T.Kamishima, M.Hamasaki, and S.Akaho, "Personalized Tag Predition Boosted by BaggTaming --- A Case Study of the Hatena Bookmark", The 3rd Int'l Workshop on Data-Mining and Statistical Science (2008)

International Workshop on Data-Mining and Statistical Science (DMSS2008)

Personalized Tag Predition Boosted by BaggTaming

A Case Study of the Hatena Bookmark

Toshihiro Kamishima — National Institute of Advanced Industrial Science and Technology (AIST)
mail@kamishima.net, http://www.kamishima.net/

Masahiro Hamasaki — (affiliation as previous author)
hamasaki@ni.aist.go.jp, http://staff.aist.go.jp/masahiro.hamasaki/index-e.html

Shotaro Akaho — (affiliation as previous author)
s.akaho@aist.go.jp, http://staff.aist.go.jp/s.akaho/

keywords: hatena, inductive transfer, bagging, collaborative tagging, BaggTaming

Summary

We proposed *BaggTaming* to boost the prediction accuracy by exploiting additional data whose class labels are less reliable. This algorithm is successfully applied to the personalized tag predicition for the data collected from the *delicious*. To check whether our method is generally effective, we test the data crawled from the *hatena* bookmark.

1. Introduction

We stated a learning problem, which we call *taming*, and develop a method for this problem in [神嶌 08b, 神嶌 08c, Kamishima 08a]. The learner for this taming requests two types of training data sets, *tame* and *wild*. The labels of tame data is highly consistent with a target concept, which we actually want to learn. In contrast, wild data are not so well maintained; thus, some labels are consistent with the target concept, while some others are not. Additionally, we assume that wild data are much more abundant than tame data. This assumption is reasonable, because it is generally difficult to provide a large tame data set due to its high maintenance cost. The goal of the taming is to acquire more accurate classifiers by exploiting wild data. To achieve this goal, we developed a *BaggTaming* method, which is a modified version of bagging [Breiman 96].

We applied this BaggTaming to a personalized tag prediction for the data set collected from the *delicious*[*1], and showed the effectiveness of our method. We expect that this taming technique would be generally helpful for a tag prediction task on another collaborative tagging service. To check this hypothesis, we performed the tag prediction task on the *hatena* bookmark[*2], which is one of popular social bookmarks in Japan.

∗1 http://delicious.com/
∗2 http://b.hatena.ne.jp/

In Chapter 2, we formalize taming and describe BaggTaming. Chapter 3 shows the experimental results on the hatena bookmark data. Finally, we conclude in Chapter 4.

2. Taming Task and Its Solution

In this chapter, we describe the formalization of taming, our BaggTaming algorithm, and its application to the personalization of the social bookmark tags.

2·1 What Is Taming

We first address the learning problem of taming. Here, we are focused on classification, though taming techniques can be applied to other types of supervised learning tasks, such as regression. An object is represented by a feature vector, $\mathbf{x}$. A variable c represents the class to which the object should be classified. A training example is a pair of an object and its class, $(c_i, \mathbf{x}_i)$. The goal of classification is to acquire a classifier that can predict an appropriate class for a input feature vector from a set of training examples.

A standard classifier is learned from only one homogeneous set of training examples. These training examples, which expresses the target concept to be learned, are assumed to be independently sampled from an identical distribution, $\mathrm{P}[c, \mathbf{x}]$. On the other hand, in our taming case, two types of training examples, tame and wild, are required. Similar to standard classification, tame data are assumed to be independently sampled from an identical

1: $s = 0$
2: for $t = 1, 2, \ldots T$ do
3: generate a training set $\mathcal{D}_t$ by sampling with replacement from $\mathcal{D}_W$
4: learn a weak classifier $\hat{f}_t(\mathbf{x})$ from training set $\mathcal{D}_t$
5: calculate the accuracy p_t of the classifier $\hat{f}_t(\mathbf{x})$ on the tame set $\mathcal{D}_T$
6: if (p_t satisfies the acceptance condition) then
7: $s = s + 1$; $\hat{f}_s(\mathbf{x}) = \hat{f}_t(\mathbf{x})$; $p_s = p_t$
8: if $s = 0$ then output a default classifier and exit
9: output the weak classifiers $\hat{f}_1(\mathbf{x}), \ldots, \hat{f}_s(\mathbf{x})$,
together with their corresponding accuracies $p_1, \ldots, p_s$.

