J Sign Process Syst (2011) 65:413–430
DOI 10.1007/s11265-010-0526-1

Information–Theoretic Multiclass Classification Based on Binary Classifiers

On Coding Matrix Design, Reliability and Maximum Number of Classes

Sviatoslav Voloshynovskiy · Oleksiy Koval · Fokko Beekhof · Taras Holotyak

Received: 15 January 2010 / Revised: 1 May 2010 / Accepted: 3 September 2010 / Published online: 17 September 2010

Abstract In this paper, we consider the multiclass classification problem based on sets of independent binary classifiers. Each binary classifier represents the output of a quantized projection of training data onto a randomly generated orthonormal basis vector thus producing a binary label. The ensemble of all binary labels forms an analogue of a coding matrix. The properties of such kind of matrices and their impact on the maximum number of uniquely distinguishable classes are analyzed in this paper from an information-theoretic point of view. We also consider a concept of reliability for such kind of coding matrix generation that can be an alternative to other adaptive training techniques and investigate the impact on the bit error probability. We demonstrate that it is equivalent to the considered random coding matrix without any bit reliability information in terms of recognition rate.

Keywords Classification · Coding matrix design · Reliability · Maximum number of classes · Complexity

1 Introduction

In this paper we will address the multiclass categorization problem in a Machine Learning formulation requiring the assignments of labels to instances that belong to a finite set of classes ($M > 2$). While multiclass versions of most classification algorithms exist (e.g., [4]), they tend to be complex [10]. Therefore, a more common approach is to construct the multiclass classifier by combining the outputs of several binary classifiers [1, 6], an approach that also extends to *error correcting output codes* (ECOC).

The ECOC framework consists of two main steps: a *coding* step, where the codeword or some representation of an entry is assigned to a row of a coding matrix, and a *decoding* step, where a given observation is mapped into the most similar codeword of the coding matrix. There are many methods of coding matrix design based on a predefined set of codewords that follow different heuristics with the overall idea to maximize the inter-codeword Hamming distances, as that is believed to correspond to the most robust coding matrix design in terms of classification accuracy. However, these predefined coding matrices are problem-independent and can not cover a broad class of varying models. The design of an optimal decoder minimizing the overall classification error probability is also mainly accomplished based on the minimum Hamming distance decoder, which is a form of hard decoding. Although several score-based decoding rules (e.g., loss-based and loss-weighted decoding) attempt to consider the effect of binary classification reliability in the overall fusion rule, theoretically justified probabilistic fusion rules are still missing. Despite several recent remarkable exceptions [5, 7, 12], these problems are little studied and the problem of joint coding matrix design and probabilistic decoding maximizing the number of uniquely recognizable classes is of great practical interest.

This paper was partially supported by SNF projects 200021-111643 and 200021-1119770 and the Swiss IM2 project.

S. Voloshynovskiy (✉) · O. Koval · F. Beekhof · T. Holotyak
Department of Computer Science, University of Geneva, 7 route de Drize, CH 1227, Geneva, Switzerland
e-mail: svolos@unige.ch
URL: http://sip.unige.ch

The remaining part of the paper is organized in the following way. The information-theoretic formulation of multiclass classification problem based on the set of binary classifiers is given in Section 2. The proposed approach is introduced in Section 3, where both performance and complexity are analyzed for two types of decoding. Experimental validation results are presented in Section 4. Finally, Section 5 contains conclusions and draws some future extensions of the obtained results.

Notations We use capital letters to denote scalar random variables X, bold capital letters to denote vector random variables $\mathbf{X}$, corresponding small regular letters x and small bold letters $\mathbf{x}$ to denote the realizations of scalar and vector random variables, respectively. The superscript N is used to denote length-N vectors $\mathbf{x} = \{x(1), x(2), \ldots, x(N)\}$ with ith element $x(i)$. $\mathbf{b}_\mathbf{x}$ is used to denote the binary version of $\mathbf{x}$. We use $X \sim p_X(x)$ or simply $X \sim p(x)$ to indicate that a random variable X is distributed according to $p_X(x)$. The mathematical expectation of a random variable $X \sim p_X(x)$ is denoted by $E_{p_X}[X]$ or simply by $E[X]$ and σ_X^2 denotes the variance of X. Calligraphic fonts $\mathcal{X}$ denote sets $X \in \mathcal{X}$ and $|\mathcal{X}|$ denotes the cardinality of the set. Finally, $\mathbf{I}_N$ denotes a $N \times N$ identity matrix.

2 Problem Formulation: Information–Theoretic Limits

In this paper we will follow the information-theoretic machine learning approach thus providing a link with coding theory for optimal joint coding matrix and decoder design and estimation of the maximum number of uniquely distinguishable classes.

Assuming that the data are independent or weakly dependent and can be treated as almost identically distributed, one can use the definition of *information density*:

$$I(N) = \frac{1}{N} \log_2 \frac{p(\mathbf{x}, \mathbf{y})}{p(\mathbf{x})p(\mathbf{y})}, \tag{1}$$

where $\mathbf{x}$ is the template for the learning set and $\mathbf{y}$ is a template of the data to be classified, N is the template length, and $p(\mathbf{x}, \mathbf{y})$, $p(\mathbf{x})$ and $p(\mathbf{y})$ are joint probability density of $\mathbf{X}$ and $\mathbf{Y}$ and their marginals, respectively. When the template distributions are known, a so-called *recognition* or *identification capacity* [11] can be used:

$$\bar{I}(X; Y) = \lim_{N \to \infty} E[I(N)], \tag{2}$$

provided that the limit is well defined and the expectation is taken with respect to the joint distribution of $\mathbf{X}$ and $\mathbf{Y}$ that reduces to the Kullback–Leibler Divergence (KLD) between $p(\mathbf{x}, \mathbf{y})$ and $p(\mathbf{x})p(\mathbf{y})$. This also corresponds to the Bayesian multiclass classifier minimizing the average probability of misclassification and is invariant under linear invertible transformations. In this case, the maximum number of classes that can be recognized with vanishing probability of error under the above conditions is limited as [3]:

$$M \leq 2^{N\bar{I}(X;Y)}. \tag{3}$$

For the case of i.i.d. Gaussian data $\mathbf{X} \sim \mathcal{N}(\mathbf{0}, \sigma_X^2 \mathbf{I}_N)$ and the memoryless additive white Gaussian model of interaction $\mathbf{y} = \mathbf{x} + \mathbf{z}$ with $\mathbf{Z} \sim \mathcal{N}(\mathbf{0}, \sigma_Z^2 \mathbf{I}_N)$, the recognition capacity is readily found as:

$$\bar{I}(X; Y) = \frac{1}{2} \log_2 \frac{1}{1 - \rho_{XY}^2} = \frac{1}{2} \log_2 \left(1 + \frac{\sigma_X^2}{\sigma_Z^2}\right), \tag{4}$$

where $\rho_{XY}^2 = \frac{\sigma_X^2}{\sigma_X^2 + \sigma_Z^2}$ is a squared correlation coefficient (SCC) between X and Y. The results can be extended to a more general model of interaction with training model $p(\mathbf{v}|\mathbf{x})$ and observation model $p(\mathbf{y}|\mathbf{x})$ shown in Fig. 1a. For the i.i.d. Gaussian case with the training data model $\mathbf{v} = \mathbf{x} + \mathbf{z}_t$ and observation model $\mathbf{y} = \mathbf{x} + \mathbf{z}_r$ with $\mathbf{Z}_t \sim \mathcal{N}(\mathbf{0}, \sigma_{Z_t}^2 \mathbf{I}_N)$ and $\mathbf{Z}_r \sim \mathcal{N}(\mathbf{0}, \sigma_{Z_r}^2 \mathbf{I}_N)$ (Fig. 1b), the recognition capacity is:

$$\bar{I}(V; Y) = \frac{1}{2} \log_2 \frac{1}{1 - \rho_{VY}^2}, \tag{5}$$

where $\rho_{VY}^2 = \frac{\sigma_X^4}{(\sigma_X^2 + \sigma_{Z_t}^2)(\sigma_X^2 + \sigma_{Z_r}^2)}$ is the SCC between V and Y that transforms into $\rho_{VY}^2 = \rho_{XY}^2$ for $\mathbf{v} = \mathbf{x}$.

In the general case, the computation of recognition capacity (Eq. 2) based on the information density

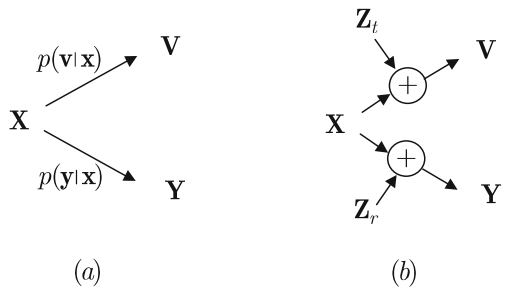

Figure 1 General model of interaction: **a** generic mapping of $\mathbf{x}$ into training data $\mathbf{v}$ and observation data $\mathbf{y}$ and **b** Gaussian setup with the additive noise $\mathbf{Z}_t$ for training and $\mathbf{Z}_r$ for classification/recognition.

(Eq. 1) requires knowledge of the corresponding joint and marginal pdfs. If these pdfs are available, one can achieve this capacity using optimal Bayessian classifier. However, due to the lack of reliable priors and the high complexity of multiclass classification, it is often reduced to the multiple binary problems. The general structure of such a reduction is shown in Fig. 2, where the outputs of L binary classifiers (BC) are combined into a binary label $\mathbf{b}_\mathbf{x}(m) \in \{-1; +1\}^L$ for a given input $\mathbf{x}(m)$. This procedure will be referred to as a binary label estimation (BLE). In this case, the general model of interaction with the training model is reduced to two binary symmetric channels (BSC) [3] defined by the binary counterparts $\mathbf{b}_\mathbf{x}, \mathbf{b}_\mathbf{v}, \mathbf{b}_\mathbf{y}$ and $\mathbf{b}_{\mathbf{z}_t}, \mathbf{b}_{\mathbf{z}_r}$ of $\mathbf{x}, \mathbf{v}, \mathbf{y}$ and $\mathbf{z}_t, \mathbf{z}_r$, respectively: $\mathbf{b}_\mathbf{v} = \mathbf{b}_\mathbf{x} \oplus \mathbf{b}_{\mathbf{z}_t}$ and $\mathbf{b}_\mathbf{r} = \mathbf{b}_\mathbf{x} \oplus \mathbf{b}_{\mathbf{z}_r}$ with $\oplus$ denoting the modulo 2 addition. Assuming that the $\mathbf{B}_\mathbf{x}$ follows the Bernoulli probability mass function with the probability $\Pr[-1] = \Pr[+1] = 0.5$ and $\mathbf{Z}_t, \mathbf{Z}_r$ are Bernoulli vectors with the parameters P_{b_t} and P_{b_r}, one can easily estimate the recognition rate of this binarized system as:

$$\bar{I}\left(B_\mathbf{v}; B_\mathbf{y}\right) = H\left(B_\mathbf{y}\right) - H\left(B_\mathbf{y}|B_\mathbf{v}\right) = 1 - H_2\left(P_{b_\sum}\right), \quad (6)$$

where $H(.)$ denotes the entropy, $H_2(P_{b_\sum}) = -P_{b_\sum} \times \log_2 P_{b_\sum} - (1 - P_{b_\sum}) \log_2(1 - P_{b_\sum})$ is the binary entropy and $P_{b_\sum} = P_{b_t}(1 - P_{b_r}) + P_{b_r}(1 - P_{b_t})$. Therefore, the maximum number of recognizable classes is bounded as:

$$M_b \leq 2^{L\bar{I}(B_\mathbf{v}; B_\mathbf{y})}, \quad (7)$$

that also provides the lower bound on the required number of binary classifiers L for a given M according to the recognition rate (Eq. 6):

$$L \geq \frac{\log_2 M_b}{1 - H_2\left(P_{b_\sum}\right)}. \quad (8)$$

It is important to note that most of existing multiclass classification strategies based on the set of binary classifiers use either one-vs-one or one-vs-all designs that require $\frac{M_b(M_b-1)}{2}$ and M_b binary classifiers respectively. Comparing these numbers with the theoretical bound (Eq. 8), one can immediately conclude that these designs are highly superficial and overestimated, for large M_b that makes these approaches highly unfeasible from the point of view of complexity and storage. Moreover, the one-vs-one or one-vs-all designs completely disregard the impact of binary classification error and keep the number of classifiers fixed. Contrarily, in the scope of the advocated approach the needed number of classifiers is proportional to the theoretically minimum requested number of bits to uniquely encode M_b classes, i.e., $\log_2 M_b$ and the additional fraction $1 - H_2(P_{b_\sum})$ is requested to compensate the ambiguity caused by the binary classification errors.

