Feature Selection with Non-Parametric Mutual Information for Adaboost Learning

Xavier Baró [a] & Jordi Vitrià [a,b]
[a] *Centre de Visió per Computador*
& [b] *Dept. de Ciències de la Computació UAB*
Edifici O - Campus UAB, 08193 Bellaterra, Barcelona, Catalonia, Spain
{xbaro,jordi}@cvc.uab.es

Abstract. This paper describes a feature selection method based on the quadratic mutual information. We describe the needed formulation to estimate the mutual information from the data. This paper is motivated for the high time cost of the training process using the classical boosting algorithms. This method allows to reuse part of the training time used in the first training process to speed up posterior training to update the detectors in front of samples changes.

Keywords. Boosting, feature extraction, feature selection, mutual information, density estimation

1. Introduction

Feature selection methods are techniques to select a reduced subset of features from a normally very large set of features in order to solve a classification problem. This procedure can reduce not only the cost of classification by reducing the number of features, but in some cases it can also provide a better classification accuracy[1].One of the most popular algorithms used on feature selection for classification is the Adaboost, a boosting algorithm introduced by Freund & Schapire in [2]. Boosting is a powerful learning concept that allows combining the performance of many simple classification functions to produce a strong classifier.
The weak classification functions, named weak classifiers, in our case are simply features with a threshold. The Adaboost algorithm associates a weight to each sample depending on the difficulty to classify it. In each iteration, the algorithm selects the weak classifier h_t which minimizes the classification error over the samples weights distribution. Then it updates the weights distribution and calculates a weight for the added weak classifier. After T rounds of boosting, the decision of the ensemble is defined as:

$$H(x) = \begin{cases} 1 & \sum_{t=1}^{T} \alpha_t h_t(x) \geq \theta \\ 0 & otherwise \end{cases} \quad (1)$$

where the α_t are the standard Adaboost ensemble weights and θ is the threshold of the ensemble.

Although this approach can be applied to a wide variety of fields, we will use it in the object detection problem, to be exact in the rare event detection. In [3], Viola & Jones presents a boosted cascade of simple features for face detection, the canonical example of rare event detection problem. They use a simple set of rectangular features, and a fast method to calculate them based on the concept of integral image. The schema proposed by Viola & Jones is especially useful in rare event detection, where we need to discard thousands or millions of images and recognize just a few. In [4] Lienhart and Maydt extends the features set used by Viola and Jones, introducing new types of features and 45^o rotated features. To evaluate this last type of features they introduce the 45^o rotated integral image.
Nowadays, this combination of extended Haar-features and a boosted cascade is widely used in rare event detection. The main problem of this method is the training time. With a training window size of 30×30 we can have more than 700.000 features, and training sets with thousands of images. In our tests applied to traffic signs[5], a set of 1000 positive examples and a cascade goal false alarm ratio of 0.00001 spends a week to train. After the training stage, the cascade of 17 stages is formed by only 323 simple classifiers.
We think that this difference between the number of available and selected features is not a coincidence, and it means that most part of the features don't help on the classification problem. In this direction we bet for the mutual information between features and classes to select a priori a small set of features to solve the classification problem reducing the training time.

2. Mutual Information

We basically will follow the work of Torkkola in [6]. In this work he proposes a transformation instead of a features selection, but makes a very detailed study of all the problems and the methods to calculate the mutual information. We will assume that each feature X is an univariate random variable and C a discrete-valued random variable representing the class labels. In following equations, uppercase P will denote a probability and lowercase p a probability density. Given a sample, the entropy or uncertainty of the class label, making use of Shannon's definition, can be expressed in terms of class prior probabilities.

$$H(C) = -\sum_c P(c) log(P(c)) \tag{2}$$

Once we have observed a feature value x, the uncertainty of the class label is expressed as:

$$H(C|X) = -\int_x p(x) \left(\sum_c p(c|x) log(p(c|x)) \right) dx \tag{3}$$

The amount by which the class uncertainty is reduced, after having observed the feature vector x, is the mutual information, which can be written as:

$$I(C, X) = \sum_c \int_x p(c, x) log \frac{p(c, x)}{P(c)p(x)} dx \tag{4}$$

The practical estimation of the mutual information from data based on expression (4) is difficult because the good estimation of the probability density function of a continuous variable is not easy. To solve this problem, in the following sections we describe a method to calculate a mutual information measure based on a reformulation of the entropy concept and a density estimator method. The result is a formulation of mutual information in terms of discrete sums.

