AN ABSTRACT OF THE PROJECT REPORT OF

Jafer Almuallim for the degree of Master of Science in Computer Science presented on June 12, 2013.

Title: Classifier Chains for Multi-Label Classification with Incomplete Labels

Many methods have been explored in the literature of multi-label learning, ranging from simple problem transformation to more complex method that capture correlation among labels. However, mostly all existing works do not address the challenge with incomplete label data. The goal of this project is to extend the work of ensemble classifier chain to learn models using training examples with incomplete label assignment. This scenario is highly expected in many real-world application. For example, in image annotation, a user provides partial tags, or label assignment, for the image. We propose a new method that consider the multi-label learning problem in which portion of label assignment is missing. A further evaluation is covered in this project to study the effect of different parameters accompany this approach.

Classifier Chains for Multi-Label Classification with Incomplete Labels

by

Jafer Almuallim

A PROJECT REPORT

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Presented June 12, 2013
Commencement June 2013

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1: Introduction

Traditional classification problems aim to associate each data instance with a single class label. Many real-life problems, however, require further generalized setting where each data instance can be associated with multiple target. In the past, multi-label classification has mainly engaged the attention of researchers working on document categorization as each instance of a document collection usually belong to more than one semantic category [13, 6, 11]. Recently, multi-label classification methods have gained an increasing attention in real-world problems such as music categorization and text classification. An instance can be associated with different subjects; in scene classification, each scene may belong to several semantic classes [11, 4, 1].

One approach to solve multi-label problem is to decompose it into multiple independent binary classification problems, in which each label can have a separate binary model with its independent parameters optimization process.

Another approach to solve multi-label problem setting is to perform problem transformation where a multi-label problem is transformed into one or more single-label problems. However, these methods do not take into account the inherent relationship between multiple labels. Previous work has been devoted toward developing new methods to better capture the correlation between labels. Read [10], for instance, developed a new method that demonstrated high predictive performance as compared to direct transformation or decomposition approaches.

In many real-life problems, however, a complete labeled data is not available. For example, a person labeling images may leave some labels incomplete for the some instance. This becomes a challenge when using existing methods.

In this project, we extend the work on ensemble classifier chains for multi label classification to further investigate, develop and evaluate the classifier chain framework in presence with incomplete labelling.

Chapter 2: Related Work

In this section, we present details about the existing literature of multi-label problem.

2.1 Problem Decomposition

A straightforward approach to multi-label problem is to decompose the problem into several binary classification, each for one label. Binary relevance (BR), one of the most famous multi-label learning methods in the literature, applies a single binary model for each label of independently from the rest of the labels. It has a linear complexity with respect to the number of labels. One main disadvantage of this method is the ignorance of the fact that information of one label may be helpful for prediction of another related label.

It has been shown that optimal predictive performance can only be achieved by methods that explicitly take label correlation into account [2].

One approach to take label into consideration is by learning a second (or meta) level of binary models. Basically, it considers output of first level prediction as an input for the second level binary model. Godbole and Sarawagi stacked BR classification output along with the full original attribute space creating a second level classification process [4]. It is referred as 2BR. Variations of 2BR has also shown improvement in accuracy compared to BR [8]. However, using additional level of classification process in 2BR requires extra iteration on both training and testing.

2.2 Problem Transformation

A common approach to multi-label classification is to apply problem transformation, where a multi-label problem is transformed into one or more single-label problems. One popular problem transformation is the binary pairwise classification (PW). A binary model is used for each pair of labels. The prediction results in a set of pairwise references instead. Thus a further step is taken to use ranking schemes as has been done in [3].

Although PW perform well in several domain, it faces quadratic complexity in term of the number of labels. It becomes intractable for large problems [3].

Another well known problem transformation method is label combination, or power-set method (LC). A multi-label problem is transformed into a multi-class single-label problem by converting label set as atomic labels [1]. Although it can model label correlations in the training data, computational complexity goes exponential with the number of labels [12].

2.3 Two Dimensional Multi-label Learner (2DAL)

Two Dimensional Multi-label Active Learning [7] is a remarkable work that has been developed to address the multi-label setting problem that also considers the label correlation. In their work, they derived a multi-label Bayesian Error Bound when a sample-label pair is selected under multi-label setting. The prediction takes into consideration the use of relationship between mutual information and the entropy between labels. They argue that only a part of more effective labels are necessary to be annotated while others can be inferred by exploring the correlations among the labels. The reason is that the contributions of different labels to minimizing the classication error are different due to the inherent label correlations.

