UPMC/LIP6 at ImageCLEFannotation 2009: Large Scale Visual Concept Detection and Annotation

Ali Fakeri-Tabrizi, Sabrina Tollari, Ludovic Denoyer, Patrick Gallinari
Université Pierre et Marie Curie - Paris 6
Laboratoire d'Informatique de Paris 6 - UMR CNRS 7606
104 avenue du président Kennedy, 75016 Paris, France
firstname.lastname@lip6.fr

Abstract

In this paper, we present the LIP6 annotation models for the ImageCLEF2009annotation task. We focus on two challenges: 1-the large scale annotation and 2-the application of an ontology. We propose two models of classifiers for the first challenge. In the case of second challenge, we try to detect the relations between the concepts via a cooccurrence matrix. By this way, we avoid the exclusive concepts tagged to one image.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: H.3.1 Content Analysis and Indexing; H.3.3 Information Search and Retrieval; H.3.4 Systems and Software; H.3.7 Digital Libraries

General Terms

Measurement, Performance, Experimentation

Keywords

SVM, Graph models, Cooccurrences Analysis, Multi-Class Multi-Label Image Classification, Visual Concepts

1 Introduction

In ImageCLEFannotation 2009 [2], there are two main challenges: the large scale annotation and the application of an ontology (hierarchies and relations) for improving the image annotation.In the first challenge, we encounter the imbalanced data set. We propose for the second challenge to reduce the score of the concept, with the lowest score, when two concepts are exclusive.

The application of our methods is compared by two classifications techniques, SVM and graph classification. SVM is used as a common classifier to give us a baseline for comparison with the Graph classification technic. As an other classifier, with the given image data set, we create a graph. The labels of different classes are propagated through this graph and at the end, we have the classification of different parts of graph.

In the section 2, we propose our models for image annotation adapted for imbalanced data set problem. In the section 3, we show a method for considering the effects of the existing hierarchy between the concepts, on the results of classifiers. The experiences are illustrated in section 4. The conclusion and perspectives will be presented in section 5.

2 Annotation models

2.1 Imbalanced class problem

Machine learning algorithms usually assume that the training set is well-balanced. Therefore they suppose that all misclassification errors have an equal cost. But data in real world is often imbalanced. It means that some classes have much more examples than the others. We can call the classes which have more examples, the majority classes and the ones which have fewer examples, the minority classes. Using overall accuracy in this case is not an appropriate evaluation measurement because of the dominating effect of the majority classes.

In the given data set, we have 53 classes. For each class, we have obviously different numbers of positive examples. For some classes, we have such few numbers of positives (negatives) that they will be dominated by the majority classes. It means we have not a balance between the majority and minority classes to learn a model. Therefore we have an imbalanced class problem which should be solved to have the good and reasonable classifications.

To solve this problem two strategies are performed:

1. Removing the examples from the majority set: in this method, we select the same number of examples from both the minority classes and the majority classes. The examples which are selected from the majority classes are chosen randomly. In this way, we will have all the examples of the minority class and in fact we remove many examples of the majority class. Therefore the number of positive and negative examples will be balanced for each class.

2. Choose a convenient loss function for the classifier:

   We define the cost of ignorance in the minority class higher than in the majority class. Then the classifier will be instigated to find a better model. So we have a loss function in which the misclassification's cost depends on the number of class members.

2.2 SVM

Given an implicit embedding ϕ and training data (x_i, y_i) from 2 classes such that $y_i \in \{-1, 1\}$, a Support Vector Machine finds a hyperplane $w^T\phi(x) + b = 0$ that separates the two classes as well as possible. The learnt hyperplane is optimal in the sense that it maximises the margin while minimizing some measures of loss on the training data.

