Content identification: machine learning meets coding

Sohrab Ferdowsi Svyatoslav Voloshynovskiy
University of Geneva
Dept. of Computer Science, SIP Group
Battle Bât. A, 7 route de Drize, 1227 Carouge, Switzerland
Sohrab.Ferdowsi@unige.ch svolos@unige.ch

Abstract

We address the content-identification problem by modeling it as a multi-class classification problem. The goal is to pave the way and establish a general framework to incorporate the powerful algorithms of the machine learning literature in learning from data into this problem. Through this end, a particular successful approach, linked with the coding theory known as ECOC is considered and studied. We argue that the conventional codings used in this approach are suboptimal by analyzing the problem from an information-theoretic viewpoint. We then advise the use of our recently proposed method for this problem. The ECOC approach converts the multi-class problem to several binary problems. We consider these equivalent binary classification tasks in more details and use the Gaussian Mixture Models instead of SVM's. This latter brings significant reduction in complexity by having an assumption on the distributions.

1 Introduction

Content identification is an active field of research where different methods of robust hashing a.k.a. content fingerprinting, hypothesis testing and computer vision are involved. Conventional approaches towards content identification assume very strict distributions for the data, e.g. they assume that the items in the database and queries corresponding to them both follow a Gaussian distribution. In practice, however, due to different acquisition conditions like the inevitable rotations and posture differences in each acquisition, these assumptions turn out to be oversimplifying. Thus, to cope with these situations, alternative approaches should be utilized.

The power behind machine learning algorithms in learning from data and their generalization capabilities guaranteed by the rigorous statistical learning theory justifies the use of these techniques in content identification applications. Through this end, the problem of content identification was proposed to be considered as a multi-class classification where different acquisitions of each item in the database are regarded as training instances and the query items are the test data in the problem of classification.

In spite of extensive research in the field, multi-class classification is still considered as an open issue in the domain of Machine Learning. One way to address this problem is the so called Error Correcting Output Codes (ECOC) approach that converts a multi-class problem into a set of binary classification problems [1].

In the ECOC approach, the learning problem is considered as a hypothetical communication channel and the imperfections or ambiguity in the learning process are considered as an equivalent channel noise. Therefore, it was argued that, in order to combat noise, one should encode the data before sending them to the channel and thus, using the channel codes. In some recent studies, the authors attempt at straightforward usage of advanced error correction codes like LDPC to achieve the accurate classification [2].

The attractiveness of the ECOC approach comes from the competitive number of binary classifiers needed in comparison to the one-vs-all and one-vs-one strategies. In the limiting case, the number l of binary classifiers needed in the ECOC case or equivalently the length of the channel code is of $\mathcal{O}(log_2(M))$, where M is the number of classes. In this sense, the classifiers can be considered as a binary tree that represents the lowest complexity for M-class classification. Using communication setup terminology, it means that the decoding of *random codewords* of length N in the codebook $M \sim 2^{NC}$can be accomplished in $\mathcal{O}(NC)$ checks, versus the typical brute force decoding which is of order $\mathcal{O}(2^{NC})$, where C denotes the channel capacity.

In a previous work [3], we invesitigated the learning problem based on the ECOC used as a method of content identification. In this information-theoretic viewpoint which is linked with the coding theory, a very important issue is the choice of the coding matrix. There we proposed a novel coding matrix to be used as an alternative to the existing algorithms based on random codes and error correction codewords.

In this work, we further consider the coding approach towards multi-class classification. We analyze the information-theoretic setup for the learning problem in section 2 and argue that due to several factors, including the statistics of channel noise, conventional channel codes cannot be used efficiently and do not have their expected error correcting capability. We then review a proposed coding matrix design which was indicated to have optimal performance. The choice of the underlying binary classifiers used is investigated in section 3. In order to reduce the complexity of binary classifications, instead of the Support Vector Machines (SVM's) with nonlinear kernels, we classify the training data by fitting a Gaussian Mixture Model (GMM) to each class. We finally conclude the paper in section 4.

2 Multi-class Classification: A Coding Based Approach

In this section we consider the problem of multi-class classification and the coding approach to solve it. The general problem formulation is presented in section 2.1 and the coding approach is presented in section 2.2. We analyze the assumptions of this model in section 2.3.