Chapter 3: The Classifier Chains Model (CC)

Classifier Chains is one approach of problem transformation in multi-label classification that was recently proposed by Read [10]. Their approach works as follows: One classifier h_i for each label in L. This seems very similar to binary relevance method, but it is different in the sense that the feature space for each binary model is extended with the output predicted for all previous classifiers, thus forming a classifier chain.

Given a new instance x to be classified, we first predict the relevance label y_1 using h_1. Next, h_2 predicts y_2 given x plus the predicted value y_1. That is, h_i predict y_i using x plus $y_1, ..., y_{i-1}$ as additional input information. Figure 1 illustrate the difference between BR and CC.

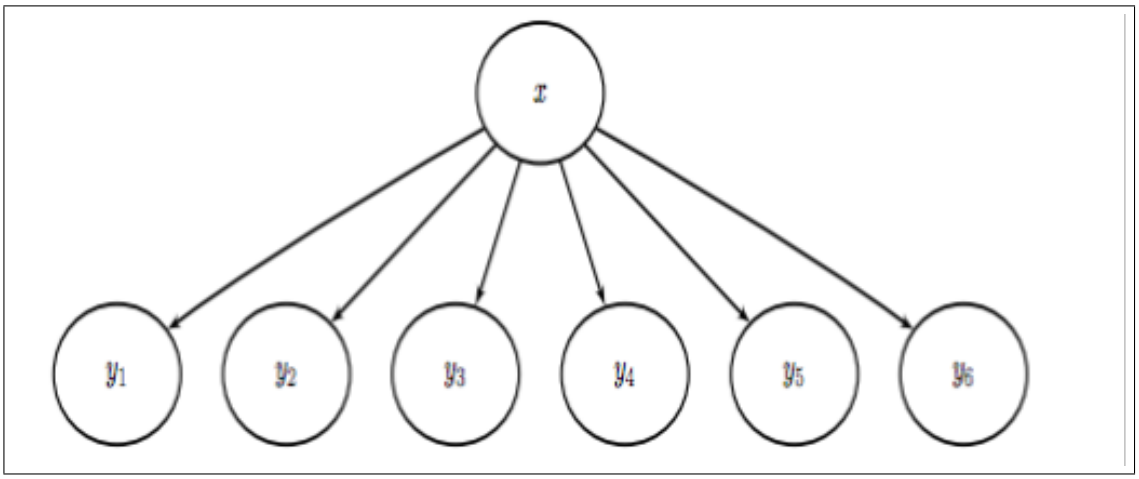

Figure 3.1: Binary Relevance.[9]

By passing all previous label information in the attributes, it will allow classifiers to take into account correlations between the labels. This present a great improvement over BR by considering the relationship between labels. Base classifiers can be more predictive when strong correlation exist in the label space.

Unlike meta-classifier (MBR), or 2BR, where an additional layer is needed for stacking approach, CC is very close to BR in term of computational complexity. On average, only

