Ranking Error-Correcting Output Codes for Class Retrieval

Mehdi Mirza-Mohammadi, Francesco Ciompi, Sergio Escalera, Oriol Pujol, and Petia Radeva
Computer Vision Center, Campus UAB, Edifici O, 08193, Bellaterra, Spain UB
Dept. Matemàtica Aplicada i Anàlisi, UB, Gran Via de les Corts Catalanes 585, 08007, Barcelona, Spain

Abstract

Error-Correcting Output Codes (ECOC) is a general framework for combining binary classification in order to address the multi-class categorization problem. In this paper, we include contextual and semantic information in the decoding process of the ECOC framework, defining an ECOC-rank methodology. Altering the ECOC output values by means of the adjacency of classes based on features and class relations based on ontology, we defined a new methodology for class retrieval problems. Results over public data show performance improvement when using the new ECOC-rank in the retrieval process.

Keywords: Retrieval, Ranking, Error-Correcting Output Codes.

1 Introduction

Information Retrieval deals with uncertainty and vagueness in information systems (IR Specialist Group of German Informatics Society, 1991). This information could be in forms such as text, audio, or image. The science field which deals with information retrieval in images is called Content-based image retrieval (CBIR). CBIR corresponds to any technology that helps to organize digital picture archives by visual content. In this sense, any system ranging from an image similarity function to a robust image annotation engine falls under the purview of CBIR [1].

In last decade, many work has been performed to describe color, shape, and texture features, without considering image semantics. Most of these works are based on the retrieving of samples from the same category. However, in our case we address the class retrieval problem. Suppose we have an image from a cat animal category. In that case, we want to retrieve similar categories and not just samples from the same animal (e.g. tiger could be a possible solution). In order to deal with this problem, we can use on the output of multi-class classifiers to rank classes and perform class retrieval. In particular, we focus on the Error-Correcting output codes framework, which combines binary classifiers to address the multi-class problem.

Up to now, the ECOC framework has been just applied to the multi-class object recognition problem, where just one output label was required. Based on the ECOC framework, we extend this methodology to address class retrieval problems. Altering the ECOC output values by means of class adjacency matrix based on features and class relations within an ontology matrix, we alter the ECOC output ranking. This new ranking is used to look at the first retrieved classes to perform class retrieval. The results of the new ECOC-Rank approach show that performance improvements are obtained when including contextual and semantic information in the ranking process. The rest of the paper is organized as follow. In section 2 we define the ECOC rank and the proposed alteration methods. Section 3 presents experimental results. Finally, section 4 concludes the paper.

2 ECOC Rank

Retrieval systems retrieve huge amount of data for each query. Thus, sorting the results from most to less relevant cases is required. Based on the framework and application, there exists different ways for ranking the results based on the associated criteria.

In the decoding process of the ECOC framework [2], a "distance" associated to each class is computed. This "distance" can be then interpreted as a ranking measure. However, this ranking is the most trivial way for sorting the results. Moreover, the output of the ECOC system does not take into account any semantic relationship among classes, which may be beneficial for retrieval applications. As an example of an image retrieval system, suppose the query of "Dog". In the feature space, it is possible that there exists high similarity between "Dog" and "Bike", so based on features, the ranking will be higher for "Bike" than for some other class which can be semantically more similar to "Dog", such as "Cat". On the other hand, it is easy to see that similarity based on features also is important, and thus, a trade-off between appearance and semantics is required. In order to embed class semantic and contextual information in the ranking process, we define two matrices that will be used to vote the ranking process: one based on adjacency and another one based on ontology. These matrices are $n \times n$ matrices for n number of classes, where each entry represents the similarity between two classes. By multiplying the ranking vector of the ECOC output by these matrices, we alter the output ranking and improve retrieval results. The rest of this section describes the design of the class adjacency matrix, ontology matrix, and their use to modify the output ECOC rank.

