Today, I’d like to talk about a new learning framework and its application to the prediction tag personalization.
We first talk about a learning framework, Taming. This framework uses two types of data sets, tame and wild.
Next, we developed a BaggTaming algorithm for this framework, that is a variant of a bagging.
Finally, we apply this BaggTaming to the tag personalization task for the social bookmarking service.
We begin with a machine learning framework, taming. Supervised learning requires training examples that are labeled as consistent as possible with the target concept. Keeping such consistency is laborious, so the management cost for labeling tends to be high. Therefore, it is generally difficult to collecting a large amount of labeled data. Consequently, the prediction accuracy tends to be low.
To relieve this difficulty, we propose a learning framework, taming. We employed two types of training data sets, tame and wild. The tame data are carefully labeled and so the labeling is highly consistent with the target concept. However, due to its labeling cost, a relatively small amount of data are available. On the other hand, the labeling quality of wild data is low, but these data are much more abundant.

By using both data sets so as to compensate each other’s weak points, we try to improve the prediction accuracy.

Next, we will talk about a BaggTaming algorithm that is designed for taming.
We begin with bagging, because our BaggTaming is a variant of a bagging. Briefly speaking, multiple weak classifiers are learned from bootstrapped training sets, and the predictions of these classifiers are aggregated. In detail, multiple training data sets are first generated by bootstrap sampling from an original training data set. Each training set is fed to a weak learner, and weak classifiers are learned. Any supervised learning methods, such as naive Bayes or SVM, can be used as a weak learner.
Once weak classifiers are learned, the final class is predicted as follows. New data to classify are fed to each weak classifier, and each classifier outputs its estimated class. The final class is the majority class among these estimated classes.
Briman showed the reason why the prediction accuracy is improved by bagging based on bias-variance trade-off. The generalization error can be decomposed into three parts: bias, variance, and noise. The bias is error depending on the model complexity. The variance is error resulted from the sampling of training data, and the noise is intrinsically irreducible error. Generally speaking, by reducing the model complexity, the bias can be decreased, but the variance is increased, and vice versa. Bagging cannot reduce bias and noise because of these reasons, but training weak learners by various types of data contributes to reduce the variance. In summary, bagging is a technique to reduce variance without sacrificing bias.
Based on the theory of bias-variance trade-off, we discuss the idea of our BaggTaming.
In order to more drastically reduce the variance, classifiers should be learned from more various types of examples. For this purpose, training examples are sampled from the wild set, because it contains more diverse data.
But, we now face one difficulty. Because the wild set may contain many non-target data, these data have to be filtered out.
For this purpose, we use a tame data set. Weak classifiers are filtered out if the prediction accuracy on the tame set is low.
We show the detail of the generation process of weak classifiers of BaggTaming. Before learning weak classifiers, we compute the baseline prediction accuracy. Tame and wild data are merged, and weak learner acquires a base classifier from this merged data. The prediction accuracy on the tame set is considered as the baseline accuracy.

Next, weak classifiers are learned. Training examples are generated by bootstrap sampling from the wild data set. From these examples, a candidate weak classifier is acquired, and the prediction accuracy on the tame set is calculated.

The accuracy is compared with the baseline. If it is worse than the baseline the candidate weak classifier is rejected; otherwise, it is accepted. This process is repeated until it is accepted. To avoid the infinite loop, if the number of iteration exceeds the threshold, the currently best classifier is accepted by default.

By repeating this process, multiple classifiers are generated. As in standard bagging, the final result of BaggTaming is derived by majority voting. Next, we apply this BaggTaming to the tag personalization for a social bookmarking service.
We first talk about collaborative tagging, such as a social bookmarking. In this service, users can register their favorite Web pages. To these Web pages, users can assign tags to attribute them. These Web pages and tags can be shared with other users. These shared tags can be exploited for classifying or retrieving Web pages.
Inconsistent Tags

Users can assign any tags based on their own criteria

Tags are generally inconsistent among different users

[Golder et al. 06] pointed out three causes for this inconsistency

**polysemous word**

- **homonymy**: a word having multiple *unrelated* meanings
  - easily distinguishable, and not problematic
- **polysemy**: a word having multiple *related* meanings

**Example:**

![Diagram showing the meaning of "window" as a hole in the wall and the pane of glass that resides within a hole in the wall.]