3. Parzen density estimator

The Parzen window method [7] is a non-parametric method to estimate the probability density function. This method involves placing a kernel function on top of each sample and evaluate the density as a sum of kernels. The Gaussian kernel is defined as:

$$G(\mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-\mu^2}{2\sigma^2}} \tag{5}$$

Now, for two kernels, we can write:

$$\int_x G(x-\mu_1, \sigma_1^2) G(x-\mu_2, \sigma_2^2) = G(\mu_1 - \mu_2, \sigma_1^2 + \sigma_2^2) \tag{6}$$

Thus, the convolution of two Gaussians centered at μ_1 and μ_2 is a Gaussian centered at $\mu_1 - \mu_2$, with a variance equal to the sum of variances. Assume now that the density of X is estimated as a sum of Gaussians centered at a sample x_i. This is the Parzen density estimation:

$$p(x) = \frac{1}{N} \sum_{i=1}^{N} G(x - x_i, \sigma^2) \tag{7}$$

where N is the number of samples.

4. Renyi's Entropy

Renyi's entropy is a more general formulation than Shannon entropy. In the general theory of means [8], the mean of the real numbers $x_1, ..., x_N$ with positive weighting (not necessarily probabilities) $p_1, ..., p_N$ has the form:

$$\bar{x} = \varphi^{-1} \left(\sum_{k=1}^{N} p_k \varphi(x_k) \right) \tag{8}$$

where $\varphi(x)$ is a Kolmogorov-Nagumo function, which is an arbitrary continuous and strictly monotonic function defined on the real numbers. In general, an entropy measure H obeys the relation:

$$H = \varphi^{-1} \left(\sum_{k=1}^{N} p_k \varphi(I(p_k)) \right) \tag{9}$$

where $I(p_k) = -log(p_k)$ is Hartley's information measure [9]. In order to be an information measure, $\varphi(.)$ can not be arbitrary since information is "additive". To meet additivity condition, $\varphi(.)$ can be either $\varphi(x) = x$ or $\varphi(x) = 2^{(1-\alpha)x}$. If $\varphi(x) = x$ is selected, (9) will become Shannon's entropy. For $\varphi(x) = 2^{(1-\alpha)x}$ Renyi's entropy of order α is obtained [10], which we will denote by H_{R_α}

$$H_{R_\alpha} = \frac{1}{1-\alpha} log \left(\sum_{k=1}^{N} p_k^\alpha \right) \qquad \alpha > 0, \alpha \neq 1 \tag{10}$$

In fact, Renyi's entropy of order α will compute interactions among α-tuples of samples, providing even more information about the complex structure of the data set[13]. When $\alpha = 2$, (10) is called quadratic entropy due to the quadratic form on the probability. For a discrete variable C and a continuous variable X, the quadratic Renyi entropy H_{R_2} is defined as [8]:

$$H_{R_2}(C) = -log \sum_c p(c)^2 \qquad H_{R_2}(X) = -log \int_x p(x)^2 dx \tag{11}$$

Note that Renyi's quadratic entropy involves the use of the square of the PDF. An important observation is that this alternate definition of entropy is equivalent to Shannon's entropy for the goal of entropy maximization [11]. Then, it follows that the quadratic Renyi's entropy in (11) equals [6]

$$\begin{array}{l} H_{R_2}(X) = -log \int_x p(x)^2 dx \\ = -log \frac{1}{N^2} \int_x \left(\sum_{k=1}^{N} \sum_{j=1}^{N} G(x - x_k, \sigma^2) G(x - x_j, \sigma^2) \right) dx \\ = -log \frac{1}{N^2} \sum_{k=1}^{N} \sum_{j=1}^{N} G(x_k - x_j, 2\sigma^2) \end{array} \tag{12}$$

Thus, Renyi quadratic entropy can be estimated as a sum of local interactions, as defined by the kernel, over all pairs of samples.