2.1 Adjacency Matrix M_A

There are different approaches in literature for measuring the similarity between two classes. Support Vector Machines margin and the distance between cluster centroid are two common approaches. Here, we follow a method similar to the second approach. However, just considering the cluster centroid would not be an accurate criteria for non-gaussian data distributions. Instead, we re-cluster each class data into a few number of clusters and measure the mean distance of centroid of the new set of representant.

Since the objective is to alter the ranking, the defined adjacency matrix should be converted to a measure of likelihood, which means that the more two classes are similar, the more the new measure among them should be higher. Thus, we compute the inverse of the distance for each element and normalize each column of the matrix to one to give the same relevance to each of the classes similarities. The details of this procedure are described in algorithm 1.

Table 1: Adjacency Matrix M_A computation.

Given the class set $c = \{c_1, c_2, .., c_n\}$ and their associated data $W = \{W_{c_1}, .., W_{c_n}\}$ for n classes

For each c_i

1) Run k-means on W_{c_i} set and compute the cluster centroids for class c_i as $m_i = \{m_{i1}, .., m_{ik}\}$

Construct distance matrix M_D as follows:

For each pair of classes c_p **and** c_q

1) $M_D(p,q) = \frac{\sum_{i=1}^{k}\sum_{j=1}^{k}\delta(m_{pi}, m_{qj})}{\frac{k(k-1)}{2}}$, being δ a similarity function

Convert distance matrix M_D to adjacency matrix M_A as follows:

For each pair of classes c_p **and** c_q

1) $M_A(p,q) = \frac{1}{M_D(p,q)}$

Normalize each column p of M_A as follows:

1) $M_A(p,q) = \frac{M_A(p,q)}{\sum_{i=1}^{n} M_A(i,p)}$

Look at the toy problem of Figure 1. In the example, three representant are computed for each class using k-means. Then, the distance among all pairs of representant are computed for a pair of classes, obtaining an adjacency distance for that two classes as $M_D(1,3) = \frac{8+10+9+7+9+8+7.5+9.5+8.5}{9} = 8.5$. After that, the remaining positions of M_D are obtained in the same way, defining the following distance matrix

M_D:

$$M_D = \begin{pmatrix} 1 & 4 & 8.5 \\ 4 & 1 & 10 \\ 8.5 & 10 & 1 \end{pmatrix} \quad (1)$$

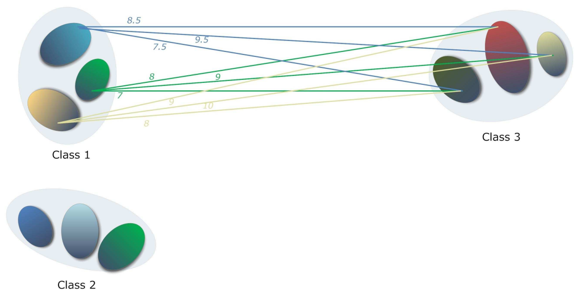

Figure 1: Toy problem for a 3-class classification task. For each class, three representant are computed using k-means. Then, the distance among all pairs of representant are computed for a pair of classes.

Finally, the adjacency matrix is computed changing distances to a likelihood values and normalizing each column of the matrix to unit. In this sense, the final adjacency matrix M_A for the toy problem of 1 is as follows:

$$M_A^{\text{Likelihood}} = \begin{pmatrix} 1 & 0.25 & 0.12 \\ 0.25 & 1 & 0.1 \\ 0.12 & 0.1 & 1 \end{pmatrix} \quad (2)$$

$$M_A = \begin{pmatrix} 0.73 & 0.18 & 0.08 \\ 0.19 & 0.74 & 0.07 \\ 0.09 & 0.08 & 0.81 \end{pmatrix} \quad (3)$$

2.2 Ontology Matrix M_O

The process up to here considered the relationship between classes by means of computational methods. However, some times no matter how good the system is, it can benefit of human knowledge. Here, we try to "inject" human knowledge of semantic similarity between classes into the system.