These theoretical results represent the basis for the further consideration of various multiclass classification strategies based on the set of binary classifiers. In following, we will concentrate on the popular ECOC approach to characterize its assumptions with respect to the achievable limits.

Once the binary class label $\mathbf{b}_\mathbf{x}(m) \in \{-1; +1\}^L$ is generated, the training stage of the ECOC consists in the assignment of a class label $m \in \mathcal{M}$, where $\mathcal{M} = \{1, 2, \cdots, M\}$, to the closest codeword $\mathbf{c}(n)$ of the coding matrix $\mathcal{C} = \{\mathbf{c}(1), \mathbf{c}(2), \cdots, \mathbf{c}(M)\}$ with $\mathbf{c}(m) \in \{-1; +1\}^L$ and storing this re-assignment $m \to n$. One can also re-index the codewords as $\mathbf{c}(\hat{n}) \to \mathbf{c}'(m)$ resulting into a coding matrix $\mathcal{C}' = \{\mathbf{c}'(1), \mathbf{c}'(2), \cdots, \mathbf{c}'(M)\}$. The explanation of the training stage re-assignment is shown in Fig. 3. The label re-assignment at the training stage can be performed in two different ways depending on the information about the training input. If it is given that the input $\mathbf{x}(m)$ belongs to one of M classes, one can use the maximum likelihood (ML) rule. In the above considered binary case, the ML based matching of $\mathbf{b}_\mathbf{x}(m)$ with $\mathbf{c}(n) \in \mathcal{C}$ is reduced to a minimum Hamming distance rule:

$$\hat{n} = \arg \min_{1 \leq n \leq M} d^H(\mathbf{c}(n), \mathbf{b}_\mathbf{x}(m)), \quad (9)$$

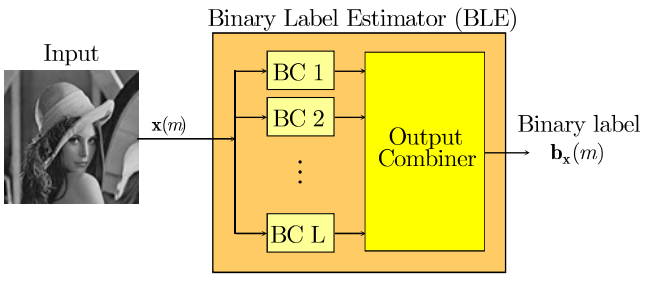

Figure 2 General structure of binary label generation based on L binary classifiers.

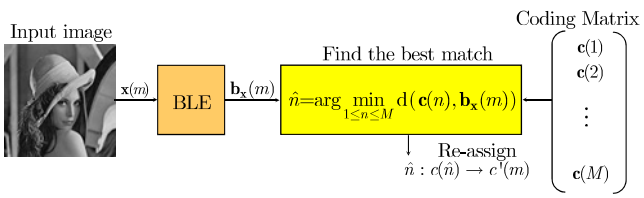

Figure 3 Training stage.

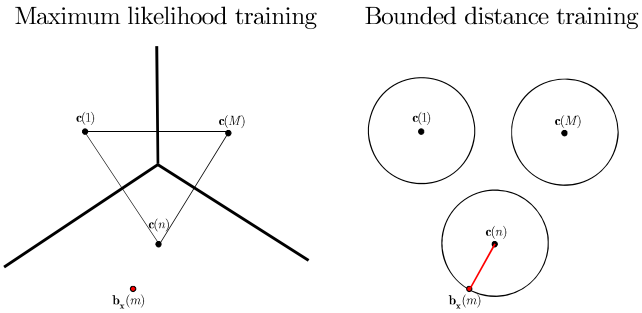

Figure 4 Training strategies: *left* the ML training and *right* BD training.

where $d^H(.,.)$ denotes the Hamming distance. Otherwise, if the training input $\mathbf{x}(m)$ is irrelevant to any class, it is preferable to train with an erasure option based on the bounded distance (BD) decision rule. According to the BD rule, one decides that the input $\mathbf{x}(m)$ belongs to some class n, if its binary counterpart $\mathbf{b}_\mathbf{x}(m)$ matches the coding matrix entry $\mathbf{c}(n)$ as:

$$d^H\left(\mathbf{c}(\hat{n}), \mathbf{b}_\mathbf{x}(m)\right) \leq \gamma L, \tag{10}$$

for a unique $\hat{n}$ and a properly defined threshold γ. If such a unique $\hat{n}$ cannot be found, the rule decides that the training sample does not belong to any of the assigned class codewords $\mathbf{c} \in \mathcal{C}$.[1]

The difference between the ML and BD-based training is schematically shown in Fig 4. In both cases, the training rules are based on the Hamming distance, where all bits are treated equally, and no soft information about bit reliability is used. Obviously, the success of label re-assignment at the training stage strongly depends on the construction of the coding matrix $\mathcal{C}$, i.e., on the success of selection of the codewords $\mathbf{c} \in \mathcal{C}$, that is by itself not a trivial problem that should be solved using all available priors. Selecting an inappropriate $\mathbf{c}$ might result in a very high training/classification error when the number of classes increases.

The main problem of ECOC based training consists in the very low probability of success of finding a correspondance between a binary class label $\mathbf{b}_\mathbf{x}(m)$ and the coding matrix. The training rule (Eq. 10) should ensure that $\mathbf{b}_\mathbf{x}(m)$ is close to some $\mathbf{c}(\hat{n})$ with some fidelity γL. There is no dependence between these two binary vectors and the probability that they match by chance is very low. Moreover, for large numbers of classes M_b, L grows according to Eq. 8 or even quadratically if one-vs-one strategies are used. The increase of the length L reduces the chance of a matching at random at an exponential rate. Equivalently, it means that the training error P_{b_t} approaches $\frac{1}{2}$ and, regardless of the value of P_{b_r}, the overall error P_{b_z} in Eq. 6 is close to $\frac{1}{2}$, resulting in recognition rate nearing zero.

The classification stage is shown in Fig. 5. The observed vector $\mathbf{y}$, which represents some distorted version of $\mathbf{x}(m)$, is transformed into a binary counterpart $\mathbf{b}_\mathbf{y}$, which is matched versus the codewords of the re-assigned matrix $\mathbf{c}' \in \mathcal{C}'$. The matching can also be accomplished based on either the ML or BD decision rules. The decoded index $\hat{m}$ indicates the class label deduced for the observation $\mathbf{y}$. As it was mentioned in the introduction, the ECOC are facing several problems related to the ambiguity in the construction of the optimal coding matrix and sub-optimal hard training/classification based on the Hamming distance. Additionally, the impact of training error on the overall system performance remains an open issue. Finally, the maximum number of reliably recognizable classes based on the ECOC is not established with respect to the limit (Eq. 7) due to the lack of direct correspondence between the selection of the codewords in the coding matrix and training/classification statistics.

For these reasons, we will consider the multiclass classification problem based on the direct assignment of class labels deduced from the training data. In this way, the link between the training data and the entries of the coding matrix will be naturally defined. Moreover, such a construction will explicitly avoid the training error and allows the application of the information-theoretic framework for the accurate computation of the achievable recognition rate with respect to the theoretical limit (Eq. 6). Finally, this framework makes it possible to easily characterize the reliability of each binary classifier and to develop new soft classification rules with the increased classification rate comparatively to the hard decision-based ECOC. In addition, we will demonstrate how this framework can lead to low-complexity classification that is especially important for multiclass applications.

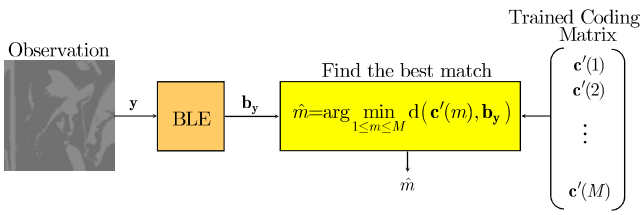

Figure 5 Classification stage.

[1] In practice, the ML or BD training is directly implemented in the error correction code decoder used in the ECOC.

3 Proposed Approach

Instead of following the above discussed construction of an ECOC coding matrix and training the classifiers, we will consider a scheme that targets maximization of the number of correctly distinguishable classes based on binary classification with respect to Eq. 7. We will demonstrate that the structure of the coding matrix that corresponds to the above objective and simultaneously maximizes minimum distance is obtained directly from the training stage by mapping each vector of a training set into a row of the coding matrix.

Without loss of generality, we will assume at this stage that we have a mapper/encoding function $f(.)$ that maps the training set and observation entries as $f : \mathcal{X}^N \to \mathcal{B}_{\mathbf{x}}^L$, $\mathcal{B}_{\mathbf{x}} \in \{-1, +1\}$, and $f : \mathcal{Y}^N \to \mathcal{B}_{\mathbf{y}}^L$, $\mathcal{B}_{\mathbf{y}} \in \{-1, +1\}$, respectively. Therefore, the link between the binary representation $\mathbf{b}_{\mathbf{x}}$ of vector $\mathbf{x}$ and its noisy counterpart $\mathbf{b}_{\mathbf{y}}$ of vector $\mathbf{y}$ is defined according to the BSC model. We assume that $\mathbf{v} = \mathbf{x}$ and noise in the direct domain might cause a bit to flip in the binary domain with a certain average probability $\bar{P}_b$. The corresponding maximum number of recognizable classes M_b is defined by Eq. 7 with $P_{b_t} = 0$ and $P_{b_\Sigma} = P_{b_r} = \bar{P}_b$. Extending the mutual information between binary representations or classifiers outputs in Eq. 7, one obtains:

$$\bar{I}\left(B_{\mathbf{x}}; B_{\mathbf{y}}\right) = H\left(B_{\mathbf{x}}\right) - H\left(B_{\mathbf{x}}|B_{\mathbf{y}}\right). \tag{11}$$

It can be immediately noticed that to maximize M_b, one needs to maximize $\bar{I}(B_{\mathbf{x}}; B_{\mathbf{y}})$ for a given L that can be achieved by: (a) maximization of $H(B_{\mathbf{x}})$ and (b) minimization of $H(B_{\mathbf{x}}|B_{\mathbf{y}})$. In the considered binary case, the maximum value of term $H(B_{\mathbf{x}})$ is 1, that can be achieved for equiprobable independent data, i.e., $\Pr(-1) = \Pr(+1) = 0.5$. This suggests that the multi-class rate maximizing coding matrix should have equiprobable independent binary entries that is known as a random coding matrix.