Fig. 1 BaggTaming algorithm

distribution, $\mathrm{P}[c, \mathbf{x}]$, that corresponds to the target concept. This set of tame data is denoted by $\mathcal{D}_T = \{(c_i, \mathbf{x}_i)\}_{i=1}^{N_T}$, where $N_T = |\mathcal{D}_T|$. A wild data set might include examples that are sampled from distributions that express irrelevant concepts, together with examples that are consistent with the target concept. We further assume that a wild data set contains examples of the target concept as at least a few times large as N_T, and the learner cannot know which examples are generated from the target distribution. a wild data set is denoted by $\mathcal{D}_W = \{(c_i, \mathbf{x}_i)\}_{i=1}^{N_W}$, where $N_W = |\mathcal{D}_W|$. Finally, we assume wild data is much abundant than tame data, i.e., $N_W \gg N_T$.

2·2 BaggTaming

To solve the above problem of taming, we developed a BaggTaming (Bootstrap AGGregated TAMING) algorithm, which is a variant of bagging [Breiman 96]. In [Breiman 98, section 7], Breiman reported an improvement technique for bagging. Likewise standard bagging, training examples are bootstrap-sampled from a given training set, and these examples are fed to a weak learner. In this step, noises are added to the feature vectors in these sampled examples. This addition of noise can improve the prediction accuracy, because this modification enables to learn various types of weak classifiers and thus contributes to reducing the variance in the prediction error. Our BaggTaming is inspired by this technique. In the process of BaggTaming, training examples are sampled from wild data instead from tame data. We expected that the variance part of the error can be more drastically reduced, because a wild data set contains more various types of examples than a tame set. However, here is one difficulty. A wild data set contains examples of irrelevant concepts, which we want to ignore, and the learner cannot obtain the information which example is of the target concept. To avoid this difficulty, we exploit a tame data, which is consistent with the target concept. If most of examples that have been used for training the weak classifier are consistent with the target concept, the empirical accuracy of the classifier can be assumed to be high. Accordingly, unusable weak classifiers are filtered out base on the accuracy on the tame set. Specifically, learning of our BaggTaming iterates the following two steps:

- Training examples are randomly sampled from the wild data set, and a weak classifier is learned from these training examples.
- The empirical accuracy of the weak classifier on the tame data is computed. If the accuracy is sufficiently high, the classifier is accepted; otherwise, it is rejected, and a new classifier is repeatedly learned from new samples.

By iterating these procedures, the learner can obtain various types of weak classifiers that are consistent with the target concept. Once weak classifiers are acquired in this way, the final class is inferred by majority voting. Our BaggTaming algorithm is shown in Figure 1. Note that this is the version used in [神嶌 08c], which is the slightly modified version in [Kamishima 08a, 神嶌 08b].

We describe the acceptance condition and a default classifier of this algorithm. First, $\hat{f}_T(\mathbf{x})$, is learned from the tame data set, $\mathcal{D}_T$, by using a weak learner. Then, the accuracy, p_T, of the classifier $\hat{f}_T(\mathbf{x})$ on the $\mathcal{D}_T$ is computed. In step 6, if p_t is statistically not worse than p_T at the significance level of α, a candidate classifier, $\hat{f}_t(\mathbf{x})$, is accepted. Note that Z-test is used for testing in the experiment. The classifier $\hat{f}_T(\mathbf{x})$ is also used as a default classifier at step 8. Because no accepted weak classifier cannot be learned from the wild data, it would be reasonable to conclude that the wild data contains no useful information and to ignore the wild data at all.