More formally, the primal formulation of SVM is

$$Min_{w,\xi} \quad \frac{1}{2}w^t w + C1^t\xi$$

subject to $y_i(w\phi(x_i) + b) \geq 1 - \xi_i$ and $\xi \geq 0$ where $\phi(x_i)\phi(x_j) = K(x_i, x_j)$ with C being a user specified misclassification penalty.[1]

We have run a SVM classifier[2], which uses a linear kernel, on the training base to learn a model for the given corpus. Therefore, a basic classifier will be obtained by this model. As explained above, we have the imbalanced class problem. Here we use a ROC area as the loss function as proposed in [1]. So we consider not only the misclassification in each learning iteration, but also the number of positive and negative examples in order to avoid the fault ignorance. ROCarea can be computed from the number of swapped pairs

$$SwappedPaires = \|\{\ (i,j) : (y_i > y_j) and (w^T x_i < w^T x_j)\}\|$$

i.e. the number of pairs of examples that are in the wrong order.

$$ROCarea = 1 - \frac{SwappedPaires}{\#pos.\#neg}$$

Here 1-ROCarea is used as the value of misclassification in loss function for each iteration.

[1] `http://research.microsoft.com/en-us/um/people/manik/projects/trade-off/svm.html`

[2] `http://svmlight.joachims.org/svm_perf.html`

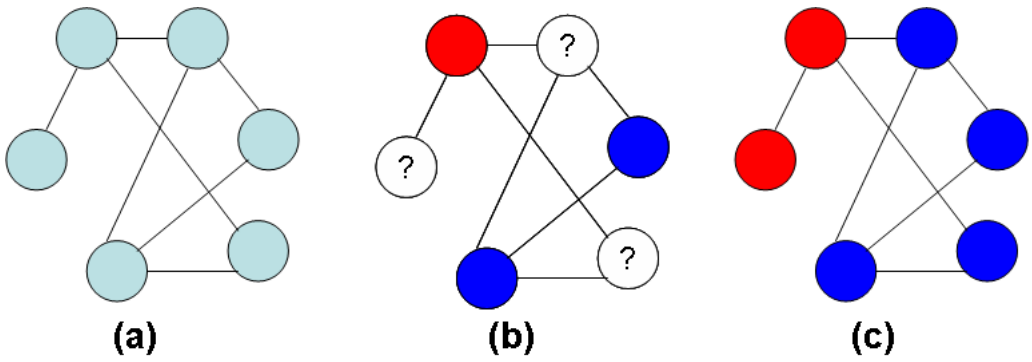

Figure 1: Semi supervised graph labelling task. The goal is to label the unlabelled nodes of a partially labeled graph. (a): Images graph: Each node represents an image, each edge represents a weighted similarity between images (b): Labeled images (colored nodes) and unlabeled images (non-colored nodes) in the same graph (c): The output is a fully labeled graph

2.3 Graph Classification

A graph is created in which the nodes represent the images. Each node is whether annotated by many concepts (images in training base) or is not annotated yet (images in test base). The idea is to classify the nodes which represent the images. The similarity value between each pair of nodes is calculated. This value will be the weight of the edge between that pair of nodes. In this study the similarity value is based on the Euclidean distance:

$$Sim(X,Y) = e^{-D(X,Y)}$$

in which D represents the euclidean distance.

We propagate the labels through the graph.[4] The propagation process is done through the edges weighted by similarity values. This is the reason why we can have the same labels for the similar images; i.e the propagation through an edge with a high value will be stronger than one with a low value.

We formulate this process as below:
$\mathcal{G} = \sum_{x_i \in X} \Delta(f_\theta(x_i), y_i) + \alpha \sum_E g_w(i,j)(f(x_i) - f(x_j))^2 + \beta \sum_E \Delta'(g_w(i,j), y(i,j))$
in which

- $\sum_{x_i \in X} \Delta(f_\theta(x_i), y_i)$ is the error of wrong labeling;
- $\alpha \sum_E g_w(i,j)(f(x_i) - f(x_j))^2$ is the factor for penalizing edges between the nodes with different labels. g_w is the predicted value for the weight of the edge.
- $\beta \sum_E \Delta'(g_w(i,j), y(i,j))$ penalize the classification errors on the edges weighted by g_w in compare with the existing edges in reality.

The goal is to minimize the $\mathcal{G}$.