The second term $H(B_{\mathbf{x}}|B_{\mathbf{y}})$ is defined by the average error probability of binary classification $\bar{P}_b$ and $H(B_{\mathbf{x}}|B_{\mathbf{y}}) = H_2(\bar{P}_b)$. In the considered setup it is not possible to control $\bar{P}_b$. Therefore, we will consider an alternative design where $\bar{P}_b$ can be considerably reduced due to basis adaptation based on decision reliability information. At the same time, we will demonstrate that this decrease of bit error probability comes at the cost of an increased size of the coding matrix that equivalently reduces the recognition rate. In fact, we will demonstrate that these two approaches are equivalent in terms of recognition rate and simply represent different designs of coding matrices. Nevertheless, this construction could be very useful for certain low complexity implementations of multiclass classifiers for a large number of classes. The only increase of the recognition rate can be achieved by changing the fusion rule based on reliability information for a fixed size of the random coding matrix.

3.1 Design of Coding Matrix and Training

According to the above analysis the coding matrix should maximize $H(B_{\mathbf{x}})$. Simultaneously, we have assumed the existence of a generic encoding function $f(.)$ that maps the real-data entries into binary representations stored in the coding matrix. To achieve the maximum of $H(B_{\mathbf{x}}) = 1$, $B_{\mathbf{x}}$ should be equiprobable and independent. In this section, we will consider a possible design of such kind of encoding function based on random projections and binarization.

The random projections are considered as a dimensionality reduction step and are performed as:

$$\tilde{\mathbf{x}} = \mathbf{W}\mathbf{x}, \tag{12}$$

where $\mathbf{x} \in \mathbb{R}^N$, $\tilde{\mathbf{x}} \in \mathbb{R}^L$, $\mathbf{W} \in \mathbb{R}^{L \times N}$ and $L \leq N$ and $\mathbf{W} = (\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_L)^T$ consists of a set of projection basis vectors $\mathbf{w}_i \in \mathbb{R}^N$ with $1 \leq i \leq L$. Instead of following a particular consideration of mapping $\mathbf{W}$, we will assume that $\mathbf{W}$ is a random matrix. The matrix $\mathbf{W}$ has elements $w_{i,j}$ that are generated from some specified distribution. An $L \times N$ random matrix $\mathbf{W}$ whose entries $w_{i,j}$ are independent realizations of Gaussian random variables $W_{i,j} \sim \mathcal{N}(0, \frac{1}{N})$ is of a particular interest for our study. In this case, such a matrix can be considered as an almost *orthoprojector*, for which $\mathbf{W}\mathbf{W}^T \approx \mathbf{I}_L$.[2]

The second step uses labeling or Grey codes to ensure closeness of labels for close vectors. Such kind of labeling is known as *soft hashing*. When only the most significant bit of the Grey code is used, it is known as binary or *hard hashing*.

The most simple quantization or binarization of extracted features is known as *sign random projections*:

$$b_{\mathbf{x}_i} = sign\left(\mathbf{w}_i^T \mathbf{x}\right), \tag{13}$$

where $b_{\mathbf{x}_i} \in \{-1; +1\}$, with $1 \leq i \leq L$ and $sign(a) = +1$, if $a \geq 0$ and -1, otherwise. The vector $\mathbf{b}_{\mathbf{x}} \in \{-1; +1\}^L$ computed for all projections represents a binary label of the class computed from the vector $\mathbf{x}$. The ensemble of

[2] Otherwise, one can apply special orthogonalization techniques to ensure perfect orthogonality.

all binary labels $\mathbf{b}_{\mathbf{x}}(m)$ with $1 \leq m \leq M$ forms a coding matrix.

It can be readily validated that under the proper selection of a random projection matrix with Gaussian basis vectors one can expect that the projected vector will follow a Gaussian distribution $\tilde{\mathbf{X}} \sim \mathcal{N}(\mathbf{0}, \tilde{\mathbf{K}}_X)$ where the covariance matrix $\tilde{\mathbf{K}}_X$ can easily be estimated. However, it does not guarantee that $\tilde{\mathbf{K}}_X$ will be diagonal to ensure uniformly distributed bits after binarization. To achieve this goal, one can apply a diagonalization operator Φ, which is designed based on the principal component analysis (PCA) transform, i.e., its basis vectors correspond to the eigenvectors of $\tilde{\mathbf{K}}_X$. As a result, one can obtain uncorrelated random variables in the vector $\tilde{\mathbf{X}}$ in general and independent components for Gaussian distributions. The independent components will lead to equiprobably valued bits which satisfies the necessary conditions for entropy maximization.

Another strategy consists in the decorrelation data at the first stage and then applying the random projection transform. To avoid the computation of optimal basis vectors one can use a block-based discrete cosine transform (DCT) with the fixed basis for each block, which closely approximates the PCA for the correlated data.

At the same time, one can notice that the binary labels are deduced directly from the training data and stored in the coding matrix thus avoiding the additional stage of matching a binary label deduced from a training vector with the closest row of the coding matrix as it is done for the methods discussed in the introduction.

Depending on the number of available training inputs per class, several constructions of the coding matrix are possible. In the case of a single training input per class, the training inputs are directly assigned into the coding matrix as shown in Fig. 6, i.e., $\mathbf{c}(m) = \mathbf{b}_{\mathbf{x}}(m)$, $m \in \mathcal{M}$. These codewords are automatically considered as the centers of the corresponding decoding regions. In the case of multiple training inputs per class, one can proceed in two different ways. Assuming K training inputs per class, i.e., for a class m one has $\mathbf{b}_{\mathbf{x}}(m, k)$ inputs with $1 \leq k \leq K$, one can directly store all K training inputs in the coding matrix by assigning to all of them a corresponding common bin index m as shown in Fig. 7. The classification is then reduced to establishing the bin index based on the BD decoder. Contrarily, one can deduce centroids for each class $\mathbf{c}(m) = \text{centroid}(\mathbf{b}_{\mathbf{x}}(m, 1), \cdots, \mathbf{b}_{\mathbf{x}}(m, K))$ and store them in the coding matrix similarly to the above case. The classification is based on the same BD decoder but with the increased decoding region.

The performance and complexity of classification severely depends on the type of information produced by the binary classifiers. In following, we will consider the classification under *hard* and *soft* decoding.

3.2 Classification Under Hard Decoding

The classification under hard decoding assumes that only binary outputs of the BLE are used for the classification, i.e., the binary counterpart $\mathbf{b}_{\mathbf{y}}$ of observation $\mathbf{y}$ is matched versus the coding matrix $\mathcal{C}$ composed of binary entries $\mathbf{b}_{\mathbf{x}}(m)$, $m \in \mathcal{M}$. The decision about the class label $\hat{m}$ is made based on the BD decoder:

$$d^H\left(\mathbf{b}_{\mathbf{y}}, \mathbf{b}_{\mathbf{x}}(\hat{m})\right) \leq \gamma L, \tag{14}$$

for a unique $\hat{m}$. If the number of errors or Hamming distance between $\mathbf{b}_{\mathbf{y}}$ and $\mathbf{b}_{\mathbf{x}}(m)$ is smaller than γL, a positive decision is taken, otherwise $\hat{m}$ is rejected.

It should be noticed that this classification strategy is a particular case of the Forney's erasure decision rule [9]:

$$p\left(\mathbf{b}_{\mathbf{y}} | \mathbf{b}_{\mathbf{x}}(\hat{m})\right) \geq 2^{\tau L}, \tag{15}$$

where τ is the threshold related to γ as $\gamma = \frac{-\tau + \log_2(1-\bar{P}_b)}{\log_2 \frac{1-\bar{P}_b}{\bar{P}_b}}$. It can be also shown that this threshold

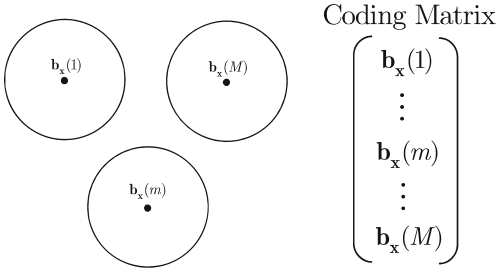

Figure 6 Coding matrix based on single training input.

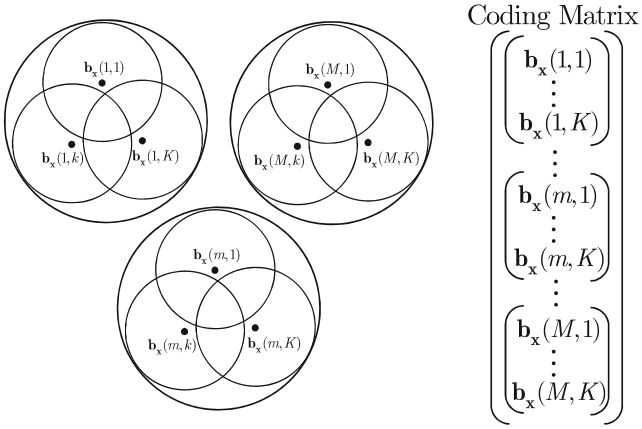

Figure 7 Coding matrix based on multiple training inputs.

should satisfy $\tau \leq -H_2(\bar{P}_b)$ for the unique decoding of index m and rejection hypothesis.

3.2.1 Performance Analysis

The performance of this classifier depends on the selection of the threshold γ with respect to the probability of error $\bar{P}_b$. Therefore, the performance analysis consists of two parts. First, we will determine the probability of bit error $\bar{P}_b$ for the considered above coding matrix. Then, the performance of the BD decoder can be expressed as the average probability of misclassification:

$$P_e = \frac{1}{M_b}\sum_{m=1}^{M_b} \Pr[\hat{m} \neq m | \text{given class } m], \tag{16}$$

with respect to the relationship between γ, $\bar{P}_b$, M and L.