2.1 Problem Formulation

A set of training instances are given which contain features $\mathbf{x}(i) \in \mathbb{R}^N$ and their labels $g(\mathbf{x}(i))$'s belonging to the M classes. These labels are assumed to be generated through an unknown hypothetical mapping function $g : \mathbb{R}^N \longmapsto \{1, 2, ..., M\}$ that we want to learn and approximate based on the given labeled training examples and using preselected approximation functions. The labels, unlike the common cases in machine learning, belong to a set of $M \geq 2$ members rather than only two classes. Because most of the existing classification algorithms are naturally designed for $M = 2$ classes, to generalize them for multi-class cases usually requires to consider the problem as several binary classification problems.

2.2 ECOC Approach Towards Multi-class Classification

The main idea behind the ECOC approach to solve the multi-class problem is to consider it as a communication system where the identity of the correct output class for a given unlabeled example is being transmitted over a hypothetical channel which, due to the imperfections of the training data, the errors in the learning process and non-ideal choice of features is considered to be noisy [1]. Therefore, it is reasonable to encode the classes using error correcting codes and transmit each of the bits through the channel, i.e., to run the learning algorithm, so that we would be able to cope with the errors in each individual binary classifier. Figure 1 illustrates this idea.

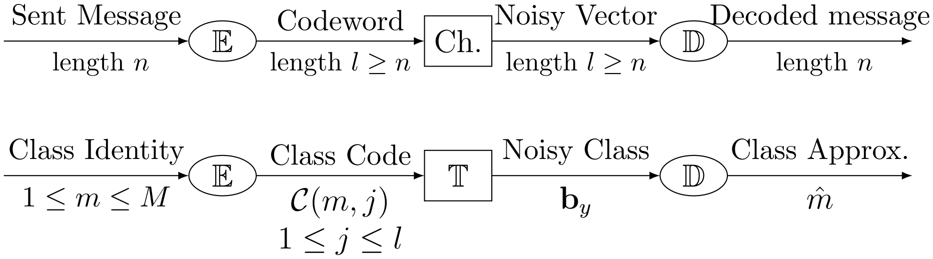


Figure 1: Communication system and its classification equivalent: $\mathbb{E}$ and $\mathbb{D}$ are the Encoding and Decoding stages in communication, respectively. $\mathbb{T}$ is the training procedure, $\mathcal{C}$ is the coding matrix, $\mathbf{b}_y$ is the derived binary code for the test example.

Concretely, we assign randomly to each of the M classes a row of a coding matrix $\mathcal{C}_{(M \times l)}$. Then we run a binary learning algorithm on all the training samples for every column of $\mathcal{C}$ so that we will have l mappings from $\mathbb{R}^N$, the original data space to the one dimensional binary space $\{0, 1\}$, or equivalently, one mapping rule from $\mathbb{R}^N$ to the l-dimensional binary space $\{0, 1\}^l$. It is impotent to note that the l learning procedures are essentially binary and symmetric, i.e., the number of instances in each binary class is approximately the same.

Given a new unlabeled example $\mathbf{y}$, we map it from $\mathbb{R}^N$ to the l-dimensional binary space through the same mapping rule learned as above. We then compare this binary representation $\mathbf{b}_y$ with the items in the database, or equivalently the rows of $\mathcal{C}$ by minimum Hamming distance rule or any other relevant decoding method.

2.3 Experimental Assessment of the Channel Code Model

Although we use the ECOC approach to address the multi-class classification problem, in this section, however, we question this model by arguing that the mismatch in the real and assumed statistics of the channel noise can not guarantee the expected error correcting capability of channel codes.

In particular, we experimentally demonstrate that the samples of learning noise are highly correlated. This is not in accordance with the assumptions behind the binary symmetric channel model used in the design of the common channel codes like the BCH, LDPC or Turbo codes.

In this part we measure the correlation properties of the channel noise of the multi-class learning problem through a simple synthetic simulation. We use a synthesized dataset of M classes of Gaussian data.