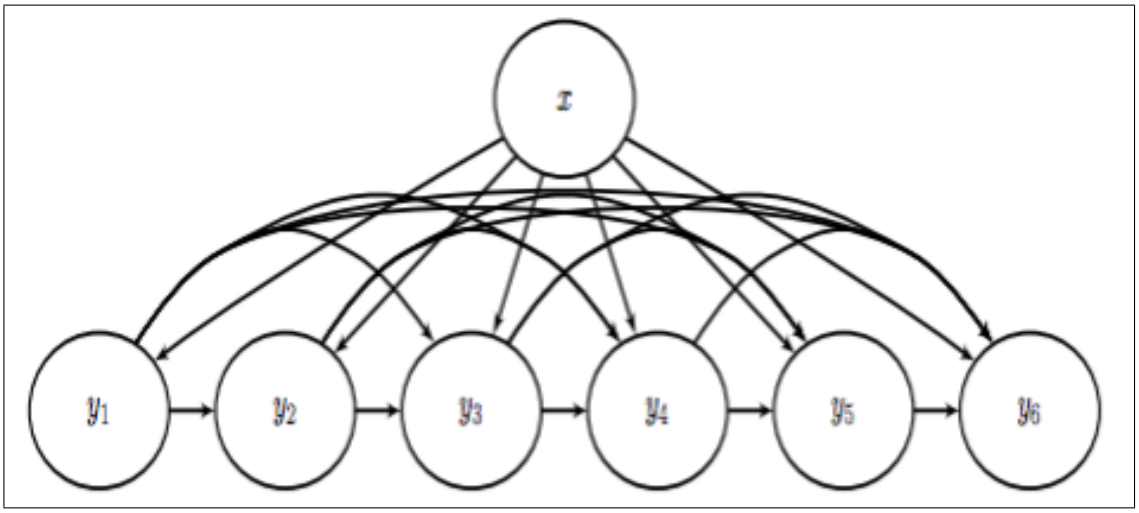

Figure 3.2: Classifier Chain.[9]

$L/2$ attributes are added to each instance and the label space is constant.

The order of chain, or the order of the label vector, plays an in important role in this setting. A different set label order may result in a greater influence on predictive performance of the classifier. Classifier Chains, however, does not make any effort to optimize the order of the label space.

Cheng [2] expended the work of CC by formulating a probabilistic interpretation. Based on probability theory, Bayes-optimal probabilistic classifier chain (PCC), an optimum prediction can be obtained by minimising the risk of using different ordering of the chain in the label space. Although PCC can produce an optimal classifier, it is computationally expensive. Given L label, PCC requires 2^L possible combinations. This makes PCC limited in practice as it was indicated by Cheng [10] to 10-15 labels. CC, in the other hand, needs to consider only random, or default, ordering of the chain. This random, or default, order may be a poor ordering compare to other variable chains. In the next section, we discuss a solution to this problem.

Chapter 4: Ensemble Classifier Chain (ECC)

In order to overcome the problem with poorly ordered chains, several CC classifiers can be trained with random order of chains. Basically, rather than selecting one good label order, ECC constructs multiple Classifier Chains each with a distinct random order of chains. ECC trains m CC classifier $h_{11}, .., h_{Lm}$; each of the CC classifier comes with a random order of chain. As a result, each of the chains produces a vector of confidence. Finally, we combine the prediction by a voting schema with specific threshold.

A common advantage of using an ensemble method is their well-known effect of increasing the overall predictive performance. ECC method reduces the risk of having an overall negative effect on classification accuracy. In addition, using an ensemble of chains requires only a linear time cost with respect to the number of iterations [10].

A potential drawback of ECC is the large number of instance processed for each chain. With L labels and m iteration, we need $m \times L$ instances for each sample input. A large portion of these data is redundant which cause higher computational complexity and memory usage. Although this does not present loss in the predictive performance, it may become a blocking point with a large scale dataset. A deep investigation has shown that a similar predictive performance can be obtain with significantly less complexity [10]. Read [10] shows that with only 75% of training instances, 50% of the attribute space, and only 10 iteration, accuracy is negligibly less than when using the full dataset.

Chapter 5: Ensembles of Classifier Chains with Incomplete Label Assignment (ECCI)

In this project, we consider the multi-label learning problem in which portion of label assignment is missing. For example, only three true value out of total five is given $< T, F, ?, F, ? >$ for a specific training instance. This scenario is expected in many practical application. For example, in image annotation, a user provides partial tag assignment labels for the image.

Many methods have been explored the literature of multi-label learning. Ranging from simple problem transformation to more complex method that capture correlation among labels. However, mostly all present work do not address the challenge with incomplete labeled data. The goal of this project is to learn an ensemble classifier chain model for the training examples with incomplete label assignment.

5.1 Estimating the true value

First, we start training all ECC models using instances with complete labels. We continue training our ECC models using the rest of training instances with incomplete assignment. As we proceed training, we encounter an instance with missing label. At this point, given previous true labels along with the attribute, we can using existing models to predict the true value of the missing label. We stop propagating the chain as we do not have an absolute true value, rather we have an estimation. A record is kept for each estimation made. We continue the same process for the same instance for each iteration, or chain order. A multiple loop might be required to fill all missing labels with an estimation.

After looping through all iterations, or chains, a true label is then assigned using a voting scheme over all records for each label and for each iteration of missing labels. We continue this process for all instances with incomplete assignment. At the end, we re-train our ECC models again using all training set.