3 Hierarchy relations between concepts

In this study, a tag collection consists of 53 classes. For each image, a subset of the collection will be assigned. For each image, we perform a hierarchical filter on its labels' prediction. The ontological relations between the concepts are considered by this filter. The idea of using this filter is to minimize the probable conflicts. With the hypothesis of the independency and negliging the logical relationships between the classes, we can annotate each image by the results obtained from the classifiers. We suppose that a classifier predicts two classes X and Y as the tags for an image.

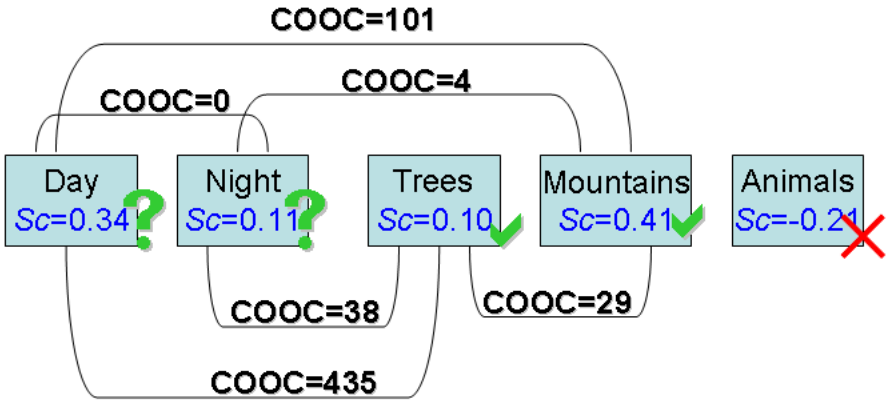

Figure 2: An example of hierarchy effects on results obtained by the classifier; *COOC* represents the value of cooccurrence between two classes. *Sc* is the real value of the class predicted by the classifier.

If the logical relationship between X and Y is negligible, we will have the same possibility for both classes to be tagged to the image. As we do not suppose any relation between X and Y, it seems to be reasonable. But if we take into account the relations between the classes, we may improve the results. As an example, if we consider X='DAY' and Y='NIGHT', then the hypothesis of appearing the both classes in same time will be contradictory. In the contradictory cases such as the example we mentioned above, the ontological or logical relations between classes will help us to improve the results by correction the misclassification.

This improvement will be done by adding more information which did not take into account during the learning process. We used the same method which proposed in [3]. In figure 2, we illustrate an example in which we suppose to have 5 classes that are predicted for an image. The prediction is done by the score values given by the classifier: for each class, we have a real value Sc_i between -1 and 1 as the output. Normally, a sign function will be performed on Sc_i to detect the class of the image. The figure 2 illustrates the class 'Animals' with a negative value which means that the image will not be assigned by this tag. The 4 other tags have the positive values. These positives values are not sufficient to annotate their corresponding tags to the image. As we see 'Day' and 'Night' are not compatible in the same time. To find these contradictions, we create a co-occurrence matrix which based on the concepts. This matrix will consider the degree of relation's strength between each concept:

$$COOC(X, Y) = \|I_X \cap I_Y\|$$

in which I_Z represents the set of images which are tagged by Z.

By considering the less values (not so much related) in this matrix, we can interpret the classes which are contradictories, i.e. they can not be presented at the same time as an image annotation. In figure 2, we find out that the weak value for COOC means a problem. It means that we can automatically find the contradictions. Now the question will be posed is: Which one? 'Day' or 'Night'. It is simple. We annotate the image by the class with stronger value and the other class will be eliminated. In this case, we save 'Day' with score value 0.34 and we remove 'night' with 0.11.

In addition, we define an optimism rate (OR), which signifies an acceptation degree of negative values. By this method, we expect to trap the misclassified examples. If we suppose OR=0, it will be the normal case. This means, the classifier is surely good enough and we don't expect to retrieve the hidden misclassifications. If OR=-0.1, all of the scores with a value more than -0.1 will be associated to positive class. By this method we decrease the risks of misclassifications and of having the contradictory classes. The algorithm 1 indicates this approach:

