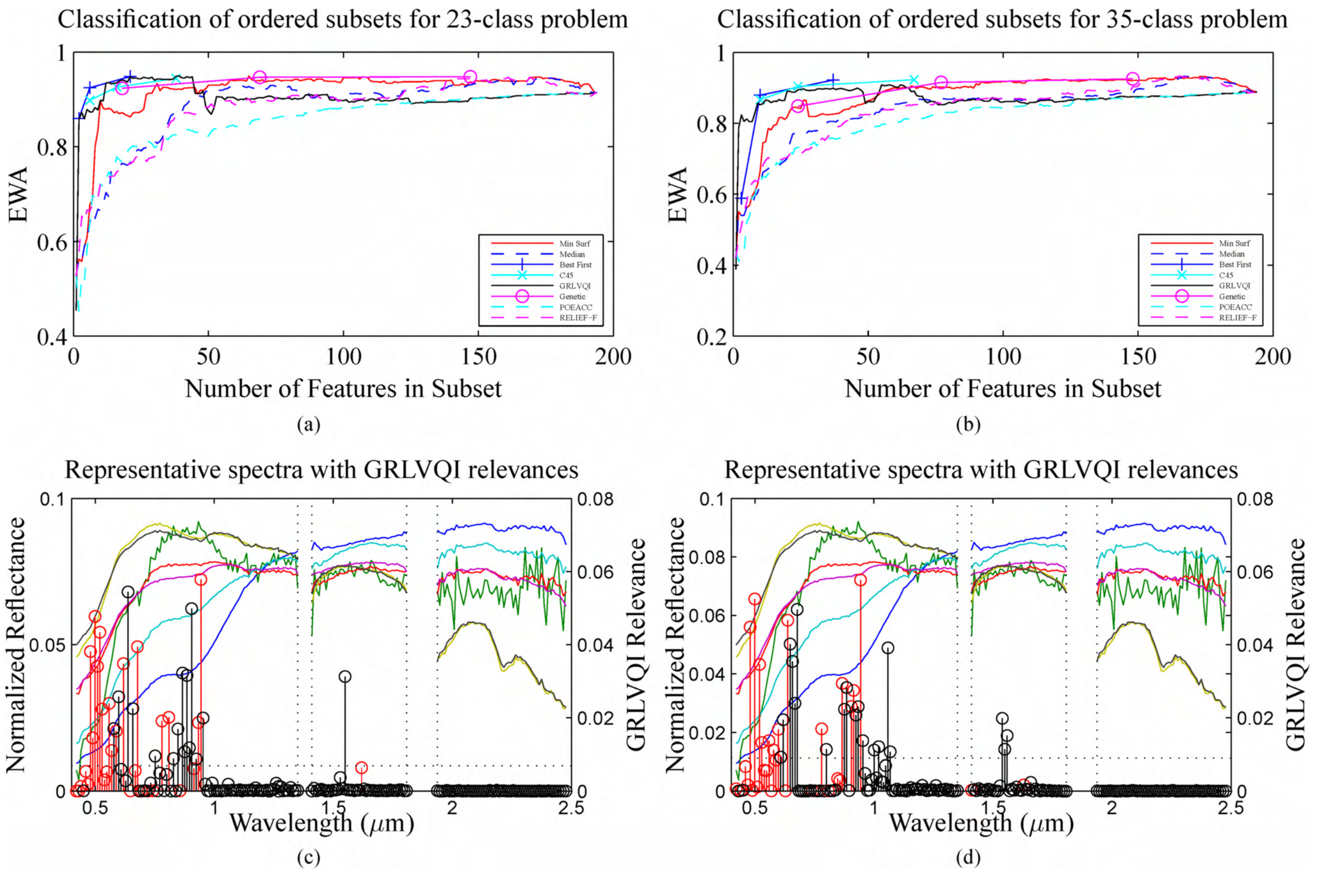
Table 1. Comparison in GRLVQI and MSB Subsets