Taxonomy based on ontology is a tree or hierarchical classification which is organized by subtype-supertype relations. For example, Dog is a subtype of Animal. The authors of Caltech 256 data set compiled a taxonomy for all the categories included in their data set. Based on this taxonomy, we also defined a similar one for the MSRCORID data set, which will be used to validate our methodology in the results section. The taxonomy of the Caltech data set can be found in [3]. The taxonomy tree defined for MSRCORID is shown in Figure 2.

Here we try to construct a similarity matrix like we did for the adjacency matrix, but now the similarity of classes is computed by means of the taxonomy tree.

In order to compute the distance among classes based on taxonomy, we look for common ancestor of nodes within the tree. Each category is represented as a leaf, and the non-leaf vertices correspond to abstract objects or super-categories. The less distance of the two leafs to their common ancestor, the less is their ontology distance. We construct the similarity matrix by crawling the tree from a leaf and rank all other leaves based on their distance. When we start from each leaf and crawl up the tree, at each step the current node is being explored based on depth-first search algorithm. In this search the less depth leaves get higher rank.

Finally, like in the case of the adjacency matrix, we need to convert distances into a measure of likelihood by inverting the values, and normalizing each column of the ontology matrix M_O to give the same importance for the taxonomy of all the classes. The whole process of computing the taxonomy distance and the ontology matrix is explained in algorithm 2. Figure 3 shows a possible ontology distance computation for the toy problem of Figure 1.

The final ontology matrix M_O obtained after computing all ranks from ontology distance and likelihood computation are the followings:

$$M_O^{\text{Ranking}} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 2 & 1 & 3 & 4 & 5 & 6 \\ 4 & 3 & 1 & 2 & 5 & 6 \\ 4 & 3 & 2 & 1 & 5 & 6 \\ 5 & 2 & 3 & 4 & 1 & 6 \\ 6 & 3 & 4 & 5 & 2 & 1 \end{pmatrix} \quad (4)$$

$$M_O^{\text{L}} = \begin{pmatrix} 1.0000 & 0.5000 & 0.3333 & 0.2500 & 0.2000 & 0.1667 \\ 0.5000 & 1.0000 & 0.3333 & 0.2500 & 0.2000 & 0.1667 \\ 0.2500 & 0.3333 & 1.0000 & 0.5000 & 0.2000 & 0.1667 \\ 0.2500 & 0.3333 & 0.5000 & 1.0000 & 0.2000 & 0.1667 \\ 0.2000 & 0.5000 & 0.3333 & 0.2500 & 1.0000 & 0.1667 \\ 0.1667 & 0.3333 & 0.2500 & 0.2000 & 0.5000 & 1.0000 \end{pmatrix} \quad (5)$$

