Feature Selection with SACI

Eliana Pantaleão[1], **Luciano Vieira Dutra**[1]

[1]Image Processing Division - National Institute for Space Research (INPE)
Av dos Astronautas, 1758 – 12227-010 – São José dos Campos – SP – Brazil

{elianap,dutra}@dpi.inpe.br

Abstract. *The goal of the feature selection task is to select the most relevant features concerning to a specific classification task, in such a way that the number of features is reduced, but not the discriminative power, with respect to the desired classes. Saci (Sistema de Análise e Classificação de Imagens) is a software for image classification and analysis developed by graduate students at INPE. This work describes the implementation for Saci of one particular multistage feature ranking algorithm using Jeffries-Matusita distance and classification accuracy.*

1. Introduction

Pattern recognition is a research area that aims to associate objects to categories or classes. Each object is represented by a set of measurements, named feature vector or pattern. In some cases, the number of available features is highly superior to the amount necessary to perform the classification task.

But not always more features mean more information. This is due to several factors, like high feature correlation, features that are irrelevant for the desired classification and small sample set. Besides increasing the time spent on the classification task, the additional features can deteriorate the classification result. This problem is called "curse of dimensionality".

The goal of the feature selection task is to select the most relevant features concerning to a specific object classification task, in such a way that the number of features is reduced, but not the discriminative power, with respect to the desired classes. This work describes the implementation for Saci of one particular multistage feature ranking algorithm using Jeffries-Matusita distance and classification accuracy.

2. Feature Selection

Some hyperspectral satellites such as Hyperion [Survey 2007] and some airborne imaging systems like AHI [AHI 2001] and AVIRIS [NASA 2007] have over two hundred data channels, consisting of contiguous narrow spectral bands. Some other applications such as document classification and gene expression analysis may involve thousands of features, due to the very nature of the used models [Saeys et al. 2007]. Even for images with fewer channels, it is usual to extract features from the objects, like texture and shape parameters and spectral operations, resulting in many features available to the classifier.

When two features are highly correlated, they may each bring a lot of information when used separated, but using them together does not add much. An irrelevant feature has no information to add to the target concept. In most cases, the number of features

is translated in the number of classifier parameters, such as synaptic weights in a neural network, or weights in a linear classifier [Theodoridis and Koutroumbas 2006]. Thus, adding irrelevant feature will compromise efficiency without accuracy gain. According to [Dash and Liu 1997], reducing the number of irrelevant/redundant features drastically reduces the running time of a learning algorithm and yields a more general concept.

When the sample set is too small, using a big feature set can lead to overspecialization, making the classifier optimal for identifying the sample vectors, but useless to the real world vectors. According to [Jain et al. 2000], the performance of a classifier is associated with the relationship between sample sizes, number of features and classifier complexity. [Theodoridis and Koutroumbas 2006] remark that a large number of samples is necessary for acceptable performance, and that this number grows exponentially with the dimensionality. If the number of samples remains the same, adding new features can degrade the classifier performance, increasing the error rate. This happens especially with parametric classifiers that estimate the unknown parameters from the samples, and use them as the true parameters in the class-conditional densities. For a fixed sample size, as the number of features is increased (with a corresponding increase in the number of unknown parameters), the reliability of the parameter estimates decreases [Jain et al. 2000].

But not only the number of features is of consequence. If we select poor features, with low discriminative ability, the classifier will not work properly. On the other hand, if we select highly discriminative features, we can get a very satisfactory classifier, with a much simpler design. So, computational complexity can also be reduced with the selection of the most relevant features. Identifying the relevant features can also, in some cases, save time and money with future measurements.

The aim of feature selection is summarized by "given a number of features, how can one select the most important of them so as to reduce their number and at the same time retain as much as possible of their class discriminatory information?" [Theodoridis and Koutroumbas 2006]. In other words, no new feature is created, the ones that are considered irrelevant or redundant are discarded, and we ideally would end up with the best possible feature subset, that is, the one with minimum size and which leads to the minimum classification error rate. In practice, we usually try to select a reduced subset of features that does not significantly decrease the classification accuracy [Dash and Liu 1997].

A feature selection algorithm has to automatically select l features from an original m features set, with $l < m$. To accomplish this goal, it could simply test the subsets of features and try to find the best one of all possible subsets. This is called exhaustive search and consists in measuring class separability or classification accuracy for all possible subsets with l features, and find the best one, i.e., the one with largest separability value (or best accuracy). This is computer expensive and in some cases even the number l of features to be selected is not known. Thus, several values of l would have to be evaluated, in order to choose the best one.

To avoid such a time consuming procedure, other methods were developed based on heuristic or random search in the attempt to reduce computational complexity by, sometimes, compromising performance. Individual ranking algorithms analyze features individually, evaluating their discriminative abilities. This approach has usually very low

computational cost, but the results are not optimal. The best set of l features is not always the set of the best l features. Any separability measure can be used and the basic procedure is its calculation for all individual features and the generation of a descending order. The l features with greater value are selected.