The bit error probability indicates the mismatch of signs between $\tilde{x}$ and $\tilde{y}$, i.e., $\Pr[sign(\tilde{x}) \neq sign(\tilde{y})]$:

$$P_b = \Pr\left[\tilde{Y} \geq 0 | \tilde{X} < 0\right] \Pr\left[\tilde{X} < 0\right] \tag{17}$$

$$+ \Pr\left[\tilde{Y} < 0 | \tilde{X} \geq 0\right] \Pr\left[\tilde{X} \geq 0\right], \tag{18}$$

or by symmetry for $\Pr[\tilde{X} < 0] = \Pr[\tilde{X} \geq 0] = \frac{1}{2}$ it can be rewritten as:

$$\begin{aligned} P_b &= \Pr\left[\tilde{Y} < 0 | \tilde{X} \geq 0\right] \\ &= 2\int_0^{\infty}\int_{-\infty}^{0} p(\tilde{y}|\tilde{x})p(\tilde{x})d\tilde{y}d\tilde{x} \\ &= 2\int_0^{\infty} P_{b|\tilde{x}}p(\tilde{x})d\tilde{x}, \end{aligned} \tag{19}$$

where:

$$\begin{aligned} P_{b|\tilde{x}} &= \int_{-\infty}^{0} p(\tilde{y}|\tilde{x})d\tilde{y} \\ &= \int_{-\infty}^{0} \frac{1}{\sqrt{2\pi\sigma_Z^2}} e^{\frac{-(\tilde{y}-\tilde{x})^2}{2\sigma_Z^2}} d\tilde{y} \\ &= Q\left(\frac{|\tilde{x}|}{\sigma_Z}\right), \end{aligned} \tag{20}$$

stands for the bit error probability for a given projection coefficient $\tilde{x}$ under the assumption that $p(\tilde{x}, \tilde{y})$ corresponds to jointly Gaussian distribution in the random projection domain. The modulo sign is used for completeness of the consideration for the above symmetrical case when $\tilde{X} < 0$. One can immediately note that some projections can be more reliable in terms of bit error probability than others and the Eq. 20 can be a good measure of bit *reliability*.

The origin of $P_{b|\tilde{x}}$ for a given configuration of $\mathbf{x}$ and $\mathbf{w}_i$ is shown in Fig. 8. The vector $\mathbf{x}$ forms an angle θ_{XW_i} with the basis vector $\mathbf{w}_i$ and the projection results into the scalar value $\tilde{x}_i$. The closer angle θ_{XW_i} is to $\pi/2$, the smaller value $\tilde{x}_i$ will be. This leads to a larger probability that the sign of $\tilde{y}_i$ will be different from the sign of $\tilde{x}_i$ that is shown by the gray area under the curve of $p(\tilde{y}_i)$. One can immediately note that since the projections are generated at random there is generally no guaranty that two vectors can be collinear. However, at the same time some of the projections might form angles with $\mathbf{x}$ that deviate from $\pi/2$ thus leading to a smaller probability of binary classification error. Incerasinf the

Substituting Eq. 20 into Eq. 19, one obtains:

$$\begin{aligned} P_b &= 2\int_0^{\infty} Q\left(\frac{|\tilde{x}|}{\sigma_Z}\right) \frac{1}{\sqrt{2\pi\sigma_X^2}} e^{\frac{-\tilde{x}^2}{2\sigma_X^2}} d\tilde{x} \\ &= \frac{1}{\pi}\arccos(\rho_{\tilde{X}\tilde{Y}}), \end{aligned} \tag{21}$$

where $\rho^2_{\tilde{X}\tilde{Y}} = \frac{\sigma_X^2}{\sigma_X^2+\sigma_Z^2}$ is the squared correlation coefficient between $\tilde{X}$ and $\tilde{Y}$.

Remarkably, the average probability of error depends on the correlation coefficient between the direct domain data and is determined by the channel and source statistics. It can be easily verified for the more general model that the average bit error probability (Eq. 21) is:

$$\bar{P}_b = \pi^{-1}\arccos(\rho_{VY}), \tag{22}$$

that coincides with Eq. 21 for the noiseless training case. Obviously, this sort of ambiguity during training

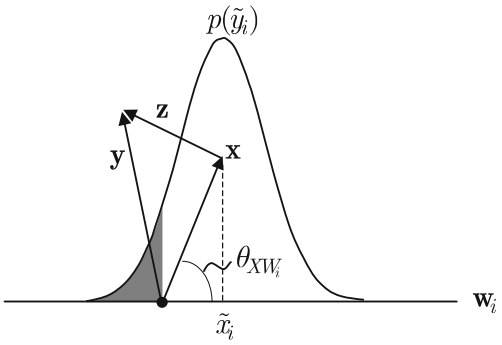

Figure 8 The bit error probability for a given $\mathbf{x}$ and some $\mathbf{w}_i$.

causes an additional increase in probability since $\rho_{XY} \geq \rho_{VY}$.

The second part of the performance analysis consists in finding the average probability of classification error for the decoding rule (Eq. 14) according to the established bit error probability (Eq. 21). We assume that all codewords in the coding matrix are symmetric and equivalently processed. Due to the symmetry of the coding matrix, one can consider a representative case of $m = 1$ for the output $\mathbf{B_y}(1)$. The average probability of error (Eq. 16) can be rewritten as:

$$\begin{aligned} P_e &= \Pr\left[(E(1) > \gamma L) \cup \bigcup_{m=2}^{M_b} (E(m) \leq \gamma L)\right] \\ &\leq P_{e_1} + \leq P_{e_2}, \end{aligned} \tag{23}$$

where the inequality follows from the union bound and $E(1) = d^H(\mathbf{B_x}(1), \mathbf{B_y}(1))$ and $E(m) = d^H(\mathbf{B_x}(m), \mathbf{B_y}(1))$.

P_{e_1} corresponds to the probability of error due to a false negative:

$$\begin{aligned} P_{e_1} &= \Pr\left[d^H\left(\mathbf{B_x}(1), \mathbf{B_y}(1)\right) > \gamma L\right] \\ &= \Pr\left[w^H\left(\mathbf{B_x}(1) \oplus \mathbf{B_y}(1)\right) > \gamma L\right] \\ &= \sum_{\gamma L \leq w^H \leq L} \binom{L}{w^H} \bar{P}_b^{w^H} \left(1 - \bar{P}_b\right)^{L - w^H} \\ &\leq 2^{-LD(\gamma || \bar{P}_b)}. \end{aligned} \tag{24}$$

where w^H denotes the Hamming weight and $D(\gamma || \bar{P}_b) = \gamma \log_2 \frac{\gamma}{\bar{P}_b} + (1 - \gamma) \log_2 \frac{1-\gamma}{1-\bar{P}_b}$ stands for the divergence between $0 \leq \gamma \leq 1$ and $1 \leq \bar{P}_b \leq 1$. By increasing the number of binary classifiers L, one can make this error probability arbitrary small.

P_{e_2} corresponds to the probability of error due to a false positive:

$$\begin{aligned} P_{e_2} &= \sum_{m=2}^{M_b} \Pr\left[d^H\left(\mathbf{B_x}(m), \mathbf{B_y}(1)\right) \leq \gamma L\right] \\ &= \sum_{m=2}^{M_b} \Pr\left[w^H\left(\mathbf{B_x}(m) \oplus \mathbf{B_y}(1)\right) \leq \gamma L\right] \\ &= \sum_{m=2}^{M_b} \sum_{0 \leq w^H \leq \gamma L} \binom{L}{w^H} \left(\frac{1}{2}\right)^L \\ &\leq (M_b - 1) 2^{-LD\left(\gamma || \frac{1}{2}\right)} \\ &= (M_b - 1) 2^{-L(1 - H_2(\gamma))} \\ &= 2^{-L\left(1 - \frac{1}{L}\log_2 M_b - H_2(\gamma)\right)}. \end{aligned} \tag{25}$$

It is important to note that if $H_2(\gamma) \leq 1 - \frac{1}{L}\log_2 M_b$, one can make this error probability arbitrary small by increasing L.

Thus, combing P_{e_1} and P_{e_2}, one obtains:

$$\begin{aligned} P_e &\leq 2^{-LD(\gamma || \bar{P}_b)} + (M_b - 1) 2^{-L(1 - H_2(\gamma))} \\ &\leq 2 \cdot 2^{-LD\left(\gamma_{\text{opt}} || \bar{P}_b\right)}, \end{aligned}$$

where $\gamma_{\text{opt}} = \frac{1 - \frac{1}{L}\log_2 M_b + \log_2(1 - \bar{P}_b)}{\log_2\left(\frac{1 - \bar{P}_b}{\bar{P}_b}\right)}$ defines the optimal threshold minimizing the average probability of error.

Remarkably, for recognition capacity achieving the maximum number of classes satisfying Eq. 7 with the mutual information (Eq. 11), i.e., $\frac{1}{L}\log_2 M_b \leq 1 - H_2(\bar{P}_b)$, the above optimal threshold yields $\gamma_{\text{opt}} = \bar{P}_b$. This means that the decoding region around each codeword is defined by the radius $\bar{P}_b L$ and the identification capacity is achieved based on the DB decoder.

Alternatively one can obtain this result by analyzing the maximum number of errors T_b in the observation $\mathbf{b_y}$ as a result of passing via the BSC. The number of bits that can be flipped is random and T_b follows binomial distribution, i.e., $T_b \sim B(L, \bar{P}_b)$. According to the weak law of large numbers, one can state with the probability close to 1 that T_b is very close to its mean $\bar{P}_b L$. Thus, the threshold should be chosen accordingly to keep all deviations due to the noise within the acceptance region that also corresponds to the above recognition capacity achieving selection of the threshold.

Not less important condition is to ensure that the observations $\mathbf{b_y}$ that are not related to any entry of coding matrix $\mathbf{b_x}(m)$ are not falsely accepted. We define the corresponding probability as probability of false acceptance under the hypothesis H_0:

$$\begin{aligned} P_f &= \Pr\left[\bigcup_{m=1}^{M_b} d^H\left(\mathbf{B_x}(m), \mathbf{B_y}\right) \leq \gamma L | H_0\right] \\ &\leq_{(a)} \sum_{m=1}^{M_b} \Pr\left[d^H\left(\mathbf{B_x}(m), \mathbf{B_y}\right) \leq \gamma L | H_0\right] \\ &= M_b \Pr\left[d^H\left(\mathbf{B_x}(m), \mathbf{B_y}\right) \leq \gamma L | H_0\right] \\ &\leq_{(b)} 2^{-L\left(1 - \frac{1}{L}\log_2 M_b - H_2(\gamma)\right)}, \end{aligned} \tag{26}$$

where (a) follows from the union bound and (b) from the Chernoff bound on the tail of binomial distributions $\mathcal{B}(L, 0.5)$ that results from $d^H(\mathbf{B_x}(m), \mathbf{B_y}) \sim \mathcal{B}(L, 0.5)$ under the hypothesis H_0. It can be readily verified that

for the above optimal threshold γ_{opt} both P_e and P_f are minimized.

3.2.2 Complexity Analysis

Being capacity achieving, the classification based on ML or BD hard decoding requires the computation of M_b Hamming distances that might be prohibitively high for large numbers of classes M_b. Thus, the classification complexity is exponential in terms of the input length, i.e., $\mathcal{O}(2^{L_b})$ with $L_b = LI(B_x; B_y) = L(1 - H_2(\bar{P}_b))$. Another alternative for the BD decoding exists based on the fact that the observation $\mathbf{b_y}$ can be at the distance T_b from the original codeword $\mathbf{b_x}(m)$. Therefore, in the case of large M_b, instead of exhaustively checking all codewords, one can only check the presence of the true codeword $\mathbf{b_x}(m)$ within the Hamming sphere around $\mathbf{b_y}$ defined by $t_{b_{\max}}$. In this case, the number of verifications one needs to perform is:

$$\mathbf{N} = \sum_{t_b=0}^{t_{b\max}} \binom{L}{t_b}, \tag{27}$$

where $t_{b_{\max}}$ is the maximum number of errors.