5. Information potentials

Assume that we have J_p samples for each class c_p. Then, the class prior probabilities are $P(c_p) = J_p/N$, with $\sum_{p=1}^{N_c} Jp = N$. Now we will use different notations for the samples of data X. A sample is written with a single subscript x_i when its class is irrelevant. If the class is relevant, we will write x_{pj}, where p is the class index and j the within-class index. N_c is the number of classes.
The density of each class c_p, as a Parzen estimate using the Gaussian kernel of width σ, is written as:

$$p(x|c_p) = \frac{1}{J_p} \sum_{j=1}^{J_p} G(x - x_{pj}, \sigma^2) \tag{13}$$

Using the definition of joint density $p(c, x) = p(x|c)P(c)$, we have

$$p(c_p, x) = \frac{1}{N} \sum_{j=1}^{J_p} G(x - x_{pj}, \sigma^2), p = 1, ..., N_c \tag{14}$$

Finally, using that the density of all data is $p(x) = \sum_c p(c, x)$, we can write

$$p(x) = \frac{1}{N} \sum_{p=1}^{N_c} \sum_{j=1}^{J_p} G(x - x_{pj}, \sigma^2) = \frac{1}{N} \sum_{i=1}^{N} G(x - x_i, \sigma^2) \tag{15}$$

Using the quadratic entropy in the calculus of mutual information, we can speak of quadratic mutual information, denoted by I_T. With continuous-valued X and discrete C, the definition of the quadratic mutual information can be written as [13]:

$$\begin{aligned} & I_T(C, X) = V_{IN} + V_{ALL} - 2V_{BTW} \\ & where \quad \begin{cases} V_{IN} \equiv \sum_c \int_x p(c, x)^2 dx \\ V_{ALL} \equiv \sum_c \int_x P(c)^2 p(x)^2 dx \\ V_{BTW} \equiv \sum_c \int_x p(c, x) P(c) p(x) dx \end{cases} \end{aligned} \tag{16}$$

Using a set of samples $\{x_i\}$, combining the equations (14),(15) and (16), and making use of (6) and (12), we get:

$$V_{IN}(\{c_i, x_i\}) = \sum_c \int_x p(c, x)^2 dx = \frac{1}{N^2} \sum_{p=1}^{N_c} \sum_{k=1}^{J_p} \sum_{l=1}^{J_p} G(x_{pk} - x_{pl}, 2\sigma^2) \tag{17}$$

$$V_{ALL}(\{c_i, x_i\}) = \sum_c \int_x P(c)^2 p(x)^2 dx = \frac{1}{N^2} \left(\sum_{p=1}^{N_c} \left(\frac{J_p}{N} \right)^2 \right) \sum_{k=1}^{N} \sum_{l=1}^{N} G(x_k - x_l, 2\sigma^2) \tag{18}$$

$$V_{BTW}(\{c_i, x_i\}) = \sum_c \int_x p(c, x) P(c) p(x) dx = \frac{1}{N^2} \sum_{p=1}^{N_c} \frac{J_p}{N} \sum_{j=1}^{J_p} \sum_{k=1}^{N} G(x_{pj} - x_k, 2\sigma^2) \tag{19}$$

These kinds of quantities can be called "information potentials" in analogy to physical particles [13]. In the next section we present a method to calculate the value of the σ used in these equations.

6. Sigma estimation

The correct selection of the sigma value has a capital importance for the correctness of the final mutual information values. This parameter depends on the data, and can be viewed as the window width in the Parzen method. In Fig. 1 is showed the effect of the variation of this value in the estimated probability density function. In [12], Silveman develops a set of equations in order to select a correct value in the case of Gaussian kernel, minimizing the mean integrated square error. The resultant equation is:

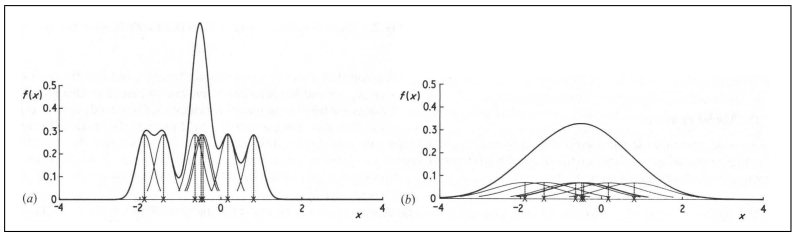

Figure 1. From [12], Kernel estimation showing individual kernels. Windows widths: (a) 0.2;(b)0.8.

$$\sigma = 0.9An^{-\frac{1}{5}} \tag{20}$$

where $A = min(standard\ deviation, interquartile\ range/1.34)$ and n is the number of samples used for the estimation.

