One Approach to Multi-Label Text Classification
Alexander D'yakonov
djakonov@mail.ru
professor at Lomonosov Moscow State University

1 Annotation

Algorithm which has taken the first place in competition "On Greek Media Monitoring Multilabel Classification (WISE 2014)" [4] is described. The competition was related to the problem of multi-label classification for articles coming from Greek printed media. We proposed a simple and effective algorithm.

2 The problem

Raw data comes from the scanning of print media, article segmentation, and optical character segmentation. The articles (texts) have been labeled by experts by using labels from the set $\{1, 2, \ldots, l\}$, $l = 203$. A label can be viewed as a topic discussed in the text. One article may have several labels. Articles are described by $n = 301561$ numerical attributes corresponded to the tokens encountered inside the texts. However organizers have computed tf-idf statistic and made unit normalization. So the article description is a real vector $(x_1, \ldots, x_n) \in \mathbb{R}^n$ of unit L_2-norm. The problem is to build multi-label classifiers for the automated annotation of articles into topics. Participants of WISE 2014 competition has a training set – it is $m = 64857$ labeled descriptions of articles from May 2013 to September 2013, and a test set contained 34923 unlabeled descriptions. Test set labels are known only to organizers. The evaluation metric for this competition is Mean F1-Score [3].

Let $(x_{t1}, \ldots, x_{tn})$ be a description of the t-th article and $(y_{t1}, \ldots, y_{tl})$ be its label vector: $y_{tj} = 1$ if and only if the t-th article has the j-th label. Many results in Russian scientific school of recognition and data mining are based on the fact that algorithm should be presented as a superposition of two algorithms: the first one obtains vector $(g_{t1}, \ldots, g_{tl})$ of estimations, g_{tj} is an "estimation of belonging" of the t-th article to the j-th class [6], [1]. The second algorithm transforms the estimation vector to a binary label vector $(a_{t1}, \ldots, a_{tl}) \in \{0, 1\}^l$. Nonzero elements of the vector are labels, which we assign to the t-th article.

As the first algorithm we use regressor (or linear combination of regressors) we'll call it regressor operator, the second algorithm should be simple enough, we'll call it decision rule. Thus we construct solution by superposition of a regressor operator and a decision rule. The main idea of our approach is that the algorithm should be simple, interpretable and efficient (have high performance). Simple algorithms are more reliable and easy to tune. Interpretability can be very useful in practice for experts on print media. Efficiency can be reached by blending of several simple algorithms, so the regressor operator will be constructed as a combinations of simple interpretable regressors.

Table 1: The best performances in local tests

decision rule:	(1)	(2)	(3)	(4)
50NN	0.6204	0.6760	0.6549	0.6759
logistic regression	0.7734	0.7829	0.7738	0.7828
ridge regression	0.7634	0.7641	0.7418	0.7642

3 Decision rule

We have investigated a variety of different decision rules:

$$C(g_1, \ldots, g_l) = (a_1, \ldots, a_l),$$

$$a_j = 1 \Leftrightarrow g_j \geq \min(p, \max(g_1, \ldots, g_l)), \tag{1}$$

$$a_j = 1 \Leftrightarrow g_j \geq p \cdot \max(g_1, \ldots, g_l), \tag{2}$$

$$a_j = 1 \Leftrightarrow g_j \geq \min\left(p \cdot (g_1 + \cdots + g_l),\, \max(g_1, \ldots, g_l)\right), \tag{3}$$

$$a_j = 1 \Leftrightarrow g_j - \frac{g_1 + \cdots + g_l}{l} \geq p \cdot \max_i \left(g_i - \frac{g_1 + \cdots + g_l}{l}\right). \tag{4}$$

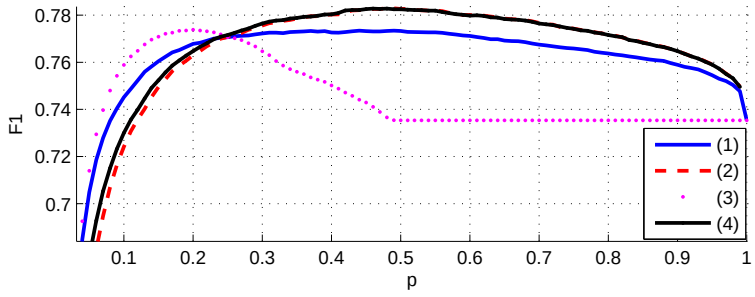

Figure 1: Performance of decision rules

Fig.1 shows their performance when we use logistic regression as a regressor operator. The second and forth decision rules seem to be more effective, see Tab.1. In our final solution we use second rule and $p = 0.55$.

4 Regressor operator

One of the most popular algorithms for text classification is ***k*NN** (k-nearest neighbors). For this problem kNN is not very effective as a regressor operator (0.6760 mean F1-score when $k = 50$), but it is useful in linear combination for the final solution. Our version of kNN uses a weighted average of the k nearest

neighbors, weighted by their cosine similarity. In estimation vector $(g_1, \ldots, g_l)$ obtained by kNN g_j equals to a sum of weights of the nearest k neighbors with the j-th label.

Nearest centroid classifier is a little worse (0.6314 mean F1-score in local tests and 0.624 in leaderboard).