5.2 Maximizing the accuracy of estimation

Using estimated, or predicted, values to fill missing labels during training can present a risk as these can ruin the predictive performance of overall ECC learning. This problem become severe as the number of missing labels increases in an instance. To mitigate this problem, we repeat the estimation step again using the updated ECC model. This process can continue until a certain threshold, or number of iterations is reached.

5.3 Selective estimation order

Given an instance with multiple missing labels, there might be different options to start with in regard which label to estimate first. As an example, consider an instance with the following labels, $< T, F, \mathbf{?}, F, \mathbf{?} >$, with a given iterations, or chain orders as follows:

Chain 1st: 1, 2, **3**, 4, **5**
Chain 2nd: 2, **3**, 1, 4, **5**
Chain 3rd: 1, 2, **5**, 4, **3**
Chain 4th: 2, **5**, 1, 4, **3**

Now, we can either start estimating with the 3rd or 5th label. In this situation, we are left off with a higher probability of estimating true labels when we start with a most confident label using existing ECC model.

Algorithm 5.3.1: ESTIMATETRUEVALUE($ECC\_models, training\_set$)

comment: for a given set of instance with incomplete label assignments

for $re\_train\_iter \leftarrow 0$ **to** *threshold*

do
- **while** *there exist a missing label in the set*
  - **do**
    - **for each** *instance* $\in$ *given set*
      - **do**
        - *Find most confident starting missing label in all chains*
        - *propagate chain sequence until missing label is reached*
        - *predict/estimate missing label and recordvalue*
    - *Compute voting for estimated labels and assign it as a true value*
- *re_train all models*
- *re_set missing label to unknown*

Chapter 6: Experiments

A complete evaluation is presented in this section in order to test all variety of methods with changing parameters as discussed earlier. We present a comparison between BR and CC methods to justify the important role of label correlation in improving predictive performance. Then we evaluate ECC performance against CC and show the effect of different parameters that plays role in the predictive performance of the classifier. We also compare our result to the performance of Two-Dimensional Active Learning for Image Classification [7] which addresses the label correlation in their core learner.

We evaluate all of our algorithms using the open-source library, LIBSVM[1]. We use Support Vector Machines based as the base-classifier for our problem transformation methods, with default parameters.

Next, we introduce the dataset in section 3.1, our evaluation measurement in 3.2, and finally we present the result of using ECC against other methods with incomplete label assignment.

6.1 Dataset

For this experiment, we focused on a real-world dataset. The set is constructed from natural scene set with six image categories. These datasets are publicly available at http://www.csie.ntu.edu.tw/ cjlin/libsvmtools/datasets/multilabel.html. This dataset was first used by multi-label image classification [1]. It contains 2407 natural scene images with one or more labels corresponding to six different categories. Label Cardinality, which is a standard measure of multi-labelled-ness, is about 1.07 label [12]. It is simply an average number of labels associated with an instance defined as followed:

$$L_{card} = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{L} t_i$$

[1] http://www.csie.ntu.edu.tw/ cjlin/libsvm/

In Read. [10], they also introduced a new measurement to have some sense about the uniformity of labelling scheme, called Puniq. In this dataset, the PUniq value is equal to 0.006.

6.2 Evaluation Measures

There are different type of evaluation in the literature of multi-label classification. A per-label basis, and a per-set basis evaluation. The former evaluate the correctness of prediction at the label base where the latter evaluate the correctness of prediction as a whole set, i.e. a set of labels $\hat{y}$ must exactly matches the true label y set (0/1 LOSS). We focus in the mper-label basis evaluation in this project.

We use f-measure, or F1 score, in our evaluation method. F1 score is the harmonic mean of precision and recall:

$$F_1 score = 2 \times \frac{precision \times recall}{precision + recall}$$

6.3 Result: Comparison of different methods

We started with a comparison of different methods, binary relevance BR, classifier chain CC, 2DAL with random sampling[2], and ensemble classifier chain ECC.

We compared the result of our experiment using the methods, CC and ECC along with the result provided by Read [10]. We also show the predictive performance of the 2DAL [7] work, which does not include the Active Learning part of their method.

The below table shows the average F1 score over all different labels. On this dataset, we perform 60/40 train/test splits. It clearly shows that ECC is making a better result than any of the listed methods. We also noticed that there is a huge difference between BR and CC. This indicates that a label correlation plays an important role in improving the predictive performance. ECC takes the same approach of considering the label correlation. In addition, ECC has a better performance than CC because it uses different sets of ordering for each chain which help overcome the problem with poorly ordered chains. For the parameters of ECC, we used 25 iterations, or chain orders.