According to the above mentioned weak law of large numbers, the most likely distorted codewords $\mathbf{b_y}$ will be on the radius $\bar{t}_b = L\bar{P}_b$ from $\mathbf{b_x}$ for sufficiently large L. It can also easily be confirmed using the Stirling approximation formula [8] that the number of these codewords will not exceed:

$$\bar{\mathbf{N}} = \binom{L}{\bar{t}_b} \approx 2^{LH_2\left(\frac{\bar{t}_b}{L}\right)}, \tag{28}$$

which yields $\bar{\mathbf{N}} \leq 2^{LH_2(\bar{P}_b)}$ for $\bar{t}_b = L\bar{P}_b$. Thus, the resulting complexity of this classification strategy is still exponentional with the input length. It is now of the order $\mathcal{O}(2^{LH_2(\bar{P}_b)} \log_2 M_b)$ that is reduced with respect to the previous case. Moreover, it is also dependent on the quality of the observation data given by the bit error rate $\bar{P}_b$. The term $\log_2 M_b$ is due to the complexity of checking the existence of a given bitstring in a sorted version of the coding matrix.

However, even though this problem is known to be NP-hard, in the next section we will consider an alternative decoding rule that preserves the same number of uniquely recognizable classes as the BD decoder but operates with considerably lower complexity.

3.3 Classification Under Soft Decoding

The above classification problem is NP-hard and there is no known algorithm to deterministically compute its solution efficiently. Therefore, in this section we will try to reformulate the decoding problem in such a way where some additional information about the binary classifier reliability is provided by the BLE from the observation $\mathbf{y}$.

According to the analysis of probability of bit error (Eq. 20), one can note that not all bits in the vector $\mathbf{b_y}$ have an equal probability of error. The larger the magnitude $\tilde{x}_i$ of projector vector $\mathbf{x}$ on the basis vector $\mathbf{w}_i$, the smaller the probability of bit error. This property is the basis for our classification based on soft decoding. Soft decoding is based on the decomposition of the projected vector $\tilde{\mathbf{y}}$ i^{th} component as:

$$\tilde{y}_i = sign(\tilde{y}_i)\left|\tilde{y}_i\right| = b_{y_i}\left|\tilde{y}_i\right|, \tag{29}$$

where $b_{y_i} = sign(\tilde{y}_i)$ and $\left|\tilde{y}_i\right|$ denotes the magnitude of $\tilde{y}_i$. It should be pointed out that the coding matrix contains only binary signs of the projected vector $\tilde{\mathbf{x}}$. Therefore, the reliability information is obtained directly from the magnitude of observation $\tilde{\mathbf{y}}$. Obviously, for a given set of training data one can always find a set of vectors $\mathbf{w}_i$, $1 \leq i \leq L$ that minimizes the overall bit error probability. However, keeping in mind the facts that (a) the number of classes might be in the order of millions; and (b) it can be constantly updated; such an optimization problem looks highly unfeasible. Contrarily, in the scope of the proposed approach no additional feature extraction based on training set or PCA is needed that considerably reduces the complexity of learning and classification procedures.

Moreover, the use of soft information makes it possible to enhance both the performance and reduce the complexity of classification that is considered in the next sections.

3.3.1 Performance Analysis

The performance analysis of classification under soft decoding includes the consideration of both achievable rate and average probability of classification error given the information about the bits' reliability. We first consider the impact of bit reliability on the achievable recognition rate along the analysis of practical decoding rules. In the second part of our analysis, we will investigate the possible reduction of the probability of bit error for different bit selection strategies. Finally, several practical decoders are considered in Section 3.3.2.

The achievable recognition rate of classification systems under hard data representation, i.e., binarized data, for both the coding matrix $\mathbf{b_x}(m)$, $1 \leq m \leq M$ and observation $\mathbf{b_y}$ is determined by $\bar{I}(B_{\mathbf{x}}; B_{\mathbf{y}})$. The

corresponding decoding strategies achieving this rate are the ML and BD decoders based on the computation of the Hamming distance $d^H(\mathbf{b}_\mathbf{x}(m), \mathbf{b}_\mathbf{y})$.

The achievable recognition rate of classification system using soft information, i.e., using the real-valued observation $\tilde{y}$, and binarized coding matrix $\mathbf{b}_\mathbf{x}(m)$, $1 \leq m \leq M$, is given by $\bar{I}(B_\mathbf{x}; \tilde{Y})$.

One can also consider a soft coding matrix $\tilde{\mathbf{x}}(m)$, $1 \leq m \leq M$, that will lead to the enhanced recognition rate $\bar{I}(\tilde{X}; \tilde{Y})$ as in Eq. 4.

Due to the data processing inequality [3], the following is true:

$$\bar{I}(\tilde{X}; \tilde{Y}) \geq \bar{I}(B_\mathbf{x}; \tilde{Y}) \geq \bar{I}(B_\mathbf{x}; B_\mathbf{y}). \tag{30}$$

The results of simulation for the average probability of classification error under the AWGN model and $M = 100$ classes using the ML decoder are shown in Fig. 9 for three setups considered above as a function of signal-to-noise ratio (SNR) defined as $\text{SNR} = 10\log_{10}\frac{\sigma_X^2}{\sigma_{Z_r}^2}$. The soft information considerably enhances the performance of classification system in part of both coding matrix and observation.

The positive impact of soft information leading to the bit reliability discrimination can be also observed for the average probability of bit error that is very important for the design of practical low-complexity decoding rules. The minimization of average probability of error based on the selection of reliable components can be practically implemented in two different ways based on *thresholding* or *order statistics*.

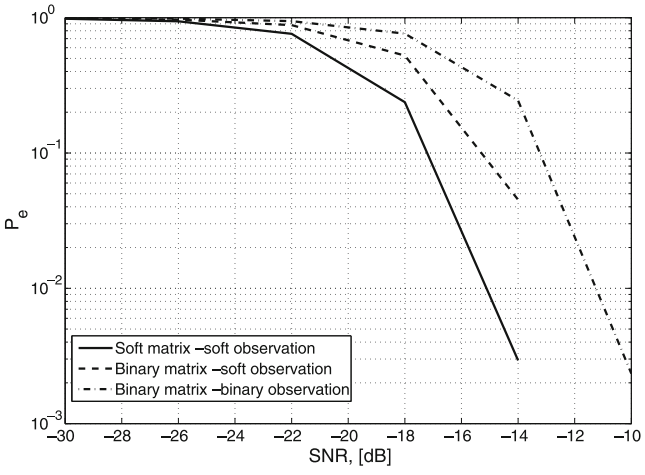

Figure 9 The average probability of classification error under AWGN for 100 classes for: **a** a soft coding matrix and soft observations, **b** a binarized coding matrix and soft observations, and **c** both a binarized coding matrix and observations.

The thresholding approach is based on the selection of all components whose magnitude $|\tilde{y}_i|$ is higher than a certain threshold $T_{\tilde{x}}$ that is shown in Fig. 10b. The corresponding average probability of bit error is:

$$\begin{aligned}\bar{P}_{b\,T} &= \frac{1}{\int_{T_{\tilde{x}}}^{\infty} p(\tilde{x}_i)d\tilde{x}_i}\int_{T_{\tilde{x}}}^{\infty} P_{b\,|\tilde{x}_i}\, p(\tilde{x}_i)d\tilde{x}_i \\ &= Q^{-1}\left(\frac{T_{\tilde{x}}}{\sigma_X}\right)\int_{T_{\tilde{x}}}^{\infty} Q\left(\frac{\tilde{x}_i}{\sigma_{Z_r}}\right)\frac{1}{\sqrt{2\pi\sigma_X^2}}e^{\frac{-\tilde{x}_i^2}{2\sigma_X^2}}d\tilde{x}_i,\end{aligned} \tag{31}$$

where the multiplier is the normalization constant corresponding to the fraction of distribution behind the threshold.

The practical application of this approach is facing two main concerns: (a) how many of overcomplete projections L are needed for any $\mathbf{x}$ to guarantee the necessary L'? and (b) what is a possible gain in $\bar{P}_{b\,T}$ versus $\bar{P}_b$? We will address the issues (a) and (b) in this section to demonstrate the feasibility of the proposed approach. At the same time, one should take into account the increase of the coding matrix size L to store the information about reliable projections that might affect the achievable recognition rate, where only L' are used for the classification. This explains the fact that the achievable recognition rate can not be enhanced due to the expected decrease of bit error probability due the simultaneous decrease of vector length from L to L'. It is easy to verify that the number of coefficients L' of random variable $\tilde{X}_i$ following Gaussian distribution and exceeding the threshold $T_{\tilde{x}}$ in L projections satisfies with high probability the following equation:

$$\Pr[L' \geq \ell] = 1 - F_{B_X}\left(L, \ell, \Pr\left[\tilde{X}_i > T_{\tilde{x}}\right]\right), \tag{32}$$

where ℓ is the necessary number of reliable coefficients in the coding matrix (like 32, 64 or 128), the

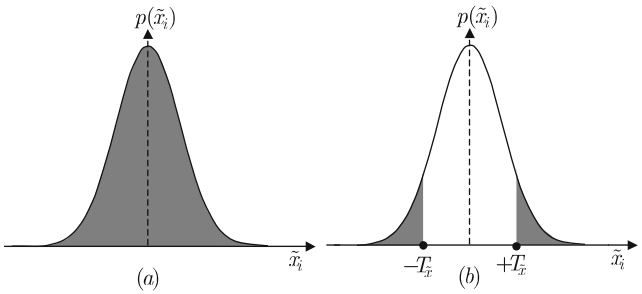

Figure 10 Bit error reliability framework: **a** all values $\tilde{x}_i$ are taken into account and **b** only the most reliable values are taken for the basis selection (to be normalized to 1).

binomial cumulative distribution function is designated as $F_{B_X}(L, \ell, \Pr[\tilde{X}_i > T_{\tilde{x}}])$ and $\Pr[\tilde{X}_i > T_{\tilde{x}}] = Q\left(\frac{T_{\tilde{x}}}{\sigma_X}\right)$. For practical applications, one can assume that to ensure the existence of the desired ℓ with high probability $1-\epsilon$, the quantity $F_{B_X}(L, \ell, \Pr[\tilde{X}_i > T_{\tilde{x}}])$ should be bounded by a small ϵ that will be further assumed not to exceed 10^{-10}. This result is shown in Fig. 11. Obviously, the larger the number of reliable bits that is requested in the coding matrix, the more projections L should be generated for a given threshold. At the same time, the increase of the threshold leads to an exponential number of projections L. Although these numbers seem to be quite high, for example for $L' = 64$ and $T_{\tilde{x}} = 2.5$, the required L is about $2 \cdot 10^4$, this can be compared to the discrete Fourier transform of image of size 512×512 for the optimal feature selection out of about $2.6 \cdot 10^5$ transform coefficients. Therefore, this problem is computationally feasible. To answer the second question about the possible gain in $\bar{P}_{b_T}$ versus $\bar{P}_b$, we will plot the corresponding results (Eqs. 21 and 31) for different $T_{\tilde{x}}$ as the function of SNR. The results are shown in Fig. 12. The proposed optimization strategy to the reliable projection selection clearly demonstrates a considerable increase in the accuracy of binary classifiers with respect to the blind projection selection. The results coincide for $T_{\tilde{x}} = 0$ that confirms the fact that all projections are blindly taken into account for the coding matrix generation. Although this approach demonstrates an excellent performance in the terms of $\bar{P}_b$, nevertheless the number of projections L' exceeding the given threshold $T_{\tilde{x}}$ is varying for each observation $\mathbf{y}$. As a consequence, thresholding is not a very useful approach for practical implementations.