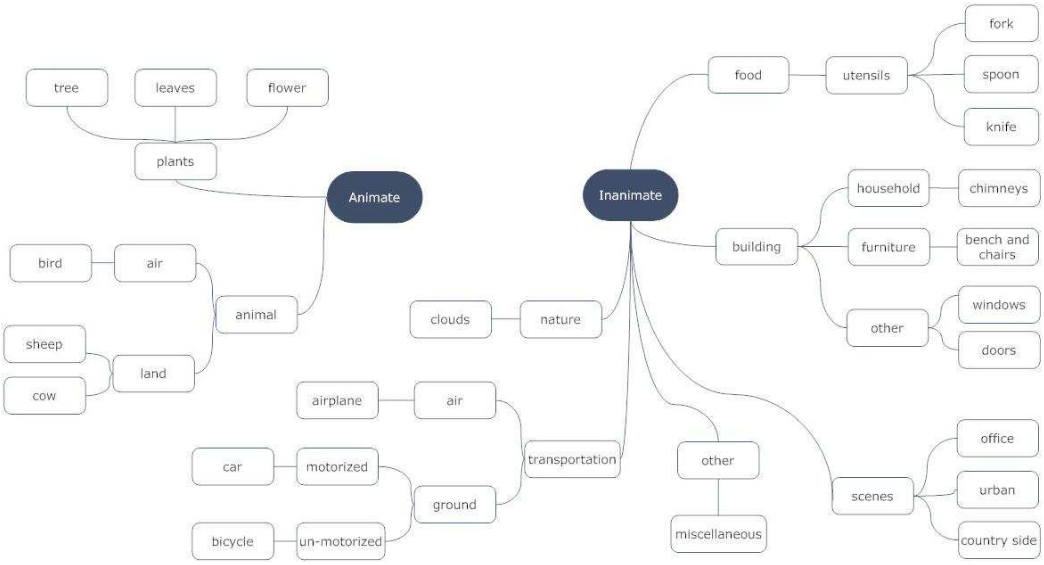

Figure 2: Taxonomy of object categories of the MSRCORID data set.

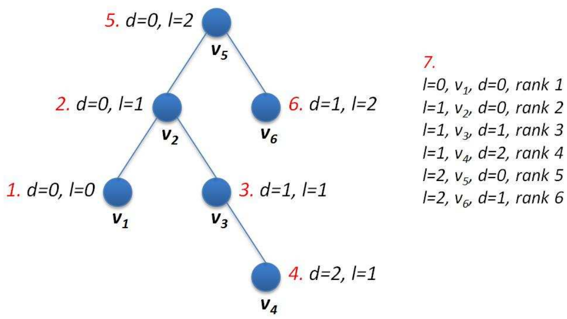

Figure 3: Example of the ontology distance computation of vertex v_1 to the rest of vertices. The steps of the distance computation are sorted. The final ranking is shown in the last step of the distance computation. This final ranking is then normalized and used as a ontology likelihood.

$$M_O = \begin{pmatrix} 0.4082 & 0.2041 & 0.1361 & 0.1020 & 0.0816 & 0.0680 \\ 0.2041 & 0.4082 & 0.1361 & 0.1020 & 0.0816 & 0.0680 \\ 0.1020 & 0.1361 & 0.4082 & 0.2041 & 0.0816 & 0.0680 \\ 0.1020 & 0.1361 & 0.2041 & 0.4082 & 0.0816 & 0.0680 \\ 0.0816 & 0.2041 & 0.1361 & 0.1020 & 0.4082 & 0.0680 \\ 0.0680 & 0.1361 & 0.1020 & 0.0816 & 0.2041 & 0.4082 \end{pmatrix} \quad (6)$$

2.3 Altering ECOC output rank using M_A and M_O

Given the output vector $D = \{d_1, .., d_n\}$ of the ECOC design, where d_i represents the distance of a test sample to codeword i of the coding matrix, first, we convert the vector D to a measure of likelihood by inverting each position of D as $D^L = \{\frac{1}{d_1}, .., \frac{1}{d_n}\}$, and normalizing the new vector so that $\sum_{i=1}^{n} D_i^L = 1$. Then, using the previous M_A and M_O matrices, the new altered rank R is obtained by means of a simple matrix multiplication, as follows:

$$R = D^L \cdot M_A \cdot M_O \quad (7)$$

3 Results

Before the presentation of the results, first, we discuss the data, methods and parameters, and validation protocol of the experiments.

Data: The data used in our experiments consists on two public data sets: Caltech 256 [4] and 'Microsoft Research Cambridge Object Recognition Image data set' [5].

Methods and parameters: We use the classical Bag-Of-Visual-Words model (BOVW) [6] of 50 visual words to describe the data sets using the Harris-Affine detector and SIFT descriptor. For the ECOC classification, One-versus-one method with Gentle Adaboost with 50 decision stumps and RBF

Table 2: Ontology Matrix M_O computation.