Logistic regression reveals to be the most efficient regressor. We use scikit-learn [5] realization of this popular algorithm and the model

```
linear_model.LogisticRegression(penalty='l1', C=6.0, tol=0.001)
```

gives 0.7829 mean F1-score in local tests, 0.7734 – in leaderbord. Parameter $C = 6.0$ (inverse of regularization strength) is optimal when we use constant value for every label. We do not tune model parameters for each label separately, the reason is that some labels are very rare, so it can result in overfitting.

Ridge regression is also much better than kNN. Scikit-learn model

```
linear_model.Ridge(0.8)
```

gives 0.7641 mean F1-score in local tests.

We can reduce number of features by performing singular value decomposition (SVD) of matrix $||x_{ti}||$. This approach was a quite effective in a similar problem [2]. However our experiments show that the more number of the largest singular values and associated singular vectors we use the better performance. And SVD calculation takes a lot of time: for acceptable time only 300 singular vectors were calculated and the performance of ridge regression was still less than on initial data.

5 Another features

We can use other features and feature transforms, for example features $(x_{t1}^d, \ldots, x_{tn}^d)$ (values to the d-th power). Value $d = 0.8$ slightly increases performance (+0.005). Another way is a features adding, for example we can sort vector $x = (x_{t1}^d, \ldots, x_{tn}^d)$ and concatenate the initial vector x and a sorted vector x_{sort}. It also slightly improves performance, but we have not used feature generations in our final solution in the competition.

6 Parameter tuning

We perform a parameter tuning on the last 14857 articles in the training set, so we do not use cross validation. Fig.2 shows that the training set is rather diverse. However results on the last articles correlates with a public leaderboard.

7 Blending

One of the most popular strategies in data mining is a blending, when we use several algorithms. The simplest way of blending in our problem is a linear

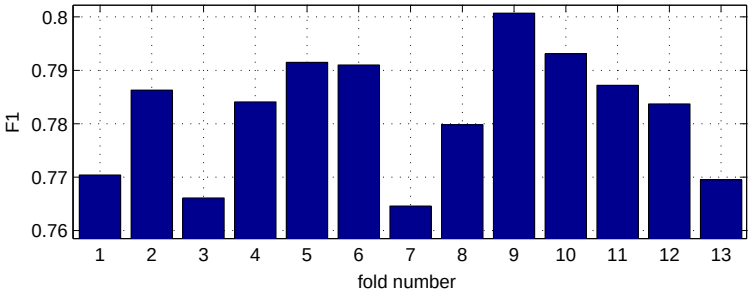

Figure 2: Performance of logistic regression on different folds

combination. Fig.3 shows performance of the operator

$$\alpha A_1 + (1-\alpha)A_2,\ \alpha \in [0,1],$$

for different pairs of regressor operators A_1, A_2 and decision rule (2). When A_1 means logistic regression, A_2 means ridge regression and $\alpha = 0.68$, the performance increases to 0.7815 mean F1-score.

As a regressor operator we use a linear combinations of regressors. According to our investigations, we choose the next basic models:

```
linear_model.LogisticRegression(penalty='l1', C=2.0, tol=0.001),
linear_model.LogisticRegression(penalty='l1', C=6.0, tol=0.001),
linear_model.LogisticRegression(penalty='l1', C=10.0, tol=0.001,
linear_model.Ridge(alpha=0.4),
linear_model.Ridge(alpha=0.8),
linear_model.Ridge(alpha=1.2),
1NN, 2NN, 3NN, 50NN.
```

For each label we train these models on the first 50000 texts from the training set and find their optimal combination by ridge regression on the other texts. To build final solution we should retrain these models on the whole training set, construct the linear combination for every label (coefficients are already known) and then apply a decision rule (2) with $p = 0.55$. This simple approach shows performance of 0.7945 mean F1-score in local tests, 0.794 in public leaderboard, 0.7969 in private leaderboard.

References

[1] A. G. Dyakonov. Algebraic closures of a generalized model of estimation algorithms. *Doklady Mathematics*, 78:936–939, 3 2008.

[2] Alexander Dyakonov. A blending of simple algorithms for topical classification. *Rough Sets and Current Trends in Computing, Lecture Notes in Computer Science*, 7413:432–438, 2012.

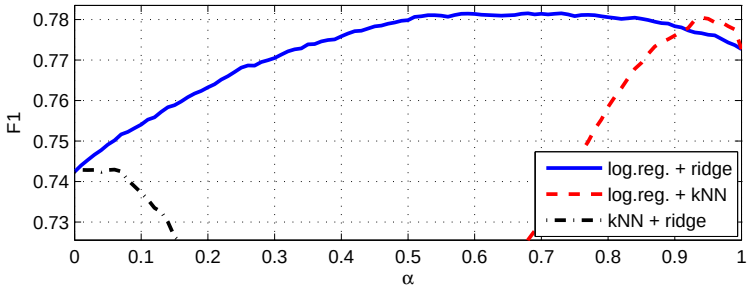

Figure 3: Performance of linear combinations

[3] F1-score. http://en.wikipedia.org/wiki/f1_score.

[4] Kaggle. http://www.kaggle.com/c/wise-2014/.

[5] scikit learn. http://scikit-learn.org/.

[6] Yu. I. Zhuravlev. An algebraic approach to recognition and classification problems. *Mathematical Methods in Recognition and Classification Problems (Hafner Press, Nauka, Moscow)*, 1986, 1978.