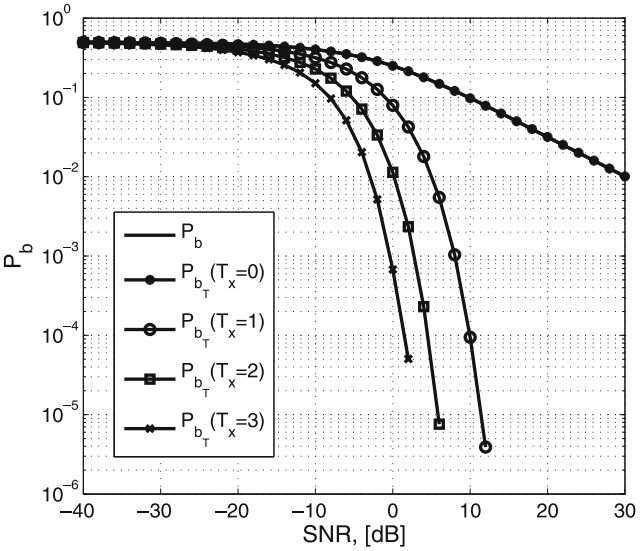

Figure 12 The average bit error probability for all and reliable only projections selection for various thresholds.

Therefore, we use the second approach based on order statistics, where the reliability information is sorted in the ascending order and the predefined quantity of projections L' out of L are considered to be reliable ones with the certain probability of bit error $\bar{P}'_b$. Obviously, if $L' = L$, all projections are used and $\bar{P}'_b = \bar{P}_b$. The information-theoretic model behind this approach will be considered in the next section along the complexity analysis. Here, we show the impact of L' on $\bar{P}'_b$ in Fig. 13 for different $L' = 8, 12, 18, 32, 128, 512$ for $L = 512$. One observers similar behavior as for

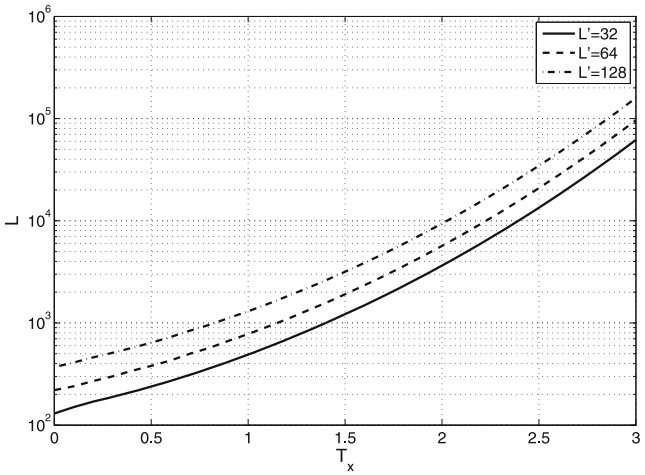

Figure 11 The estimation of necessary number of projections L for the desired number of reliable bits in coding matrix for $\sigma_X^2 = 1$.

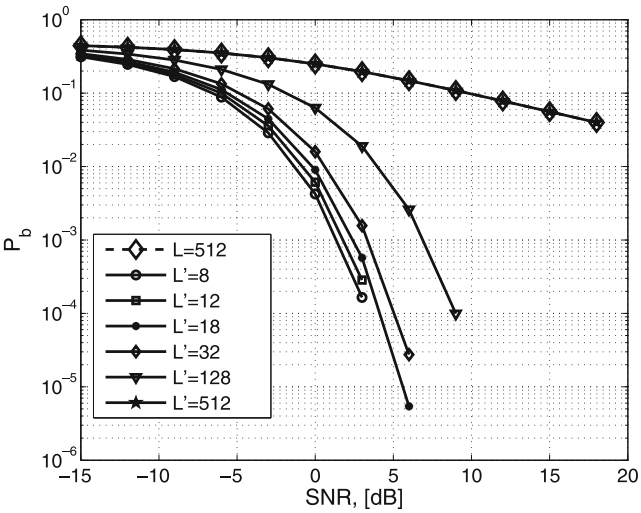

Figure 13 The average bit error probability for the most reliable components based on the order statistics.

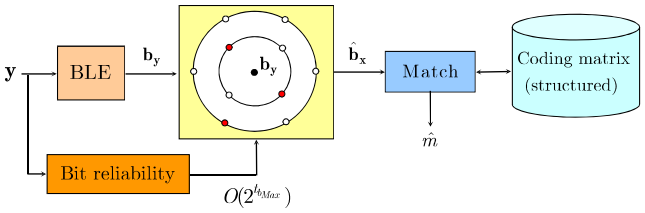

Figure 14 Classification based on the reliable bits producing the most likely codewords within the Hamming sphere.

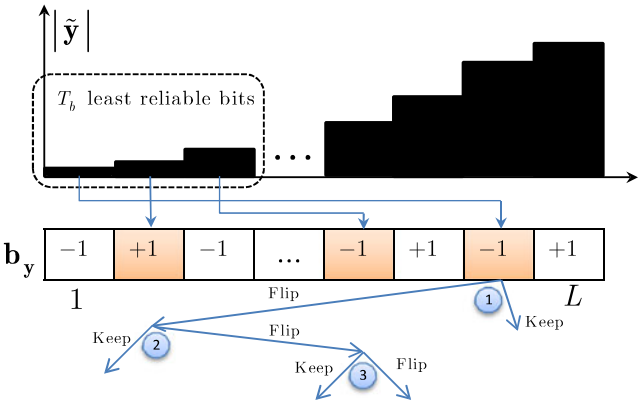

Figure 16 Decoding strategy based on branch and bound algorithm and least reliable bits.

the thresholding approach but with a fixed number of reliable components. Obviously, these two approaches are related to each other since one can always recompute the requested number of reliable bits into the equivalent threshold as $T_{\tilde{x}} = \sigma_Y Q^{-1}(1 - L'/(2L))$, where $\sigma_Y = \sqrt{\sigma_X^2 + \sigma_{Z_r}^2}$ and $Q^{-1}(.)$ stands for the inverse $Q-$function.

3.3.2 Complexity Analysis

In this section we present two practical approaches to the design of low-complexity soft decoders based on a set of binary classifiers. The first approach is based on the BD decoder described in Sections 3.2.1 and 3.2.2 and the second one uses a so-called overcomplete projections.

The BD decode considers all possible candidates that are within the distance γL or $t_{b_{\max}}$ from $\mathbf{b_y}$. However, contrarily to the decoder presented in Section 3.2.2, soft information is used to select only those codewords within the Hamming sphere that are the most likely, as shown in Fig. 14. The obtained set of codeword candidates is checked versus the database. If several candidates are chosen, the one with the highest likelihood is preferred. Optionally, the decoder can output the list of several most probable candidates ranked by their likelihoods that can be useful in some identification, retrieval and data mining applications of the considered classification.

The generation of candidate codewords within the Hamming sphere is explained in Fig. 15. According to the results presented in the previous section regarding order statistics based reliable bit extraction, the ordered magnitudes $|\tilde{\mathbf{y}}|$ are used to find T_b least reliable bits in the corresponding locations in the binary vector $\mathbf{b_y}$. One possible implementation of the decoding using a *branch and bound* algorithm [2] is shown in Fig. 16, when one first starts with the least reliable bit and sequentially flips the remaining T_b bits creating all possible combinations that are checked versus the coding matrix. Remarkably, the complexity of this algorithm is reduced to $\mathcal{O}(2^{T_b} \log_2 M) = \mathcal{O}(2^{L\bar{P}_b} \log_2 M)$ with respect to the previous case $\mathcal{O}(2^{LH_2(\bar{P}_b)} \log_2 M)$.

Moreover, one can also enhance the classification accuracy by using soft information about $|\tilde{\mathbf{y}}|$.

The second low-complexity decoding approach is obtained considering only the most reliable bits con-

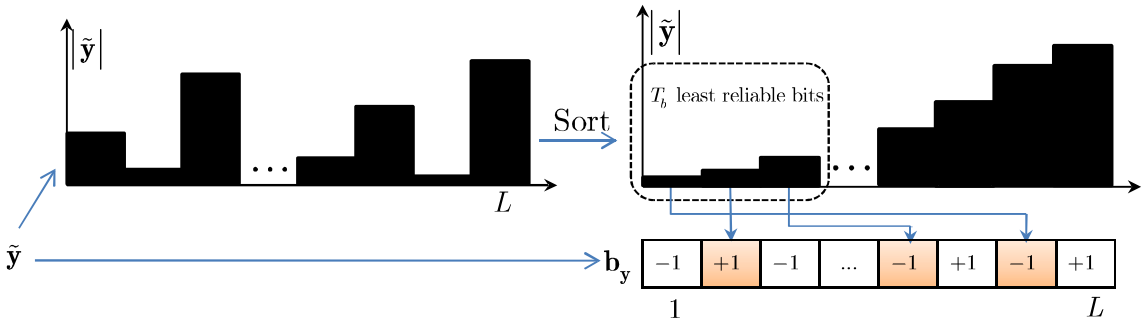

Figure 15 Projected observation decomposition into reliability and binary components.