The classification procedure of our BaggTaming is similar to that of standard bagging. Majority voting is used to predict a class for a new vector, but each vote is weighted. The accuracy p_t would be high if the $\mathcal{D}_t$ contains many examples that are consistent with the target concept. Therefore, we weigh each weak classifier by p_t. Specifically, the class to which the feature vector $\mathbf{x}$ should belong is determined by

$$\hat{c} = \arg\max_{c \in \mathcal{C}} \sum_{t=1}^{s} p_t \mathrm{I}[c = \hat{f}_t(\mathbf{x})]. \tag{1}$$

2·3 Application of BaggTaming to the Personalized Tag Prediction of Social Bookmarks

Social bookmark services enable users to register their favorite Web pages. To these registered pages, users can assign tags, which are words that express the contents, characteristics, or categories of the tagged pages. These tags are useful for searching or classifying their own reg-

istered pages. In addition, these registered pages and assigned tags can be shared among users of the service. With these shared tags, users can search the pages that other users have registered, and they can find like-minded users. Because users can freely choose their favorite tags, the semantics of social tags can vary greatly [Golder 06]. As a consequence, the tags labeled by one user may be inappropriate to another user. When searching for documents with shared tags, users might find undesired documents or miss relevant pages.

To address this problem, we will provide the way to predict the tags that are personalized for a specific target user. First, for each candidate tag, we acquire a binary classifier to discriminate whether the target user will assign the candidate tag to Web pages. To be personalized this discrimination, this classifier is trained from the Web pages that are tagged by the target user before, because such tags are considered to be reflecting the target user's viewpoint. Using such classifiers enables to induce whether a given candidate tag is appropriate for any Web pages from the target user's viewpoint. However, the Web pages that can be used for training are generally insufficient, because one user cannot assign tags for not so many Web pages. Accordingly, the learned classifiers cannot make fully precise prediction.

Our BaggTaming is helpful for coping with this difficulty. The Web pages that tagged by the target user are highly consistent with the user's concept, while the number of pages are generally small. Additionally, we can exploit the enormous Web pages that have been tagged by the non-target users, though the most of these are inconsistent with the target user's concept. We apply our BaggTaming by treating the target user's and the non-target users' tagging information as tame and wild data, respectively. BaggTaming boosts the prediction accuracy classifier, because these wild data partially contain useful information for learning the target user's concept and such information is helpful for improving the prediction.

3. A Case Study of the Hatena Bookmark

In this chapter, we applied our BaggTaming to the data collected from *hatena* bookmark.

We first overview this data set. We crawled the collaborative tagging site, *hatena* bookmark, in November, 2006. The numbers of unique URLs, tags, and users were 488978, 117264, and 15526, respectively. We found 6165052 bookmarks, which were pairs of a URL and tag.

Details of the experimental procedures are same as in the our previous work [Kamishima 08a]. We picked up the top 20 popular tags. For each tag, which we call a *target tag*, we refer the user who assigned the target tag most frequently as a *tame user*, and the second to twentieth users are denoted by *wild users*. The tags personalized for the tame user is predicted by exploiting the tags that were assigned by wild users. As the features to represent URLs, we adopted the top 100 popular tags except other than target tag. We abandoned the data to which no these top 100 tags are assigned. The sizes of tame and wild data are summarized in Table 1. Note that the sizes of data sets are generally larger than those of the *delicious* data.

Table 1 The sizes of collaborative tagging data sets

target tag	tame	wild	target tag	tame	wild
web	5776	70531	2ch	8691	65493
ネタ	8691	62146	music	5504	61914
blog	7220	72804	software	1923	46284
news	4912	44920	あとで読む	1780	19201
社会	3867	70390	book	2345	53362
hatena	7903	73400	design	1554	37171
neta	2420	62345	google	11132	71846
tool	4634	35245	web2.0	3289	52738
tips	4875	34566	life	5506	54355
game	2657	53533	programming	4823	28676

NOTE: The "target tag" columns show the words used as target tags. The "tame" and "wild" columns show the sizes of the corresponding tame and wild data sets.