[2]This result does not consider neither active learning nor online methods. The 2DAL evaluation presented here, is the learner predictive performance only with random selection.

Table 6.1: Comparison of different methods

Label	BR	2DAL	CC	ECC
Label 1	0.5797	0.6744	0.7467	0.7839
Label 2	0.7923	0.9002	0.8839	0.8791
Label 3	0.6634	0.8927	0.8116	0.8252
Label 4	0.8000	0.8071	0.8254	0.8291
Label 5	0.5181	0.6122	0.6049	0.6480
Label 6	0.5008	0.6856	0.7183	0.6848
average	0.6423	0.7620	0.7651	0.7750

6.4 Results: The number of iterations in ECC

In this section, we discuss the importance of the number of iterations. Below we show the effect of using different number of iterations, or chains orders, when using ECC.

Table 6.2: The Number of Iterations Parameter in ECC

# iteration	1	5	10	15	20	25
Label 1	0.7467	0.753	0.7622	0.7706	0.7711	0.7839
Label 2	0.8839	0.8889	0.9104	0.8768	0.8938	0.8791
Label 3	0.8116	0.8276	0.8163	0.8188	0.807	0.8252
Label 4	0.8254	0.8235	0.8389	0.8173	0.8276	0.8291
Label 5	0.6049	0.6247	0.6555	0.6573	0.6461	0.648
Label 6	0.7183	0.703	0.6523	0.7003	0.6981	0.6848
average	0.7651	0.7701	0.7726	0.7735	0.7739	0.7750

Illustrated below, the F1 score tends to converge when it gets close to 18 chains. The fact is that the size of the ensemble required is variable and dependant on the the problem [5]. This present a problem that the ensembles constructed may to be unnecessarily large, which requires additional computational resources. A further work has been conducted to develop Selective Ensemble of Classifier Chains (SECC) method, which reduces the ensemble size of ECC[5].

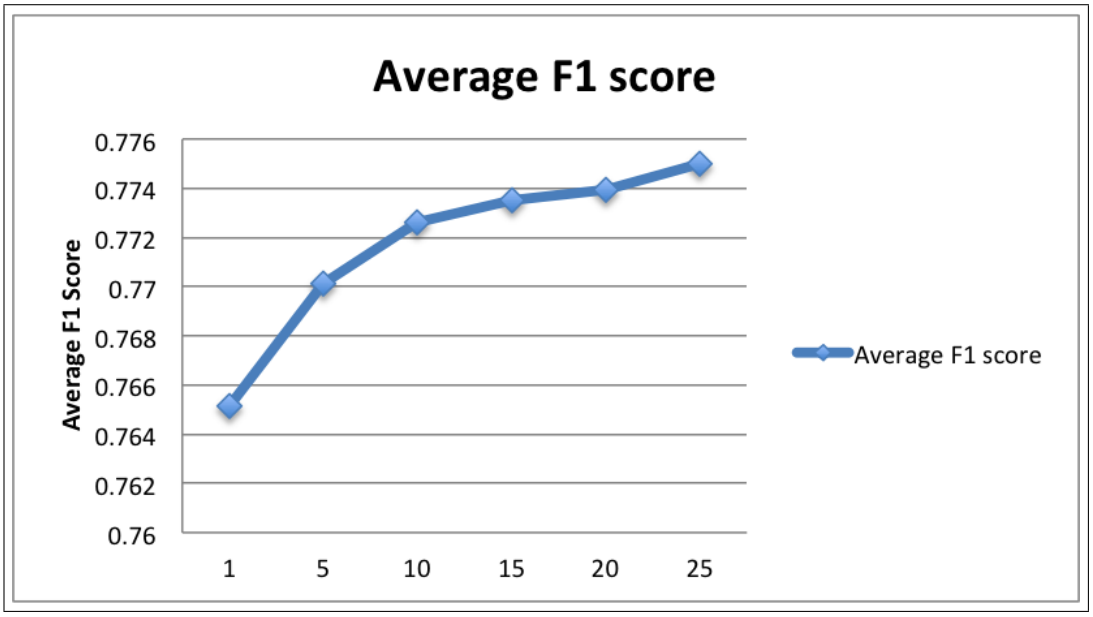

Figure 6.1: Comparison of using different number of iterations.