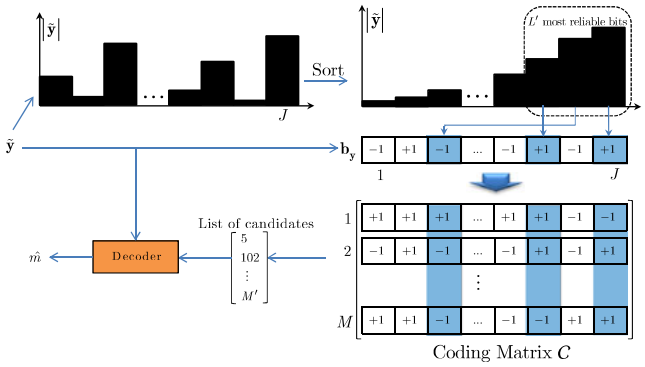

Figure 17 Channel splitting based decoding.

trary to the above decoding and completely disregarding the least reliable bits. The block diagram of this approach is shown in Fig. 17. The approach uses J overcomplete projections to select the L' most reliable components based on the sorted magnitudes $|\tilde{\mathbf{y}}|$. The number of projections L' is selected in such a way to guarantee a very low probability of error similar to the results presented in Fig. 13. Accordingly, one can easily compute either the mean $\bar{t}'_b = L'\bar{P}'_b$ or maximum number $t'_{b\,\max} = B^{-1}(1-\epsilon, L', \bar{P}'_b)$ of error bits in L' chosen bits, where $\bar{P}'_b$ denotes the probability of bit error in the L' most reliable bits, $B^{-1}(.)$ stands for inverse binomial cumulative density function and ϵ is an arbitrarily small chosen probability that the number of error bits exceeds $t'_{b\,\max}$. The experimental results for $J = 512$ and $L' = 8, 12, 18, 32, 128, 512$ are shown in Fig. 18. It is important to note that one can expect with

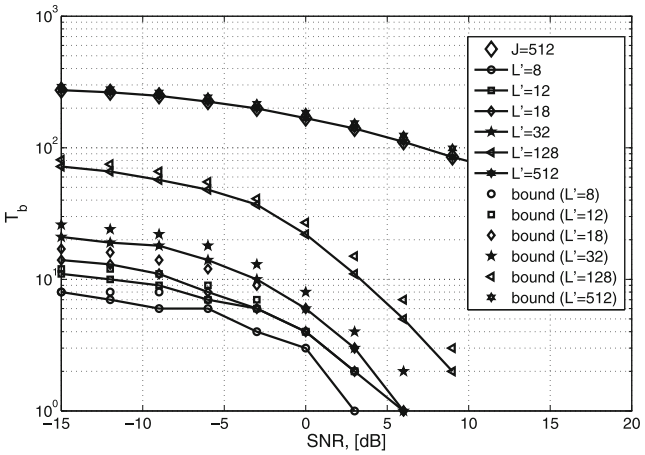

Figure 18 Number of error bits in $L' = 8, 12, 18, 32, 128, 512$ most reliable bits selected out $J = 512$ projections based on the order statistics with the corresponding upper bound estimates.

high probability zero errors in the L' most reliable bits for a relatively small L' after a certain SNR. Therefore, the corresponding bits in $\mathbf{b_y}$ can be considered to be error-free, that makes it possible to straightforwardly find the corresponding codewords $\mathbf{b_x}(m')$ with $m' \in \mathcal{L}' = \{1, \cdots, M'\}$ in the coding matrix that have the same bits in the L' most reliable bit positions. These codewords form a list of candidates $\mathcal{L}'$ for further verification that can be even performed using the ML decoder with the acceptable complexity due to the relatively small cardinality M' of list $\mathcal{L}'$. The cardinality of list of candidates is $M' = 2^{\log_2 M - L'}$. For example, for 1 million classes ($M = 2^{20}$) and $L' = 12$, the list of candidates is $M' = 2^8$, i.e., 256 candidates that can be easily verified. Our experiments in Matlab™indicate that the classification of a single item to one of the 1 million classes approximately requires 121.62 seconds while the proposed method provides a result in about 0.82 second for $SNR \geq 5dB$.

It is also of interest to generalize the one-step decomposition described above into a multi-stage or hierarchical approach, as is schematically shown in Fig. 19. The sorted magnitude vector $|\tilde{\mathbf{y}}|$ is split into S blocks of length L_j, $1 \leq j \leq S$. The probability of bit error in each block $\bar{P}'_j$ can be computed that constitutes to the equivalent BSC_j. Therefore, the entire J bits are split into L_J equivalent BSCs. The resulting probability of error for all J bits can be computed as:

$$\bar{P}_b = \frac{1}{J}\sum_{j=1}^{S} L_j \bar{P}'_j, \tag{33}$$

and coincides with Eq. 21.

The resulting recognition rate represents the weighted sum of recognition rates of all equivalent BSCs:

$$R = \frac{1}{J}\sum_{j=1}^{S} L_j \left(1 - H_2\left(\bar{P}'_j\right)\right) \tag{34}$$

Therefore, one can obtain a flexible trade-off between the complexity and achievable rate by properly selecting the block sizes L_j for each stage of hierarchical search and sequentially reducing the list of candidates. It is important to note that this result can not be achieved by directly considering all L bits simultaneously. The proposed framework is conceptually similar to multistage decoding used in multilevel error correction codes and multiple access communication channels [3].

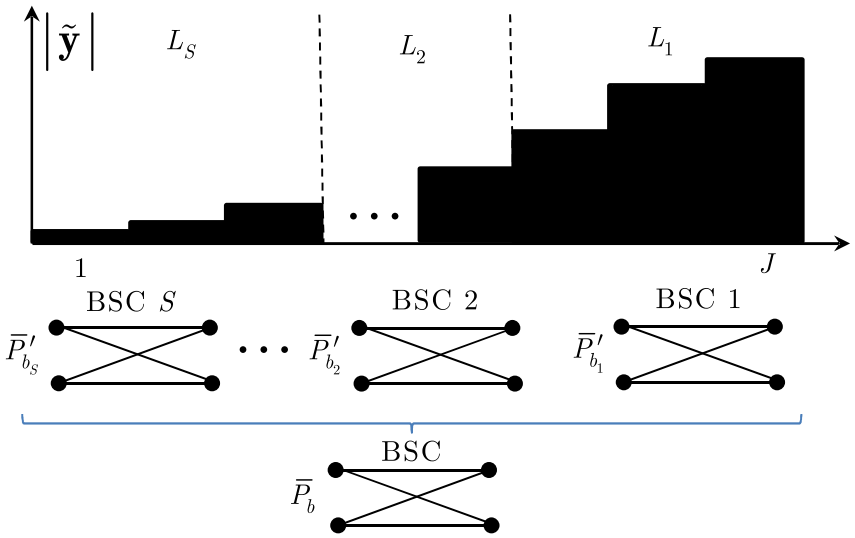

Figure 19 Channel splitting generalized model model.

4 Results of Computer Simulation

Using computer simulation, we have investigated (1) the probability of bit-error for different distortions in images; (2) the probability of classification error in a system of 2^{20} synthetic classes using the proposed branch-and-bound decoder; and (3) the probability of classification error when using an overcomplete transformation.

4.1 Bit Error Probabilities

The bit error probability provides an idea about the corresponding amount of errors and necessary number of trials per observation. The test database for these tests consists of $M = 1'000'000$ entries. We only use $N = 32 \times 32$ blocks for each image for simulation purposes. A binary feature vector of length $L = 32$ is extracted from each block and stored in a database. The tests have been performed under various distortion models including additive white Gaussian, uniform noise and lossy JPEG compression. The type of noise reflects the incompleteness of the knowledge of the data user regarding the classes and thus the mismatch with the labels stored in the database.

The bit error probabilities for the AWGN, additive uniform noise and lossy JPEG compression are shown in Figs. 20, 21 and 22, respectively. The observation model is considered in terms of the signal-to-noise ratio (SNR) defined as $\text{SNR} = 10 \log_{10} \frac{\sigma_X^2}{\sigma_Z^2}$ for additive noises and in terms of quality factor for JPEG compression.

In conclusion, the AWGN model of distortion leads to the largest probability of bit-error, and can therefore

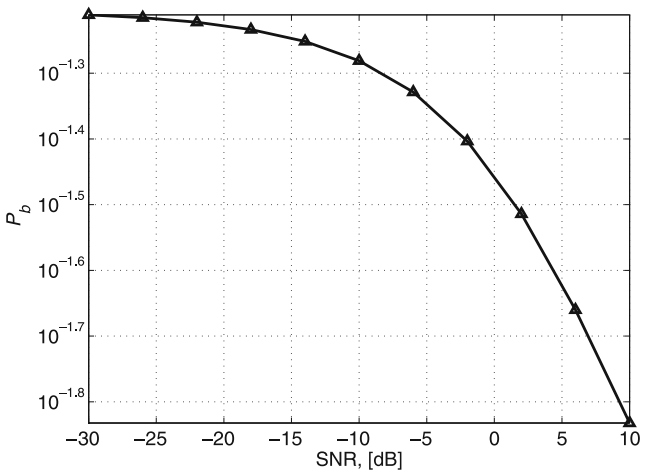

Figure 20 Bit error probability for AWGN.

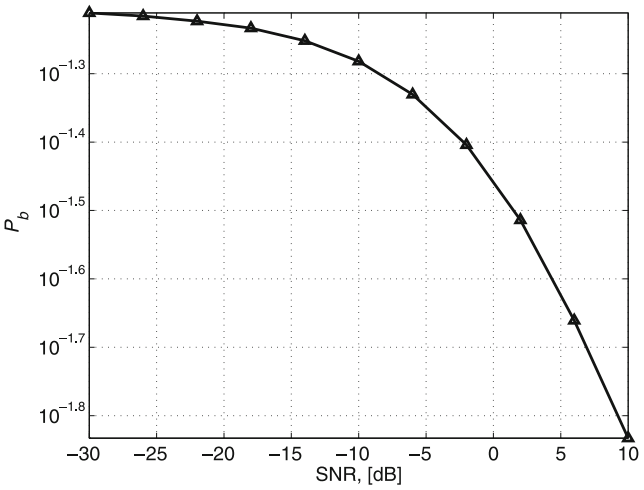

Figure 21 Bit error probability for additive uniform noise.

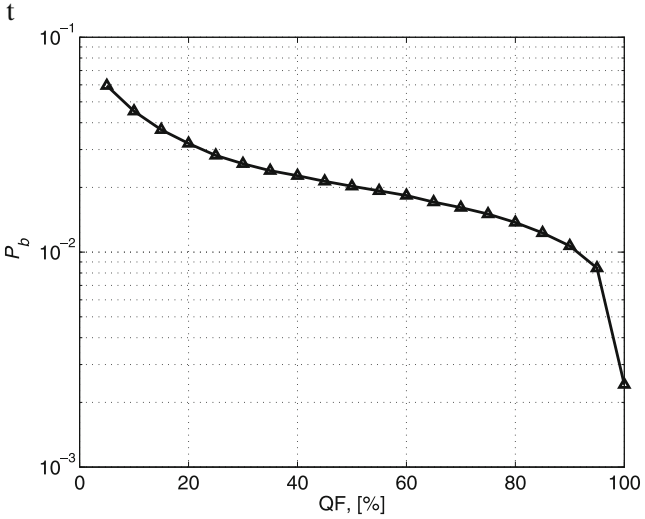

Figure 22 Bit error probability for lossy JPEG compression.

be considered as a worst-case scenario. The probability of bit-error for uniform noise is almost equivalent to that of the Gaussian distortion, nonetheless we have chosen the AWGN model of distortions for large-scale testing on both synthetic data and real images.

4.2 Branch-and-Bound Decoding

A number of simulations have been carried out, based on the Gaussian setup, where $M = 2^{20}$, $N = 1{,}024$, $L = 32$. The results are displayed in Fig. 23. In order to limit the calculation time, the number of flipped bits for the branch-and-bound decoders has limited to 16 ($= \frac{1}{2}L$), or less if that can be predicted based on the equivalent Binary Symmetric Channel. We have tested two versions of the branch-and-bound algorithm, one using the approximation of the cross-correlation, and a second one using the regular Hamming distance. These two variants are denoted as "NP-soft", for using soft information; and "NP-hard" when using integer Hamming distances, respectively. Both versions do select which bit to flip first based on reliability.