Given the class set $c = \{c_1, c_2, .., c_n\}$ and the taxonomy graph G

For each leaf vertex v_i in G, $i \in [1, .., n]$, where n is the number of classes

1) Visiting vertex $v_j = v_i$, Up Level $l = 0$, Depth $d = 0$

Position list for each vertex v_p: $M_P(v_p) = [L_{v_p}, D_{v_p}]$

where L_{v_p} is the level of v_p and D_{v_p} is the depth of v_p

2) **Do while there are unvisited vertices**

1) $VisitVertice(v_j)$

Function VisitVertice(v_p):
If v_p is not visited
visitChild(v_p)
if $\exists\, parent(v_p)$
$l = l + 1$
$M(v_p) = [l, d]$
$VisitVertice(parent(v_p))$

Function VisitChild(v_p):
for each child v_p^c of v_p:
if v_p^c has not been visited:
if child(v_p^c) $! = \varnothing$
VisitChild(v_p)
else
$d = d + 1$
$M(v_p) = [l, d]$

3) Filling the ranks
$r = 0$
for $\nu = [1, .., \max(l)]$
for $\omega = [1, .., \max(d)]$
if $v_q | M_P(v_q) = [\nu, \omega]$ is a leaf vertex of G
$M_O(i, q) = r$
$r = r + 1$

Convert distance matrix M_D to ontology matrix M_O as follows:

For each pair of classes c_p and c_q
1) $M_O(p, q) = \frac{1}{M_O(p,q)}$

Normalize each column p of M_O as follows:
1) $M_O(p, q) = \frac{M_O(p,q)}{\sum_{i=1}^{n} M_O(i,p)}$

Support Vector Machines with parameters $C = 1$ and $\sigma = 0.5$ have been used. We use the Linear Loss-weighted decoding to obtain the class label [7]. For the adjacency matrix construction, the k parameter of k-means has been experimentally set to 3. For ranking the hist count we looked for one to seven matches at the first 15 positions using vector ontology and semantic distances of 0.001 and 0.0001.

Validation measurements: In order to analyze the retrieval efficiency, we defined an ontology distance based on taxonomy trees to look for the retrieved classes at the first positions of the ranking process. As explained in the previous section, the ranking result R is a sorted set of classes, where the first items have the highest rank. Then, we define an ontology distance m based on the taxonomy tree and adjacency matrices. Each c_i in R is accepted if its ontology distance d_i compared to the true label class is less than m. The accepted results in the end of the list R are not desired, so another parameter k is used to analyze the results of the first positions of the ranking. If there are more than N accepted classes based on the value of m at the first positions defined by k, then we achieve a test hit. In order to perform a realistic analysis, we included this validation procedure in a stratified 10-fold evaluation procedure. The algorithm that summarizes the retrieval validation is shown in table 3.

3.1 Caltech 256 retrieval evaluation

In this case, we have defined an ontology distance of 0.001 and 0.0001 for Adaboost ECOC base classifier based on the taxonomy tree and the ontology distance defined in previous sections. For both distances we computed the BOVW features for this data set with different values of k first positions and number of hits. Some obtained performance surfaces are shown in Figure 4. The performances are also shown in Table 4 estimated as the mean performance surface for each experiment. Note that we compared the classical ECOC output (Raw) with the ranking alteration using the adjacency matrix, ontology matrix, and both. In this case, the best results are obtained just altering the ECOC output by the ontology matrix.

Table 3: ECOC-Rank evaluation.

Given the sorted list of classes based on their rank $R = \{r_1, .., r_n\}$
For each item r_i in the top k positions of R
$acceptedCount = 0$
1) $d = OntologyDistance(r_i, TrueLabel)$
2) if $d > m$ then $acceptedCount+ = 1$
1) If $acceptedCount > N$ then Hit

Table 4: Performances of Caltech 256 data set for different methods and parameters using Gentle Adaboost ECOC base classifier and ontology distance evaluation.