Algorithm 1 Hierarchy

1: For each image I in test set, Get labels score set S which is predicted by classifiers: $S = \{Sc_i\}_{i=1..53}$ in which $Sc_i \in [-1, 1]$ is the score predicted by the classifier for label Sc_i.
2: Sort S descending
3: $k \leftarrow 1$
4: WHILE $Sc_k \geq OR$
5: $\forall t > k$, IF $COOC(c_k, c_t) = 0$ THEN
$Sc_t \leftarrow Sc_t + 2 * OR$
resort Sc descending
$k \leftarrow k + 1$
ENDIF
ENDWHILE

Step1	Spring -0.08	Night 0.01	Animals -0.21	Trees 0.19	Mountains 0.41	Day 0.34
Step2	Mountains 0.41	Day 0.34	Trees 0.19	Night 0.01	Spring -0.08	Animals -0.21
Step3	**Mountains** **0.41**	Day 0.34	Trees 0.19	Night 0.01	Spring -0.08	Animals -0.21
Step4	**Mountains** **0.41**	**Day** **0.34**	Trees 0.19	Spring -0.08	Night -0.19	Animals -0.21
Step5	**Mountains** **0.41**	**Day** **0.34**	**Trees** **0.19**	Spring -0.08	Night -0.19	Animals -0.21
Step6	**Mountains** **0.41**	**Day** **0.34**	**Trees** **0.19**	**Spring** **-0.08**	Night -0.19	Animals -0.21
Step7	**Mountains** **0.41**	**Day** **0.34**	**Trees** **0.19**	**Spring** **-0.08**	*Night* *-0.19*	*Animals* *-0.21*

Table 1: An example of hierarchy effects on results obtained by the classifier; *COOC* represents the value of co-occurrence between two classes. *Sc* is the real value of the class predicted by the classifier.

We used this logic to estimate the misclassifications. Note that in this approach, we try to modify the output of our learning model and not the proper learning model. Table 1 illustrates how this method works. Here we have 6 different candidate topics to be tagged to a given image.

We have the cooccurrence non-zero value between all pairs of the concepts except for the pair (Day,Night). When an exclusivity is detected, the less score will be subtracted by 2 times of value of OR. We continue until having the score greater than OR.

4 Experiments

We have a training base including 5000 images of Flickr and a test base of 13000 images of Flickr. Each image is tagged by one or more of the 53 given concepts.

4.1 Visual features

As we need the features to learn, we use the color-based visual descriptors in this paper. Therefore we segment the images into 3 horizontal regions with the same sizes. For each region, we compute a color histogram in the HSV space. We believe that these visual descriptors are particularly interesting for general concepts (i.e. not objects), as for instance: sky, sunny, vegetation, sea... (see figure 3).

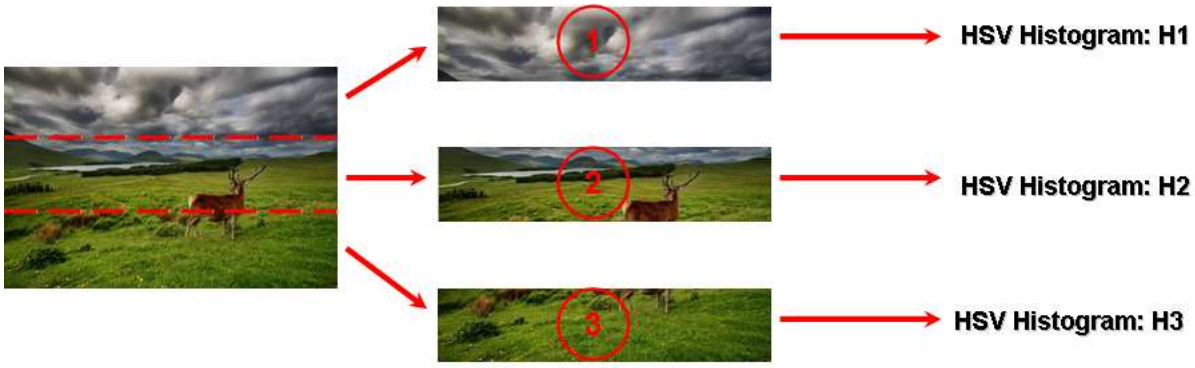

Figure 3: Dividing image on three horizontal segment to extract the histogram (HSV) of each part

4.2 Submitted Runs

We have performed the SVM classification (run1) and applied 3 different OR values. OR=0 (run2), OR=-0.1 (run3) and OR=-0.2 (run5). We consider also a classification without hierarchy (run1) to study the effects of hierarchy and its probable improvements. As a new approach, we had run a graph classifier and applied the hierarchy on it (run4).