# of features in subsets	10	20	30	40	50
# of common features (23-class)	4	11	15	26	33
# of common features (35-class)	3	6	13	25	32
MSB subset EWA (23-class)	0.8963	0.8657	0.9239	0.9213	0.9304
GRLVQI subset EWA (23-class)	0.8987	0.9367	0.9409	0.9461	0.8757

An interesting question to answer is whether the selected subsets are consistent among well performing feature selection methods. Surprisingly, two feature selection methods with similar performances of their subsets (MSB and GRLVQI) have strikingly different feature rankings. Table 1 shows the number of common features selected for a given data set (i.e., the size of the intersection of the subsets) among the two methods is relatively small. For the feature subsets composed of 10 features for the 23-class problem, MSB and GRLVQI have 4 features in common so many features are unique to the given feature selection method. Table 1 highlights MSB and GRLVQI subset accuracies for the 23-class problem.

As a maximum margin classifier, GRLVQI ranks features based on maximizing the separation between the classes. MSB utilizes overlap of class-based histograms (Bhattacharyya coefficient) so the margin is not optimized. Even though some features selected by GRLVQI and MSB are distinct, the features may be "close" to each other in wavelength and provide similar information. Fig. 2 (c) and (d) illustrates that many of the highly ranked features are "close" to each other in wavelength. The graph overlays sample spectra with GRLVQI relevances. The stems indicate the GRLVQI relevances and the top 30 ranked wavelengths from MSB are red. The graph indicates the two methods select similar wavelengths. Longer wavelengths may not be preferred due to the noise in reflectances of some materials as illustrated by material G (green). Of note, the MSB feature rankings for this data set suffer since a sufficiently large number of features, 33, reside on the initial minimum surface.

A problem with feature selection method based on rank or weightings such as the MSB is that a useful subset is not intuitively obvious from the rankings. In the case of this paper, useful subsets for the classification task are indicated based on actually performing classification of ordered subsets. Pending the method used, classification may be time intensive. Future work entails selection of a subset from the MSB method without performing feature selection or selecting an arbitrary number of features. One possible approach is



(a)

(b)

(c)

(d)

Fig. 2. Classification performance of subsets generated by a grouping of feature selection methods on hyperspectral data sets: (a) 23 and (b) 35-class problems. Plots (c) and (d) show representative spectra with normalized reflectance values on the left axis: A (blue), G (green), H (red), L (cyan), O (purple), Q (orange), and R (black). The stem plot indicates GRLVQI relevances on the right axis. For GRLVQI, the top 30 ranked features have relevances greater than horizontal dotted line. The red stems denote the top 30 ranked features for MSB. The features for the (c) 23 and (d) 35-class problems are illustrated.

to determine where the median Bhattacharyya coefficient of the multidimensional histograms generated for each ordered subset plateaus.

Disclaimer

The views expressed in this paper are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the United States Government.

4. REFERENCES

[1] M. Dash and H. Liu, "Feature selection for classification," in *Intelligent Data Analysis 1*, 1997, vol. 2, pp. 131–156.

[2] M. Mendenhall and E. Merényi, "Relevance-based feature extraction for hyperspectral images," *IEEE Transactions on Neural Networks*, vol. 19, April 2008.

[3] W. Utschick, P. Nachbar, C. Knobloch, A. Schuler, and J. Nossek, "The evaluation of feature extraction criteria applied to neural network classifiers," in *Proceedings of the Int. Conf. on Doc. Anal. and Rec.*, 1995, pp. 315–318.

[4] N. Thacker, F. Aherne, and P. Rockett, "The Bhattacharyya metric as an absolute similarity measure for frequency coded data," *Kybernetika*, vol. 34, no. 4, pp. 363–368, 1998.

[5] E. Merényi, W. Farrand, J. Taranik, and T. Minor, "Classification of hyperspectral imagery with neural networks: Comparison to conventional tools," *PE&RS*, submitted.

[6] J. González, M. Mendenhall, G. Peterson, and B. Mullins, "Numerical analysis for relevant features in intrusion detection," in *Proceedings of the Recent Advances in Intrusion Detection*, 2009, submitted.