The centers of classes $\mathbf{x}_c(i) \in \mathbb{R}^N$, with $1 \leq i \leq M$, are generated as realizations $\mathbf{X}_c \sim \mathcal{N}(\mathbf{0}, \sigma^2_{inter}\mathbb{I}_N)$. The j^{th} instance of i^{th} class is generated as $\mathbf{x}_j(i) = \mathbf{x}_c(i) + \mathbf{Z}_{intra}$ with $\mathbf{Z}_{intra} \sim \mathcal{N}(\mathbf{0}, \sigma^2_{intra}\mathbb{I}_N)$. The test data are generated as Additive White Gaussian Noise with covariance matrix $\sigma_z^2\mathbb{I}_N$ added to the centres of each class where σ_Z^2 is controlled by SNR.

As was explained in 2.2, the rows of the binary coding matrix denoted as $\mathbf{b}_X(k)$ are fed to the learning channel to produce the $\mathbf{b}_Y$'s. Thus, we consider the k^{th} sample of the learning channel noise as

$$\mathbf{b}_Z(k) = \mathbf{b}_X(k) \oplus \mathbf{b}_Y(k),$$

with $1 \leq k \leq l$.

For each of the M classes, we generated 100 test data for different values of SNR. The estimator of the autocorrelation of each noise string is considered as:

$$\hat{R}_Z(j) = \begin{cases} \frac{1}{l}\sum_{i=1}^{l-j} \mathbf{b}_Z(i+j)\mathbf{b}_Z(i) & j \geq 0 \\ \hat{R}_Z(-j) & j < 0 \end{cases}$$

with $-l \leq j \leq l$. We then average the $\hat{R}_Z(j)$'s for all the noise realizations.

Figure 2 illustrates the resulting autocorrelation sequences of channel noises. We had $M = 100$ classes and used randomly generated binary values as the elements of the coding matrix $\mathcal{C}$. There were 10 instances in each class and 100 testing data per class. The results are compared with the channel noise for the fingerprinting approach in content identification where it is considered that the equivalent Binary Symmetric Channel of identification produces *i.i.d.* noise. In order to make a meaningful comparison, we evaluated the autocorrelations for each method for results that correspond to the same accuracy for the learning method and probability of correct identification for the identification method ($SNR = -2.14db$ for learning and $SNR = 2.94db$ for fingerprinting). For the fingerprinting method, we considered the center of each class corresponding to the representative of each item.

According to Figure 2, while in this experimental setup, the channel noise for fingerprinting remains *i.i.d.*, the equivalent channel noise for learning problem shows high amounts of correlation in the autocorrelation sequences and thus, far from the *i.i.d.* assumption.

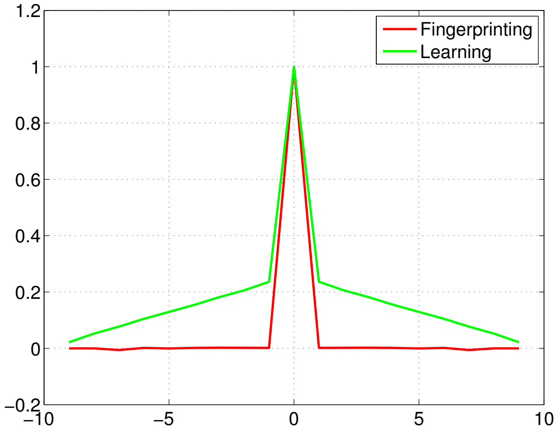
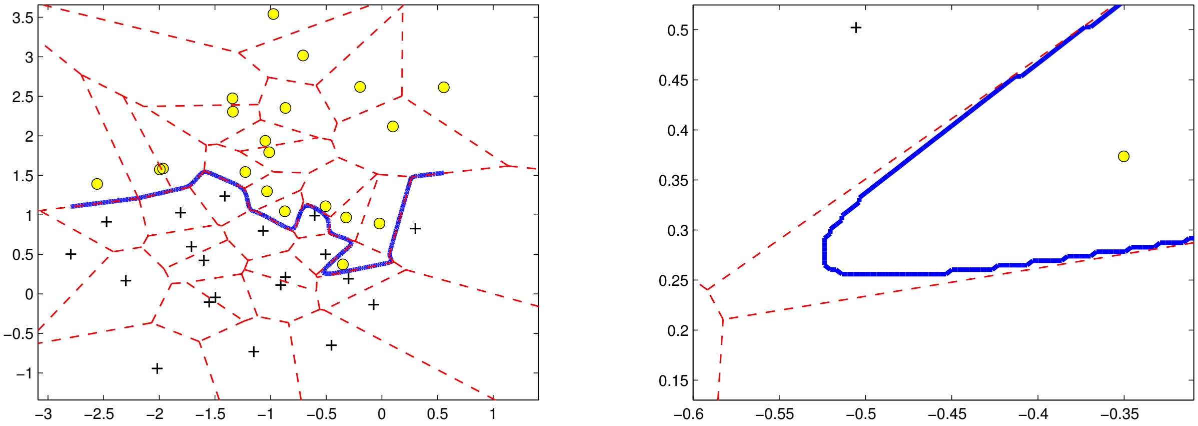