7. Results

First we briefly presents some statistics over the detection step to justify the necessity to use some methods to reuse the time spent training the system. All this results are extracted from a traffic sign recognition system based on Adaboost and Haar-like features[5]. We divided the traffic signs in 5 classes (yield, danger, prohibition, command and kilometric points). For each class we trained a different detector, which is a cascade of detectors trained using Adaboost. In table 1 we show the mean number of features per stage, and one can see that is a number really smaller than the original 700.000 features of the features set. The results obtained when we apply our detectors to the test are quite low due to the difference in the orientation, illumination and/or kind of the signs. We want to add the failed signs to the train set to improve the detectors. Now it means to spend a week for each detector, and here is where we want to introduce the feature selection methods, that can reduce drastically this time. To measure the influence of the

Table 1. Training set size and performance of each detector after analyzing 9510 frames of 1020×1024. All the detectors are trained on a size of 30×30 except the kilometric points trained at 24×24. HR is the obtained hit ratio and FA the false alarm ratio.

Sign type	# Training Samples	Mean features/stage	# Signs	HR	FA
Yield	425	9.176	179	93.08%	1060/179=5.92
Danger	545	12.125	385	89.59%	854/385=2.21
Prohibition	993	19	481	83.36%	371/481=0.77
Command	356	11.667	115	70.88%	1382/115=12.01
Km points	218	8	148	76.99%	2928/148=19.78

mutual information to the boosting process, we compare the convergence speed of the Adaboost selecting small sets of features. The following tests are programmed in Matlab, using a sampled features set of 8000 rectangular features and a set of 400 samples (50% positive and 50% negative). We use the Discrete Adaboost algorithm, fixing the number of iterations at 100.

Using all the features the training process spend more than 4 hours and half, and it converge at iteration 13. If we select only the 100 features with the higher mutual information, the training time is reduced to only 5 minutes, and it converges at iteration 39.

Using only the 100 worst features, and maintaining the training time, the convergence of the Adaboost is delayed to the iteration 83 (see Fig. 2). It is important to emphasis that at the end of the training process, in all cases the detector obtain the same detection rates. Finally, we compare the effect of the number of features over the convergence speed.

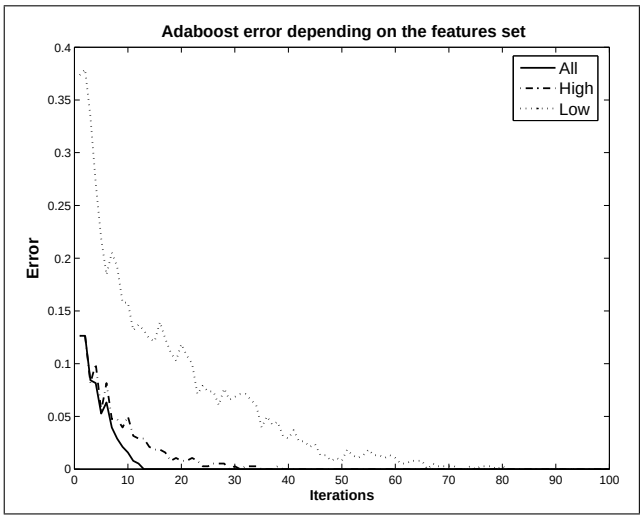

Figure 2. Convergence of the Adaboost depending on the features set. First using all features, then only the 100 with the higher MI and finally the 100 with lower MI.

Using the same training set as before, we train a classifier using the N features with better mutual information. The results are in Fig. 3. We can see that from a certain number of features, to add more features has a moderate effect over the convergence speed. Our interpretation is that from a certain number of features, the new features do not apport important information for the classification process.