Problem	Adjacency	Ontology	Adj & Ont	Raw
m=0.001	0.4394	0.6901	0.4389	0.5530
m=0.0001	0.0718	0.1479	0.0719	0.0785

3.2 Microsoft Research Cambridge Object Recognition Image data set

In this case, we have defined an ontology distance of 0.001 and 0.0001 for Adaboost and RBF SVM ECOC base classifiers based on the taxonomy tree and the ontology distance defined in previous sections. For both distances we computed the BOVW features for this data set with different values of k first positions and number of hits. A sample of results are shown in the performance surfaces of Figure 5 and Figure 6 for Adaboost and SVM, respectively. The performances are also shown in Table 5 estimated as the mean performance surface for each experiment. In this experiment, though most of the experiments improve the classical ECOC rank, the adjacency matrix is selected as the first choice.

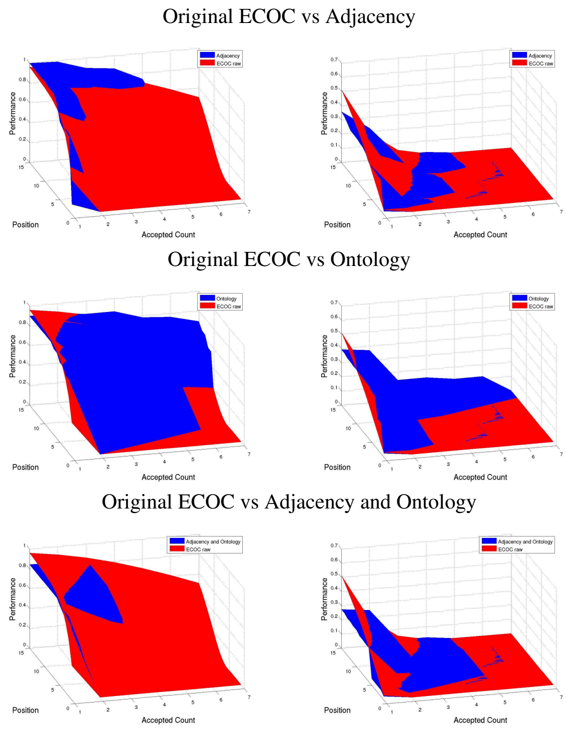

Figure 4: Results on Caltech 256 data set for Gentle Adaboost ECOC base classifier. Left column using ontology distance m=0.001 and right column using m=0.0001.

Table 5: Performances of Microsoft Research Cambridge Object Recognition Image data set for different methods and parameters using Gentle Adaboost ECOC base classifier and ontology distance evaluation.

Problem	Adjacency	Ontology	Adj & Ont	Raw
ADA m=0.001	0.3154	0.1744	0.2996	0.1568
ADA m=0.0001	0.1777	0.0659	0.1576	0.0667
SVM m=0.001	0.3714	0.1798	0.3001	0.2038
SVM m=0.0001	0.2511	0.0676	0.1577	0.0950

4 Conclusion

In this paper we altered the decoding process of the ECOC framework to define a new measure of semantic ranking that is applied on class retrieval problems. In order to include contextual and semantic information, we defined two matrices that mutates the ECOC output. An adjacency matrix is defined based on the feature space, and an ontology matrix is designed based on taxonomy trees. Results over public data show performance improvement when using the new ECOC-rank in the re-