It is difficult for a user to identify relevant pages

In collaborative tagging, users can assign any tags freely. Therefore, tags are generally inconsistent among users. Golder et al. pointed out three causes for this inconsistency.

The first one is a polysemous word.

Polysemy refers a word having multiple related meanings. For example, someone use a word “window” to refer a hole in the wall. Another refers the pane of glass that resides within the hole.

Because the related meanings confuse users, it is difficult for users to find relevant pages.
The second cause is the level of the specificity. The term that describes an item vary along a hierarchy of specificity ranging from very general to very specific. Most users choose the basic level as the specificity level. But, the basic level may not be unique. For example, you find a black intruder in a kitchen. No one screams “Yow! Arthropod!” However, one would say “Oops! Bugs!”, while another would say “Terrible roach!” Both of them can be considered as the basic level. Users may select the different level of the specificity, and different tags can be assigned to the same page.
Inconsistent Tags

**synonymous word**
multiple words having the same meaning

Example:  television = TV

**the same item may be referred by different tags**

Semantics of tags or the criteria of the tag selection differs for each user

Tags that are appropriate for a user may not be appropriate for another user

The final cause is synonymous words. Synonymy refers that the multiple words having the same meaning. For example, television and TV. In this case, the same item may be referred by different tags.

Semantic of tags or the criteria of the tag selection differs for each user; thus, tags that are appropriate for a user may not be appropriate for another.
In such a case, shared tags cannot perfectly meet everyone’s needs. So, we try the tag personalization, that is the assignment of tags designed for a specific user.

To find the preference pattern of a specific user, all that have to do is analyzing the tags that were tagged by the user before. However, the number of such tags are generally small. Due to this shortage of training examples, the quality of the tag personalization is generally low. In a context of recommendation, this problem is called by “a cold-start problem.”
To relieve this difficulty, we use our BaggTaming algorithm and other users’ tags. Tags assigned by the target user are considered as tame data. These data are fully personalized, but the number of data is relatively small. On the other hand, tags assigned by the non-target user are considered as wild data. These not fully personalized, but are much more abundant. Employing our BaggTaming would improve the quality of the tag personalization. Next, we empirically show this improvement.
The personalized tag prediction task is formalized as classification. Specifically, a class is binary, assigned or not-assigned. As features, we adopted the number of other tags assigned to the target Web page.
For the target user and for each tag in a set of candidate tag, weak classifiers are learned by using a BaggTaming technique. Once classifiers are learned, the system can predict whether each tag is appropriate for a given Web page in the target user’s view. This personalized tag prediction used for the purpose like these.

NOTE: Learning classifiers for every candidate tags is computationally expensive. We plan to introduce the techniques for the multi-label text classification to remedy this drawback.
We next show an experimental procedure. As the target tags, we picked up the 20 most popular tags. For each target tag, the top user of this tag is a tame user, and the second to twentieth users are wild users. Pages assigned by the tame user and the wild users are treated as tame and wild data, respectively. Prediction accuracies of the BaggTaming is compared with those of the standard bagging whose weak classifiers are trained by using only tame data.
Tag Data Overview