Run1-LIP6RUN1 We used exclusively visual information here. Each image was cut in 3 horizontal regions with the same size and for each part, the visual features were extracted by HSV histogram. As a result, for each image, we had a vector in 51 dimensions (3*17).

Run2-LIP6RUN2 We used the same visual information as the run1 but the prediction obtained in this level is modified by the hierarchy method. We consider OR=1.0 which means we expect to find the misclassification till the scores -0.1. Thereby we try to apply the hierarchy to remove the contradictions.

Run3-LIP6RUN3 The same thing as Run2 but with value -0.15 for OR.

Run4-LIP6RUN4 Here we have used the PCA of the images as the visual information. We run the graph classification method on these data and the obtained prediction is modified by the hierarchy method with the value -0.1 for OR.

Run5-LIP6RUN5 The same thing as Run2 but with value -0.2 for OR.

4.3 Results and discussion

Table 2 shows the official results. We can notice that the best results are obtained with the SVM classifieur without using the hierarchical relations between concepts. Our run based on graph classification obtains lower scores than the random run, this means that we should have make a mistake building this run. The more OR is low, the more the EER and the AUC are low, but we can notice that the average annotation scores do not change varying OR values. The average annotation scores for the last 4 runs are very lower than the scores of the random run, this should mean that we may inverse the applications of the OR filter.

Run	EER	AUC	Average Annotation Score with Annotator Agreement	without Annotator Agreement
SVM	0.372	0.673	0.445	0.414
SVM+Hierarchy:OR=-0.1	0.384	0.651	0.261	0.239
SVM+Hierarchy:OR=-0.15	0.407	0.630	0.262	0.240
SVM+Hierarchy:OR=-0.2	0.410	0.623	0.261	0.240
GraphClassifier+Hierarchy:OR=-0.1	0.498	0.192	0.260	0.238
Random	0.500	0.499	0.384	0.351

Table 2: Official results. All runs are fully automatic.

5 Conclusion

In this working note, we performed two classifiers to annotate the unlabeled images: SVM and Graph. We had an imbalanced data set and we tried to solve this problem. Graph classification did not give us the good results. We will study this method more in our future works. We tried to use the ontological relations between the concepts to improve the results of the classifiers. The experiences show that the application of ontology did not give better results. We might modify the hierarchy method in order to have more improvement.

As perspectives, we will change the first method of balancing the data. We will try to add the examples of minority class in the data-set still having the same number of minorities and majorities. Therefore, we expect to decrease the information lost. We will more study the effects of OR and we will try the method of hierarchy without OR to find out if it is efficient in reality or has better effects. In hierarchy method, we may be able to detect the other types of relations; for example, we expect to find the generalization relations between the concepts and by that we can modify the results of classifiers.In graph classification, we will perform a multi-classes model to find out if it will be better than 53 binary models.

Acknowledgment

This work was partially supported by the French National Agency of Research (ANR-06-MDCA-002 AVEIR project).

We thank Nicolas Usunier for his help with loss fucntion definition in SVM classification.

References

[1] Thorsten Joachims. A support vector method for multivariate performance measures. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2005.

[2] Stefanie Nowak and Peter Dunker. Visual concept detection and annotation task. In *CLEF working notes 2009*, 2009.

[3] Sabrina Tollari, Marcin Detyniecki, Ali Fakeri-Tabrizi, Christophe Marsala, Massih-Reza Amini, and Patrick Gallinari. Using visual concepts and fast visual diversity to improve image retrieval. In *Evaluating Systems for Multilingual and Multimodal Information Access – 9th Workshop of the Cross-Language Evaluation Forum*, 2008.

[4] Huang J. Zhou D. and Schlkopf B. Learning from labeled and unlabeled data on a directed graph. In *Proceedings of the 22nd International Conference on Machine Learning*, 2005.