Figure 2: Autocorrelation of noise, learning channel vs. identification channel

2.4 A Novel Approach Towards Coding Matrix Design

When used as the method of choice for binary classification, Support Vector Machine (SVM) with properly tuned kernels draws a hyperplane in a space of higher dimension than the original data space of $\mathbb{R}^N$ where the transformed data are linearly separable, such that this hyperplane has the maximum margin from the boundary training instances of each class (support vectors). As shown in Figure 3, the hyperplane drawn in the higher dimensional space, when considered in the original domain is reflected as complex decision boundaries approximating the Voronoi cells of the support vectors.

Assuming the use of kerneled SVM as the base binary classifier, the design of a good coding matrix should imply efficient learning of the Voronoi cells of the support



(a) decision boundary in the original data space

(b) zoomed

Figure 3: SVM boundaries (solid lines) following the Voronoi cells (dashed lines) of the support vectors.

vectors which means learning at least once these cells and at the same time avoiding redundant learning of them. The first requirement links with having good performance via optimal class separability and the latter implies maintaining the number of columns (classifiers) as small as possible, i.e. $l = \lceil \log_2 M \rceil$ or equivalently, having maximal rate of communication. This optimal design, in fact happens when the rows of $\mathcal{C}$ are chosen as the modulo-2 equivalent of the the class numbers. Therefore, as an example, in a 20-class problem, the codewords will be of length $l = \lceil \log_2 M \rceil = 5$, the 1^{st} class should be encoded as the binary string $'00000'$, and the 6^{th} class should be encoded as $'00101'$.

In fact learning the Voronoi regions of the support vectors is also equivalent to the maximum likelihood rule under Gaussian assumption for the classes. While ML decoding rule requires the knowledge of Voronoi regions for all classes, in our approach, these regions are distributed among the l binary classifiers and are learned implicitly. The fusion of the results of all binary classifiers, equivalently produces the entire coverage for all classes defined by their Voronoi regions. Therefore, the results of these fused binary decodings should coincide with the optimal ML decoder.

For more detailed explanations on this approach, as well as experimental studies suggesting its efficiency, the reader is refered to [3] and [4].

It is also very important to mention that given a test example to be classified, unlike any other method, the complexity of decoding in this approach comprises only the SVM functional evaluations and does not incur any Hamming distance computation or decoding computations as in LDPC or other coding methods. The reason is due to the fact that a produced codeword is directly referring to a storage point in memory corresponding to the most likely class. However, the SVM functional evaluations is involved and could be very high, if the nonlinear kernels are too complex. In the next section we consider this fact in more details.

3 Choice of Binary Classifiers

After converting the multi-class problem to $l = \lceil \log_2 M \rceil$ binary classification problems, the important issue is the choice of these binary classifiers which is influenced by the specifications of the application. The computational budget of both training and testing (or equivalently encoding and decoding) stages, the memory storage issues and the amount of prior information we might have about the distribution of the data are among the factors that should be considered in this respect.

The richness behind VapnikChervonenkis learning theory and generalization guarantees it provides, and also the efficiency of the practical implementations of the algorithm makes the SVM the method of choice in many binary classification problems. In [3] and [4], SVM with Gaussian kernels was used and shown to provide good performance.