As stated by [Jain et al. 2000], the best would be to choose (or generate) "features that allow pattern vectors belonging to different categories to occupy compact and disjoint regions in the feature space". This means that the effectiveness of the feature set can be evaluated by how well patterns from different classes are separated. To measure how far two classes are when a particular feature subset is considered, several kinds of separability measures can be used. The Jeffries-Matusita distance uses the Bhattacharyya distance B_{ij} [Theodoridis and Koutroumbas 2006] and is given by [Richards 1993]:

$$JM_{ij} = \sqrt{2(1 - e^{-B_{ij}})}$$

The values of the Jeffries-Matusita distance are between 0 and $\sqrt{2}$, and the function has a saturating behavior when class separability increases (asymptotic limit is $\sqrt{2}$). It tends to suppress high separability values, while overemphasizing low separability values [Kavzoglu and Mather 2002]. Sometimes, for implementation purposes, the JM^2 is used instead.

If Gaussian distributions are considered, the average JM distance can be written as:

$$JM = \frac{2}{N(N-1)} \sum_{i=1}^{N} \sum_{j=1}^{i-1} JM_{ij}$$

3. Saci

Saci (*Sistema de Análise e Classificação de Imagens*) is an image classification and analysis software that has been developed since 2003 by graduate students that attend the course "Pattern Recognition". As part of the evaluation process of the course, the students had to implement one image algorithm, that was then added to the Saci project, previously called SCID. So far, Saci has tools to open images in several formats, a ROI (region of interest) selection tool, several methods for feature selection and extraction, k-means unsupervised classification and some supervised statistic and deterministic classification methods. For all classification methods, a confusion matrix is generated, and some classification evaluation metrics such as kappa and tau are shown.

The software was entirely developed using IDL (Interactive Data Language) [ITT 2008] and is compatible with Envi (Environment for Visualizing Images) IDL, in the sense that it can read Envi formats for images and ROIs. The system is continually being updated, but it is not available for public download yet. Since each part was developed by a different student, and in a short period of time, usually one trimester, the system has some problems. For example, some of the algorithms will not work unless the image is square. The *Quilombo* group is working to solve these problems and a consistent version is expected for anytime this year.

4. Implementation

The implemented method is based on a method detailed in [Huber and Dutra 1998] and is a multistage search in the space of all possible feature subsets. On the first stage, the JM

distance is used to evaluate and rank the features. Then, best ranked features are combined and evaluated by a wrapper method, using SVM as the classifier.

When a separability measure like JM distance is used, results can only be compared for sets with the same number of features. This is due to the monotonicity of this kind of measure. If more features are added, the value of JM distance always increase, although classification might have a worse result.

In this method, $nsets$ subsets of the original feature set are generated, containing a number $subsetsize$ of features. Then, for each set, a rank is generated, using function $rank_i$:

$rank_i = \frac{1}{M/2} \sum_{j=1}^{M/2} \frac{h_{ij}}{j}$ where $M = 2^m$

$h_{ij} \sum_{k=1}^{j} f_{ik}$, where f_{ik} is 1 if feature i is in ranked subset k, and 0 otherwise.

The best s_i features from each subset are selected, and compose a new subset F_s, with $s = \sum s_i$ features. Then, the best l features are selected from F_s using classification accuracy.

The choice of the parameters is left to the user, and choosing $s = m$ will make the algorithm skip the JM phase.

5. Results and conclusions

The implementation is not fully operative yet, because of memory allocation problems. Initial attempts to process a Hyperion image with 204 channels were impossible to fulfill. Then, texture features were extracted from a standard RGB image, resulting in 42 channels. Yet, it was not possible to apply the method. For smaller examples, with only 3 channels, the results were the same for this method and the others available in Saci, such as Exhaustive Search, SBS and SFS.

6. Acknowledgements

Eliana Pantaleão thanks CAPES for financial support.

References

AHI (2001). Airborne hyperspectral imager. http://www.higp.hawaii.edu/ahi/.

Dash, M. and Liu, H. (1997). Feature selection for classification. *Intelligent Data Analysis*, 1(3):131–156.

Huber, R. and Dutra, L. V. (1998). Feature selection for ers-1/2 insar classification: High dimensionality case. pages 1605–1607. IEEE.

ITT (2008). Visual information solutions. http://www.ittvis.com/.

Jain, A. K., Duin, R. P. W., and Mao, J. (2000). Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):4–37.

Kavzoglu, T. and Mather, P. M. (2002). The role of feature selection in artificial neural network applications. *International Journal of Remote Sensing*, 23(15):2919–2937.

NASA (2007). Aviris - airborne visible/infrared imaging spectometer. http://aviris.jpl.nasa.gov/.

Richards, J. A. (1993). *Remote Sensing Digital Image Analysis - An Introduction*. 2 edition.

Saeys, Y., Inza, I., and Larranaga, P. (2007). A review of feature selection techniques in bioinformatics. *Bioinformatics*, 23(19):2507–2517.

Survey, U. S. G. (2007). Eo-1 user's guide. http://eo1.usgs.gov/.

Theodoridis, S. and Koutroumbas, K. (2006). *Pattern Recognition*. Academic Press, San Diego, 3 edition.