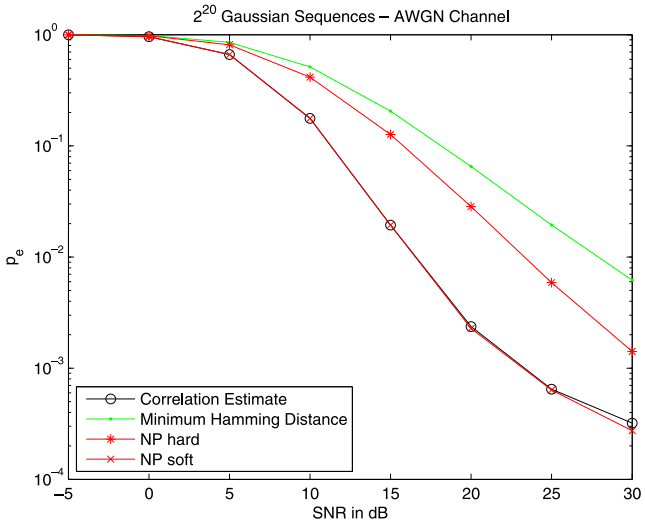

Figure 23 Accuracy of different decoders.

Clearly, the use of an approximation of the cross-correlation significantly improves the performance in terms of accuracy compared to the Hamming distance.

A very interesting phenomenon is that the branch-and-bound decoder outperforms the straightforward exhaustive Hamming-distance decoder even when only using the Hamming distance as a metric. The only explanation we have for this is that in certain cases, several codewords will have the same Hamming distance to the channel output. In such cases, both decoders will select the codeword they first come across. For the exhaustive decoder, that is the codeword with the lowest index in the codebook. For the NP-hard decoder on the other hand, the most likely variations on the channel output are tried first, so the first codeword found has a higher probability of being correct than the one with the lowest index of all codewords having the same Hamming distance to the channel output.

Compared to the exhaustive search ML-based cross-correlation decoder, the NP-soft decoder shows almost identical performance, but at a significantly reduced complexity, especially for higher SNRs.

It would be inturesting to compare the performance of the proposed method to those of existing methods. Unfortunately, there are several issues that make such a comparison unpractical, if not undesirable. Certain methods, such as one-vs-one binary strategies or Linear Discriminant Analysis require $O(M^2)$ operations during either training or classification, which, for the considered cases where M can be as high as 2^{20}, is prohibitively time-consuming.

Apart from practical arguments, a fair comparison is extremely difficult to make as any comparison between any two multi-class classification systems based on binary classifiers would reasonably require that both systems use an equal number of binary classifiers, i.e. a fair comparison should be done based on a chosen fixed value of L. Due to the general nature of one-vs-one or one-vs-all strategies, such a restriction poses a non-trival problem given the number of classes used in the experiments. Conversely, it is not straightforward to construct multi-class classifiers out of only 32 of these binary classifiers that can successfully classify 2^{20} classes.

4.3 Overcomplete Transforms and Hierarchical Decoding

To study the effects of overcomplete transforms, a set of images has been taken and cut into blocks of 64×64 pixels each, producing 16,384 blocks in total of 4,096 pixels. These have been projected 512 times each, and we consider what happens when only 12 out of 512 generated bits are used. To summarize, $M = 2^{14}$, $N =$ 4,096, $J = 512$ and $L' = 12$. Each block has been taken to represent one class each. Distorted versions of the images are then used as observations.

In the first test, the images are distorted by adding Gaussian noise, in accordance with the AWGN channel. In the Fig. 24, an exhaustive ML-based Hamming distance decoder using the full $J = 512$ bits has been compared to a decoder that selects candidates based on the $L' = 12$ most reliable bits, and then selects a final answer based on all $J = 512$ bits of each class in the set of candidate classes. The full exhaustive search decoder represents a performance limit in the binary domain. It is clear that the use of the faster procedure produces less accurate results, thus confirming the existence of a trade-off between complexity and accuracy.

The second test is based on JPEG-compressed images. The same two decoders are compared as in the previous test. As we can see in Fig. 25, the effects of compression are less severe than for adding Gaussian noise. This implies the conclusion that the amount of distortions and the probabilistic overlap between classes has an influence on the performance in terms of accuracy on the system.

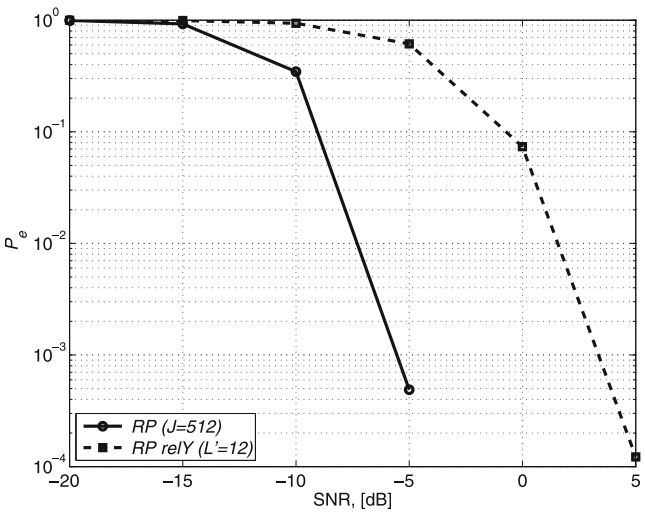

Figure 24 Probability of classification error of images versus SNR of AWGN noise.

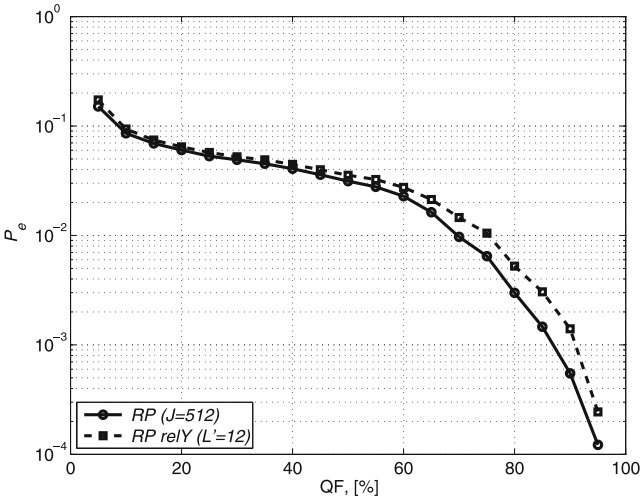

Figure 25 Probability of classification error of images versus JPEG quality factor.

5 Conclusions

We have considered the multiclass classification problem based on sets of independent binary classifiers. We have analyzed the properties of such kind of matrices and their impact on the maximum number of uniquely distinguishable classes from an information-theoretic point of view.

The relation between the reliability of bits due to projections and the bit error probability has been investigated and shown to be of crucial importance for the complexity and accuracy of classification. We demonstrate that it is equivalent to the considered random coding matrix without any bit reliability information in terms of recognition rate.

Several relatively low-complexity algorithms have been proposed for classification that approximate the accuracy of more time-consuming but optimal alternatives.

References

1. Allwein, E. L., Schapire, R. E., Singer, Y., & Kaelbling, P. (2000). Reducing multiclass to binary: A unifying approach for margin classifiers. *Journal of Machine Learning Research, 1*, 113–141.
2. Beekhof, F., Voloshynovskiy, S., Koval, O., & Holotyak, T. (2009). Fast identification algorithm for forensic applications. In *First IEEE international workshop on information forensics and security*. London, UK.
3. Cover, T., & Thomas, J. (1991). *Elements of information theory*. New York: Wiley.
4. Crammer, K., & Singer, Y. (2001). On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research, 2*, 265–292.

5. Dekel, O., & Singer, Y. (2002). Multiclass learning by probabilistic embeddings. In *In NIPS* (pp. 945–952). MIT Press.
6. Dietterich, T. G., & Bakiri, G. (1995). Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research, 2*, 263–286.
7. Escalera, S., Pujol, O., & Radeva, P. (2008). Loss-weighted decoding for error-correcting output coding. In *3rd int. conf. on computer vision theory and applications* (pp. 117–122). Madeira, Portugal.
8. Feller, W. (1968). *An introduction to probability theory and its applications* (Vol. I). New York: Wiley.
9. Forney, G. D. (1968). Exponential error bounds for erasure, list, and decision feedback schemes. *IEEE Transactions on Information Theory, 14*, 206–220.
10. Hsu, C. W., & Lin, C. J. (2002). A comparison of methods for multi-class support vector machines. *IEEE Transactions on Neural Networks, 2*, 415–425.
11. O'Sullivan, J. A., & Schmid, N. A. (2001). Performance analysis of physical signature authentication. *IEEE Transactions on Information Theory, 47*, 3034–3039.
12. Passerini, A., Pontil, M., & Frasconi, P. (2004). New results on error correcting output codes of kernel machines. *IEEE Transactions on Neural Networks, 1*, 45–54.

Oleksiy Koval received his M.S. Degree in electrical engineering from the National University Lvivska Politechnika, Lviv, Ukraine, in 1996. In 1996–2001, he was with the Department of Synthesis, Processing, and Identification of Images, Institute of Physics and Mechanics (Lviv, Ukraine) as a researcher and Ph.D. student. He received his Ph.D. degree in electrical engineering from the National University Lvivska Politechnika, in 2002. Since 2002, he has been with Stochastic Information Processing Group, University of Geneva, from which he received his Ph.D. degree in stochastic image modeling in 2004, where he is currently an Assistant Professor. His research interests cover stochastic image modeling for different image processing applications, digital watermarking, information theory, and communications with side information.

Sviatoslav Voloshynovskiy received his radio engineering degree from Lviv Polytechnic Institute in 1993, and a Ph.D. degree in electrical engineering from State University Lvivska Politechnika, Lviv, Ukraine, in 1996. In 1998–1999, he has been with the University of Illinois at Urbana-Champaign, USA, as a Visiting Scholar. Since 1999, he has been with University of Geneva, Switzerland, where he is currently an Associate Professor with the Department of Computer Science, and Head of the Stochastic Information Processing Group. His current research interests are in information- theoretic aspects of multimedia security based on digital data hiding and robust hashing, stochastic image modeling and machine learning. He has coauthored over 200 journal and conference papers in these areas as well as 10 patents. He has served as a consultant to private industry in the above areas.

Fokko Beekhof received an MSc from the technical University of Delft in 2004 for work on load-balancing in computational clusters, and then worked as a scientific C++-programmer on lattice Boltzmann simulations at the University of Geneva. From late 2006 until now, he has been pursuing a PhD in the SIP group of the University of Geneva. His main research interests are in robust hashing, identification, authentication and classification; and in particular the accuracy, complexity and privacy aspects of algorithms in these domains.

Taras Holotyak received MSc and PhD degrees in electrical engineering from the National University Lvivska Politechnika, Lviv, Ukraine, in 1997 and 2001. In 2003–2005, I have been with State University of New York at Binghamton, USA, as a postdoc. Since 2006, he has been pursuing a PhD in computer sciences with Stochastic Information Processing Group, Department of Computer Sciences, University of Geneva. His current research interests consist in privacy-preserving identification and authentication methods in large-scale databases. With respect to this framework he works on performance-complexity-privacy trade-off and investigates theoretical limits of such systems operation.