6.5 Results: ECC with Incomplete Labels (ECCI)

To study the problem of incomplete label assignment, we evaluate the proposed approach on the same dataset, the natural scene set. The focus of this experiment is to show how we can handle incomplete assignment in multi-label classification. We also put an effort to mitigate the risk of ruining the predictive performance by iteratively re-estimating the true value of missing labels.

To simulate the effect of the number of missing label in an instance on the predictive performance, we randomly choose on average, 1, 1.5, 2, 2.5, or 3 labels from the set and remove the label assignments. Below we show the predictive performance as a function of average number of missing labels in an instance.

Table 6.3: ECC with Incomplete Labels

# iteration	1	1.5	2	2.5	3
Label 1	0.7644	0.7351	0.7139	0.5897	0.3904
Label 2	0.8746	0.8622	0.8025	0.5259	0.2932
Label 3	0.8172	0.8106	0.7657	0.6012	0.3225
Label 4	0.8491	0.8358	0.7899	0.612	0.3313
Label 5	0.6396	0.6541	0.6337	0.5106	0.4191
Label 6	0.7083	0.6667	0.6375	0.5386	0.3621
average	0.7755	0.7607	0.7238	0.5630	0.3531

As expected, the predictive performance tends to get worse as we increase the number of incomplete labels. With that said, we also notice that the predictive performance on average seems to do well under 2.

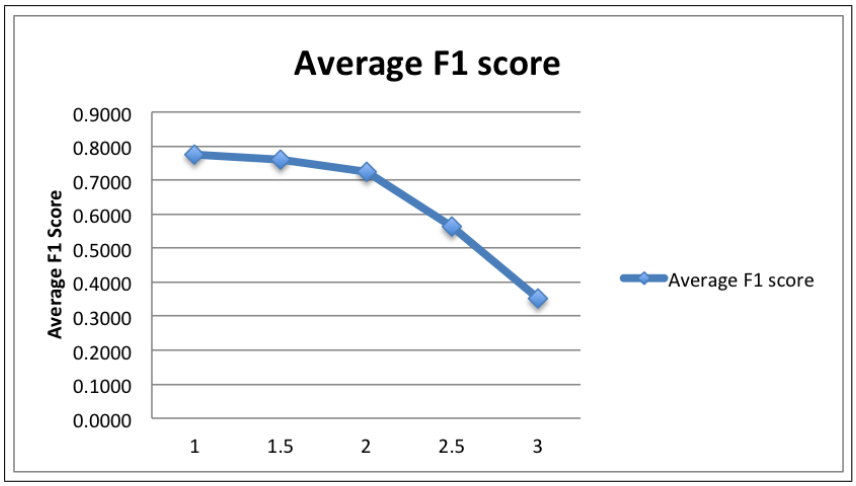

Figure 6.2: Effect of average number of missing label in an instance.

6.6 Results: ECCI with different number of re-training iteration

From the previous section, it is obvious that as the number of missing label increases in the dataset, the predictive performance tend to get worsen. One explanation to this problem is that the percentage of correctness for the predicted, or estimated, true value is low.

In order to help maintain a better predictive performance with incomplete label assignment, we choose to maximize the accuracy of the estimation process. We simply repeat the estimation step again using the updated ECC model. we repeat this process for 4 iterations.

To simulate the effect of using different number of re-training iterations, we conduct an experiment using an average of 1.5 missing labels in an instance. We run the experiment as a function of number of iterations. The table below shows the result:

Table 6.4: Maximizing Estimation of Incomplete Labels

# iteration	none	1	2	3	4
Label 1	0.6308	0.7338	0.7521	0.7413	0.7413
Label 2	0.8626	0.8889	0.8763	0.861	0.861
Label 3	0.7519	0.7955	0.8224	0.8127	0.8143
Label 4	0.7891	0.8235	0.807	0.8304	0.8304
Label 5	0.5355	0.5843	0.6322	0.6421	0.6421
Label 6	0.4242	0.5792	0.6527	0.6754	0.6754
average	0.6656	0.7342	0.7571	0.7604	0.7607

Clearly the predictive performance increases when we process the estimation of true value for incomplete labels. We also notice that the performance increases with the re-training process.