However, the complexity of both training and testing phases could be intractable in many practical applications. The complexity of training for kernelled binary SVM is between quadratic and cubic of the whole number of instances and the complexity of testing is quadratic to the number of classes because in the current setup, all classes become support vectors. This complexity is due to the fact that the number of free parameters of the algorithm is big, which as was shown in [4], implies that the algorithm might overtrain the data. Moreover, the standard implementation of SVM does not have any assumption on the data distribution, while in identification applications, one might have some prior knowledge of the distribution. Therefore, due to these factors, one might think of using alternative approaches. We consider next the use of Gaussian Mixture Models for the binary classification.

3.1 Gaussian Mixture Model for Classification

After converting the multi-class problem to binary problems using the proposed approach, half of the classes are randomly labeled as ones and the other half as zeros. Therefore, equivalently, in each binary classification there are two classes with very complicated distributions each. Since every distribution can be approximated using a mixture of Gaussians, we fit a GMM to each class from the training set $\mathcal{X}$ and the binary labels $\mathcal{C}_j(i)$'s with $1 \leq j \leq l$ and $1 \leq i \leq M$ where $\mathcal{C}_j(i)$ is the i^{th} element of the j^{th} column of the coding matrix:

$$p_0(\mathbf{x}|\mathcal{X}) = \sum_{i:\mathcal{C}(i)=0} \pi_i \phi(\mathbf{x}; \mu_i, \Sigma_i),$$

$$p_1(\mathbf{x}|\mathcal{X}) = \sum_{i:\mathcal{C}(i)=1} \pi_i \phi(\mathbf{x}; \mu_i, \Sigma_i),$$

$\phi(\mathbf{x}; \mu, \Sigma)$ is a multivariate Gaussian kernel characterised by its mean and covariance, respectively and the π_i's are the mixing coefficients.

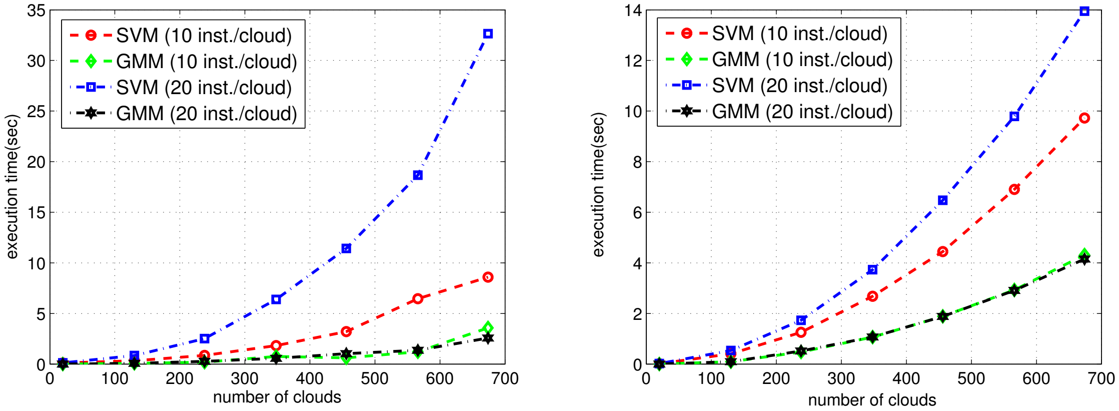
The simplest situation is the case where each of the original multi-labeled classes is modeled with one Gaussian kernel, so that we will have a total of $M/2$ kernels for each of p_0 and p_1. The unknown parameters are derived using the Expectation-Maximization algorithm. Here we could incorporate any knowledge about the mean and covariance of classes into the algorithm resulting in its very fast convergence and thus small complexity of training.

Given a test data $\mathbf{y}$, we classify it by evaluating the value of each of the two PDFs and finding the one which is maximal. The complexity of testing an example in this case will be linear in M which is much more reasonable than the kerneled SVM.

3.2 Experimental Comparison

Having reduced the multi-class problem to several binary problems using the proposed coding, we compare in this part the performance of the binary SVM classifier with the fitted GMM. The experimental setup is the same as in section2.3, but here instead of multi-labeled classes, we assign randomly to each of the generated Gaussian clouds zeros and ones to represent one of the equivalent binary problems.