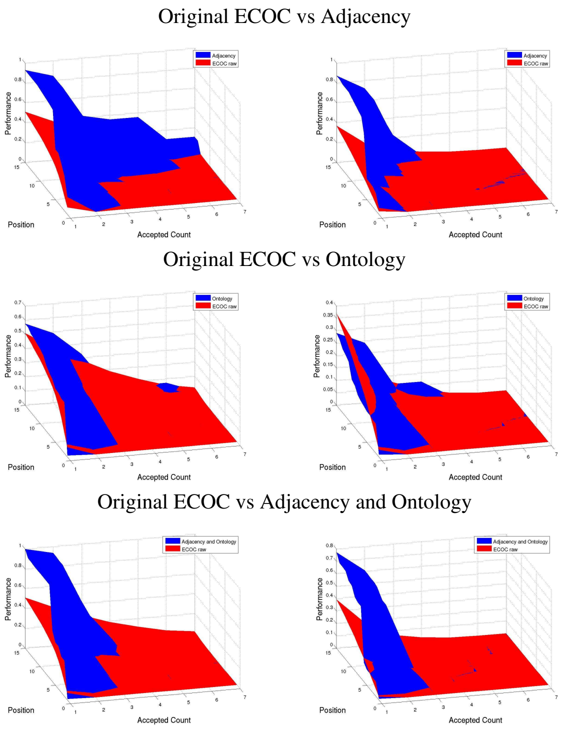

Figure 5: Results on Microsoft Research Cambridge Object Recognition Image data set for Gentle Adaboost ECOC base classifier. Left column using ontology distance m=0.001 and right column using m=0.0001.

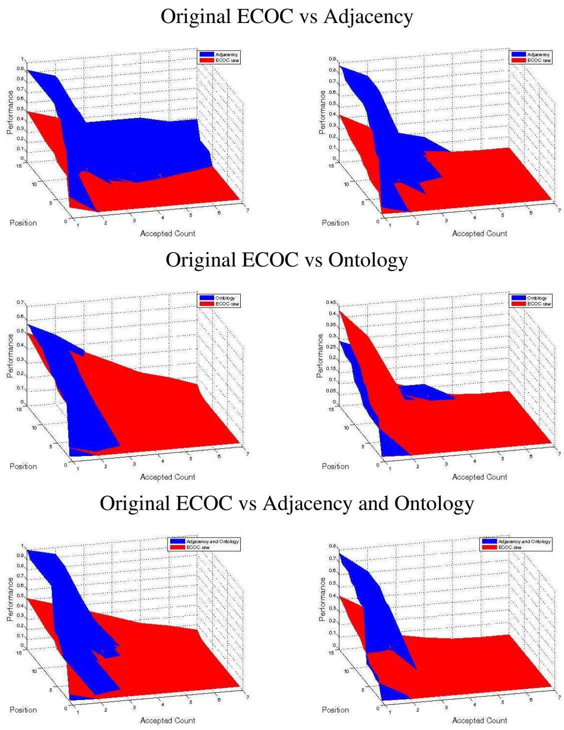

Figure 6: Results on Microsoft Research Cambridge Object Recognition Image data set for RBF SVM ECOC base classifier. Left column using ontology distance m=0.001 and right column using m=0.0001.

trieval process.

5 Acknowledgement

This work has been partially supported by the projects TIN2006-15694-C02 and CONSOLIDER-INGENIO 2010 (CSD2007-00018).

References

[1] R. Datta, D. Joshi, J. Li, J. Z. Wang, Image retrieval: Ideas, influences, and trends of the new age, ACM Comput. Surv. 40 (2) (2008) 1–60.

[2] T. Dietterich, G. Bakiri, Solving multiclass learning problems via error-correcting output codes 2 (1995) 263–282.

[3] G. Griffin, A. Holub, P. Perona, Caltech-256 object category dataset, Technical Report, California Institute of Technology.

[4] G. Griffin, A. Holub, P. Perona, Caltech-256 object category dataset, Tech. Rep. 7694, California Institute of Technology (2007).
URL `authors.library.caltech.edu/`

[5] Microsoft research cambridge object recognition image database.
URL `http://research.microsoft.com/`

[6] G. Csurka, C. Dance, L. Fan, J. Willamowski, C. Bray, Visual categorization with bags of keypoints, in: ECCV, 2004, pp. 1–22.

[7] S. Escalera, O. Pujol, P. Radeva, On the decoding process in ternary error-correcting output codes, in: PAMI, 2009.