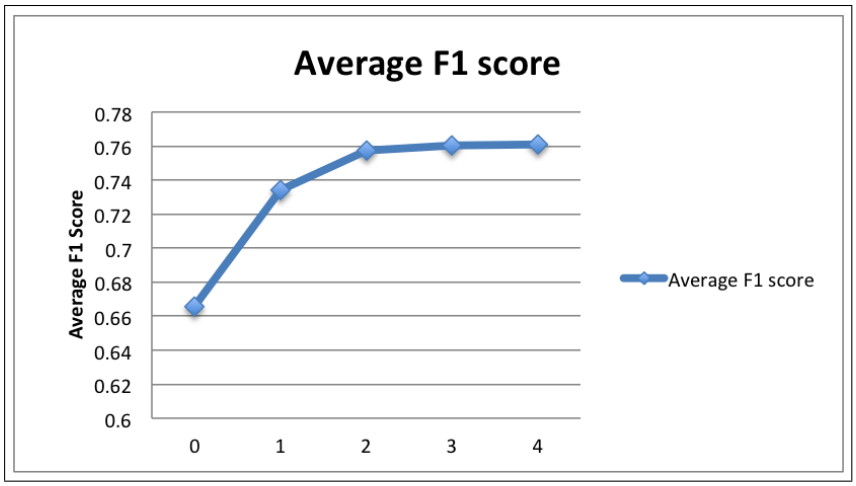

Figure 6.3: Effect of re-training process.

Chapter 7: Future Work

In future work, we can extend this project to apply an active learning selection. Active Learning is one of the most used methods in image classification, as it can significantly reduce the human cost in labeling training samples. Active Learning methods iteratively annotate a set of selected instances so that the classification error is minimized with each iteration. A previous work by 2DAL [7] shows a great improvement in the predictive performance when using an active learning scheme.

Bibliography

[1] Matthew R. Boutell, Jiebo Luo, Xipeng Shen, and Christopher M. Brown. Learning multi-label scene classification, 2004.

[2] Krzysztof Dembczynski, Weiwei Cheng, and Eyke Hüllermeier. Bayes optimal multilabel classification via probabilistic classifier chains. In Johannes Fürnkranz and Thorsten Joachims, editors, *ICML*, pages 279–286. Omnipress, 2010.

[3] Johannes Fürnkranz, Eyke Hüllermeier, Eneldo Loza Mencía, and Klaus Brinker. Multilabel classification via calibrated label ranking. *Mach. Learn.*, 73(2):133–153, November 2008.

[4] Shantanu Godbole and Sunita Sarawagi. Discriminative methods for multi-labeled classification. In *In Proceedings of the 8th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 22–30. Springer, 2004.

[5] Nan Li and Zhi-Hua Zhou. Selective ensemble of classifier chains. In Zhi-Hua Zhou, Fabio Roli, and Josef Kittler, editors, *Multiple Classifier Systems*, volume 7872 of *Lecture Notes in Computer Science*, pages 146–156. Springer Berlin Heidelberg, 2013.

[6] Andrew Kachites McCallum. Multi-label text classification with a mixture model trained by em. In *AAAI 99 Workshop on Text Learning*, 1999.

[7] G.-J. Qi, Xian-Sheng Hua, Yong Rui, Jinhui Tang, and Hong-Jiang Zhang. Two-dimensional active learning for image classification. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, 2008.

[8] Guo-Jun Qi, Xian-Sheng Hua, Yong Rui, Jinhui Tang, Tao Mei, and Hong-Jiang Zhang. Correlative multi-label video annotation. In *Proceedings of the 15th international conference on Multimedia*, MULTIMEDIA '07, pages 17–26, New York, NY, USA, 2007. ACM.

[9] Jesse Read, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank. Classifier chain. http://www.tsc.uc3m.es/ jesse/talks/UC3M-Charla2.pdf.

[10] Jesse Read, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank. Classifier chains for multi-label classification.

[11] Robert E. Schapire and Yoram Singer. Boostexter: A boosting-based system for text categorization. In *Machine Learning*, pages 135–168, 2000.

[12] Grigorios Tsoumakas and Ioannis Vlahavas. Random k-labelsets: An ensemble method for multilabel classification. In *Proceedings of the 18th European conference on Machine Learning*, ECML '07, pages 406–417, Berlin, Heidelberg, 2007. Springer-Verlag.

[13] Yiming Yang. An evaluation of statistical approaches to text categorization. *Journal of Information Retrieval*, 1:67–88, 1999.