We once train the data with a kerneled SVM, then we fit one GMM for each of the zero-labeled and one-labeled set of data with the standard Expectation-Maximization algorithm. As initial values of the algorithm, we use the empirical mean of the data points for each cloud. For dimension $N = 10$, two different numbers of instances per cluster, $\sigma^2_{inter} = 10$ and $\sigma^2_{intra} = 1$ and varied number of total clusters, the training/fitting time for SVM/GMM are sketched in Figure 4(a).



(a) training/fitting time

(b) testing

Figure 4: Time complexity of algorithms versus the number of clouds

For the test data, again we add Gaussian clouds with covariance $\sigma_z^2 \mathbb{I}_N$ to the class centers. We define the signal to noise ratio as $SNR = 10 \log_{10} \frac{\sigma^2_{inter}}{\sigma^2_Z}$. For 100 test data per cluster generated with $SNR = 5db$ and varied number of clouds, we compare the testing time for each of the methods in Figure 4(b). The performance of the methods in the same setup are measured by their accuracy as plotted in Figure 5. Due to the specifications of the current setup, the ground truth here is considered to be Euclidean distance comparison with the a priori known cloud centers.

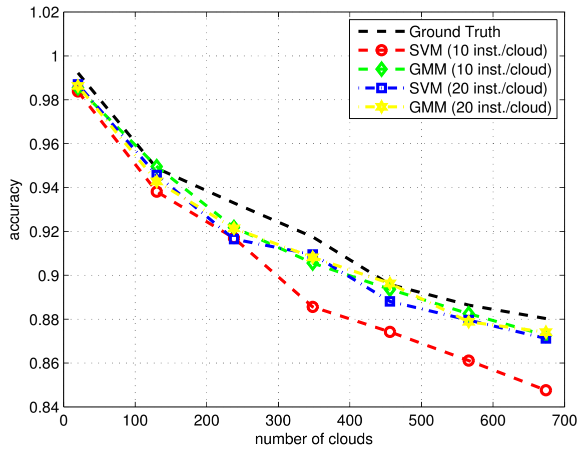


Figure 5: Accuracy versus cloud numbers

As is clear from the figures, both training and testing time complexities of the GMM are significantly lower than SVM which is polynomial in the number of classes. It is also important to point out that the good performance of GMM is due to the current setup where every original multi-label class is Gaussian. However, for more complicated scenarios, one could model each of the original classes as a mixture of Gaussians instead of one cloud.

4 Summary

The communication system setup for the ECOC approach in multi-class classification was examined to show why the conventional codes are working below expectation. A proposed coding approach was considered which due to the efficiency of the space partitioning and the homogeneity of the equivalent binary problems has an optimal performance. Apart from the coding, the choice of the base binary classifiers was considered in terms of complexity and the assumption on distributions. We showed that the GMM could be considered as an alternative for SVM, expecially in identification scenarios where one is concerned with complexity and may have some prior knowledge on the data.

Acknowledgments

The work presented in this paper was supported by a grant from Switzerland through the Swiss Contribution to the enlarged European Union (PSPB-125/2010).

References

[1] Dietterich, T.G., Bakiri, G.: Solving multiclass learning problems via error-correcting output codes. J. Artif. Intell. Res. (JAIR) **2** (1995) 263–286

[2] Marrocco, C., Simeone, P., Tortorella, F.: Embedding reject option in ecoc through ldpc codes. In: Proceedings of the 7th International Conference on Multiple Classifier Systems. MCS'07, Berlin, Heidelberg, Springer-Verlag (2007) 333–343

[3] Ferdowsi, S., Voloshynovskiy, S., Kostadinov, D.: Content identification: binary content fingerprinting versus binary content encoding. In: Proceedings of SPIE Photonics West, Electronic Imaging, Media Forensics and Security V, San Francisco, USA (January, 23 2014)

[4] Ferdowsi, S., Voloshynovskiy, S., Gabriel, M., Korytkowski, M.: Artificial intelligence and soft computing - 13th international conference, icaisc 2014, zakopane, poland, june 1-5, 2014, proceedings. In: ICAISC (1). Lecture Notes in Computer Science, Springer (2014)