8. Conclusions and future work

The convergence speed shows that the mutual information between features and class labels have a direct relationship with the convergence speed of the Adaboost.
To calculate the mutual information between each feature and the class labels is too expensive in time to be calculated each time. The main idea is use the first samples set to calculate the mutual information and select a small subset of features. Then each time that we add new samples to our training set, we will repeat the training process only with this selected features, and it will reduce drastically the time used to maintain all detectors up to date.
In this first approach, we only select the features with higher mutual information, but is logical to use the mutual information to select also the features with the minimal mutual information with the other features, to eliminate the redundant features.

Acknowledgements

This work has been partially supported by MCYT grant TIC2003-00654, Spain. This has been developed in a project in collaboration with the "Institut Cartogràfic de Catalunya" under the supervision of Maria Pla.

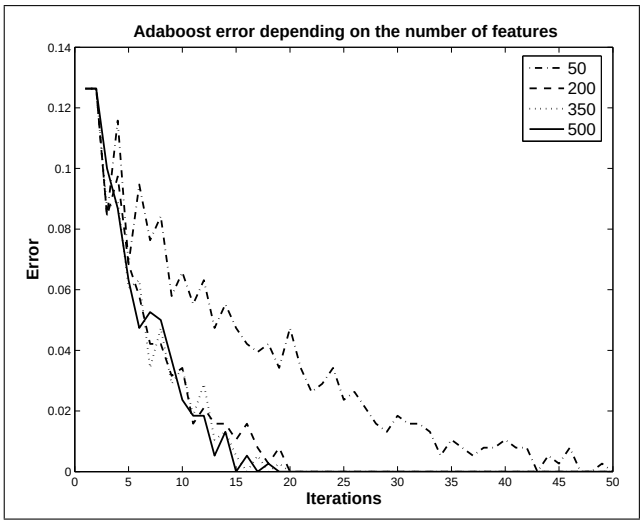

Figure 3. Convergence of the Adaboost depending on the number of features. Features are sorted by their mutual information value and the features set are the N features with the higher MI.

References

[1] A. K. Jain and B. Chandrasekaran. "Dimensionality and sample size considerations". In P. R. Krishnaiah and L. N. Kanal, editors, Pattern Recognition Practice, volume 2, chapter 39, pages 835-855. North-Holland, 1982

[2] Freund, Y. & R.E. Schapire ."Experiments with a New Boosting Algorithm". Proceedings of the Thirteenth International Conference on Machine Learning, pp. 148-156, Morgan Kaufmann. 1996.

[3] P. Viola & M. Jones. "Rapid object detection using a boosted cascade of simple features". In Proc. CVPR, pages 511-518, 2001.

[4] R. Lienhart & J. Maydt. "An Extended Set of Haar-like Features for Rapid Object Detection". Proceedings of the International Conference on Image Processing, Rochester, USA, IEEE, Piscataway, 2002, 900-903.

[5] X.Baró, J.Vitrià. "Fast Traffic Sign Detection on greyscale images". Catalan Conference on Artificial Intelligence, 2004. In press Recent Advances in Artificial Intelligence Research and Development, IOS Press, Amsterdam, 2004.

[6] K. Torkkola and W. Campbell, "Mutual Information in Learning Feature Transformations", Proceedings of ICML 2000, Stanford, CA, June 29-July 2, 2000.

[7] E. Parzen, "On the estimation of probability density function and the mode", The Annals of Mathematical Statistics, 33:1065, 1962.

[8] A. Renyi. "On mesures of entropy and information". In Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, pages 547-561. Universtiy of California Press, 1961

[9] R.V. Hartley "Transmission of information". Bell System Technical Journal, 7, 1928.

[10] A. Renyi. "Some Fundamental Questions of Information Theory". Selected Papers of Alfred Renyi, Vol 2, pp 526-552, Akademia Kiado, Budapest, 1976.

[11] J.N. Kapur. "Measures of Information and Their Applications", John Wiley & Sons, 1994.

[12] B. W. Silverman, "Density Estimation for Statistics and Data Analysis", Chapman and Hall, London, 1986.

[13] J.C. Principe, J.W. Fisher and D.Xu. "Information theoretic learning". In Simon Kaykin, editor, Unsupervised Adaptive Filtering. Wiley, New York, NY, 2000.