We compared our BaggTaming with a baseline method, which is standard bagging whose weak classifiers were trained by using the tame data. For both methods, we adopted again a naive Bayes learner with multinominal model [McCallum 98] as weak classifiers. The number of weak classifiers for a standard bagging was 1000, which is equivalent to the number of iterations, T, of BaggTaming algorithm. The significance level for the acceptance test α was 1%, and the size of $\mathcal{D}_t$ was the same as the tame data set. We performed a 5-fold cross-validation test as described in [Kamishima 08a].

The accuracies on the tame data set are shown in Table 2. Similar to the *delicious* case, our BaggTaming was becoming superior as the number of tame data decreases. This fact enhances the usefulness of BaggTaming, because a taming technique is more useful when less tame data are available. These observations are consistent with our previous result on the *delicious* data. Contrary to the *delicious* case, no improvement was observed if the tame data are abundant. This is due to the fact that the numbers of the *hatena* tame data are much larger than those of the *delicious* data; thus, the sufficient data for the tag prediction were available. However, because such sufficient amount of training data are usable for most of tags, this is not problematic. Further, when abundant tame data is avail-

Table 2 Prediction Accuracies on The Tame Data Set

tag name	size of tame data sets											
	ALL		1/2		1/4		1/8		1/16		1/32	
	BT	bagg	BT	bagg	BT	bagg	BT	bagg	BT	bagg	BT	bagg
web	0.747	0.748	0.718	0.721	0.691	0.705	0.645	0.649	**0.690**	0.632	**0.742**	0.570
ネタ	0.508	0.511	0.528	0.535	0.470	**0.551**	0.443	**0.686**	0.514	**0.712**	0.516	**0.716**
blog	0.576	0.561	0.546	**0.585**	0.592	0.598	**0.675**	0.645	0.686	0.668	**0.760**	0.665
news	0.812	0.814	0.812	0.814	0.811	0.812	0.799	0.785	0.709	**0.782**	0.669	**0.781**
社会	0.715	**0.740**	0.715	0.734	0.708	0.705	0.700	**0.725**	0.685	0.700	0.655	**0.694**
hatena	0.929	0.927	0.839	0.839	0.834	0.832	0.819	0.814	0.815	0.811	0.806	0.799
neta	0.647	0.655	0.641	0.644	0.644	0.630	0.623	0.605	0.594	0.567	**0.584**	0.544
tool	0.802	0.807	0.821	0.831	0.806	**0.825**	0.684	**0.782**	0.748	**0.811**	0.814	0.803
tips	0.478	0.485	0.499	0.510	0.523	0.537	0.505	**0.564**	0.550	0.564	**0.674**	0.615
game	0.727	**0.773**	0.790	**0.824**	0.774	**0.810**	0.763	**0.798**	0.742	0.761	0.747	**0.795**
2ch	0.425	0.425	0.401	0.397	**0.547**	0.431	**0.579**	0.505	0.552	0.551	0.555	0.547
music	0.908	0.914	0.912	0.912	**0.924**	0.901	**0.925**	0.895	**0.923**	0.877	**0.925**	0.883
software	0.665	0.661	**0.705**	0.661	**0.762**	0.674	**0.800**	0.728	**0.806**	0.749	**0.808**	0.713
あとで読む	0.678	0.701	0.719	0.699	**0.729**	0.683	**0.757**	0.696	**0.760**	0.696	**0.774**	0.717
book	0.854	0.848	0.858	0.849	0.856	0.856	0.860	0.850	0.855	0.845	**0.860**	0.810
design	0.692	0.678	0.741	0.705	**0.762**	0.705	**0.762**	0.689	**0.752**	0.713	0.752	0.727
google	0.636	0.641	0.620	0.617	0.625	**0.643**	0.661	0.660	**0.768**	0.706	**0.833**	0.721
web2.0	0.635	0.641	0.653	0.665	**0.706**	0.675	**0.744**	0.654	**0.791**	0.663	**0.825**	0.660
life	0.764	0.764	**0.762**	0.743	0.772	0.757	0.799	0.797	0.823	0.817	**0.838**	0.749
programming	0.499	0.506	0.493	**0.531**	0.538	**0.567**	0.565	0.559	**0.663**	0.626	**0.746**	0.651
win / lose ($\alpha = 1\%$)	0 / 2		2 / 3		6 / 5		7 / 5		8 / 3		12 / 4	
win / lose ($\alpha = 50\%$)	1 / 2		3 / 1		5 / 3		7 / 3		9 / 2		11 / 2	

NOTE: The column "tag name" shows the strings of the target tags. The column pair "ALL" shows the results when all tame data were used, and the column pairs labeled "1/2" – "1/32" show the results when the training tame data were reduced to the 1/2 – 1/32 of the ALL case, respectively. The left "BT" and right "bagg" columns of each pair show the results derived by our BaggTaming and baseline bagging, respectively. Each row shows the accuracies for the corresponding target tag. Bold face indicates that the accuracy was larger than that derived by the other method, and the difference was statistically significant. The last row "win / lose ($\alpha = 1\%$)" shows the number of target tags for which our method won/lost baseline bagging. In the last row, we additionally showed the win/lose counts when the parameter the acceptance ratio, α is set to 50%.

able, more qualified classifier can be acquired by more careful selection of weak learners. Such selection can be enforced by enlarging the parameter, α, which is the significance level of statistical test. As shown in the last row of Table 2, the prediction accuracies are less frequently worsen by adopting BaggTaming. Note that we also tested bagging classifiers whose weak classifiers were trained by using the union of the tame and wild data sets. We observed that such classifiers were slightly inferior than bagging with weak classifiers trained solely by using the tame set.

4. Conclusion

We have proposed a taming, which is a task to learn a small set of highly reliable data and a large set of less reliable data. We developed a BaggTaming algorithm for this task. In this paper, to check whether our technique works well for another data set, we applied our BaggTaming to collaborative tagging data crawled from the *hatena* bookmark. Similar to the previous results, tags were more precisely predicted by using our BaggTaming. We plan to improve the efficiency and to apply tasks other than tag prediction.

Acknowledgments

We wish too thank Dr. Yutaka Matuo and Dr. Atsushi Fujii for their valuable advices.

♢ References ♢

[Breiman 96] Breiman, L.: Bagging Predictors, *Machine Learning*, Vol. 24, pp. 123–140 (1996)

[Breiman 98] Breiman, L.: Arcing Classifiers, *The Annals of Statistics*, Vol. 26, No. 3, pp. 801–849 (1998)

[Golder 06] Golder, S. A. and Huberman, B. A.: Usage Patterns of Collaborative Tagging Systems, *J. of Information Science*, Vol. 32, No. 2, pp. 198–208 (2006)

[Kamishima 08a] Kamishima, T., Hamasaki, M., and Akaho, S.: BaggTaming — Learning from Wild and Tame Data, in *ECML/PKDD2008 Workshop: Wikis, Blogs, Bookmarking Tools – Mining the Web 2.0 Workshop* (2008)

[神嶌 08b] 神嶌 敏弘, 濱崎 雅弘, 赤穂 昭太郎：飼いならし — 飼育・野生混在データからの学習, 人工知能学会全国大会（第 22 回）論文集, 2D1-3 (2008)

[神嶌 08c] 神嶌 敏弘, 濱崎 雅弘, 赤穂 昭太郎：飼いならしを用いた協調タグ付けのタグ予測, 2008 年度統計関連学会連合大会報告集 (2008)

[McCallum 98] McCallum, A. and Nigam, K.: A Comparison of Event Model for Naive Bayes Text Classification, in *AAAI-98 Workshop on Learning for Text Categorization*, pp. 41–48 (1998)