Sizes of the tame and wild data sets of the 20 target tags

<table>
<thead>
<tr>
<th>Tag</th>
<th>Tame</th>
<th>Wild</th>
<th>Tag</th>
<th>Tame</th>
<th>Wild</th>
</tr>
</thead>
<tbody>
<tr>
<td>blog</td>
<td>603</td>
<td>24201</td>
<td>web2.0</td>
<td>917</td>
<td>25256</td>
</tr>
<tr>
<td>design</td>
<td>1405</td>
<td>25353</td>
<td>politics</td>
<td>5455</td>
<td>21857</td>
</tr>
<tr>
<td>reference</td>
<td>6323</td>
<td>19512</td>
<td>news</td>
<td>67</td>
<td>28385</td>
</tr>
<tr>
<td>software</td>
<td>3512</td>
<td>30264</td>
<td>howto</td>
<td>6359</td>
<td>23335</td>
</tr>
<tr>
<td>music</td>
<td>6311</td>
<td>22914</td>
<td>imported</td>
<td>172</td>
<td>3165</td>
</tr>
<tr>
<td>programming</td>
<td>4498</td>
<td>25931</td>
<td>linux</td>
<td>1151</td>
<td>24288</td>
</tr>
<tr>
<td>web</td>
<td>1291</td>
<td>31024</td>
<td>blogs</td>
<td>3472</td>
<td>18437</td>
</tr>
<tr>
<td>tools</td>
<td>3493</td>
<td>23625</td>
<td>tutorial</td>
<td>3518</td>
<td>28593</td>
</tr>
<tr>
<td>video</td>
<td>1870</td>
<td>30334</td>
<td>games</td>
<td>3218</td>
<td>22588</td>
</tr>
<tr>
<td>art</td>
<td>6258</td>
<td>16574</td>
<td>free</td>
<td>3509</td>
<td>23543</td>
</tr>
</tbody>
</table>

Table 1 of our original article is incorrect.
Article with errata can be downloaded from Workshop’s or Kamishima’s homepage.

This show the sizes of the tame and wild data sets of the 20 target tags.
You can see that the number of the tame data is much smaller than that of the wild data.
Here, we apologize that Table 1 of our original article incorrect. Article with errata can be downloaded from Workshop’s or Kamishima’s homepage.
### Experimental Results

<table>
<thead>
<tr>
<th>Size of tame data</th>
<th>ALL</th>
<th>1/2</th>
<th>1/4</th>
<th>1/8</th>
<th>1/16</th>
</tr>
</thead>
<tbody>
<tr>
<td>Win/Lose (BT/Bagg)</td>
<td>5/2</td>
<td>8/3</td>
<td>8/2</td>
<td>10/2</td>
<td>11/1</td>
</tr>
</tbody>
</table>

- **BT**: BaggTaming trained by tame+wild data
- **Bagg**: bagging trained by tame data

NOTE: bagging trained by tame+wild data is much worse than **Bagg**

“Win/Lose” show the number of counts that the prediction accuracies of our **BT** is higher/lower than those of the **Bagg** among 20 tags.

While fixing the size of the wild sets, the size of tame sets are gradually reduced from “ALL” to “1/16”

- Our BaggTaming is constantly superior to the bagging
- The advantage of our BaggTaming becomes clearer as the number of tame data lessen

This is a summary of experimental results. The prediction accuracies on the tame data sets are compared. BT and Bagg mean results of our BaggTaming and a standard bagging. We show the number of tags that the our BaggTaming wins or loses among 20 target tags. For example, “5/2” means that our BaggTaming is significantly superior to a standard bagging in five tags, and is significantly inferior in two tags.

Further, we also tested the case where the number of tame data is much smaller. For example, the “1/2” column shows the results when the number of tame data is reduced to a half of the original while fixing the size of the wild sets.

It would be reasonable to say that these two conclusions:
- Our BaggTaming is constantly superior to the bagging.
- The advantage of our BaggTaming becomes clearer as the number of tame data lessen.

BaggTaming is useful when the tame data is less available. This observation enhance the usefulness of our BaggTaming.
We briefly discuss related work. Taming is one of variants of inductive transfer or transfer learning. Inductive transfer is not formally defined but refers the problem of retaining and applying the knowledge learned in one or more tasks to efficiently develop an effective hypothesis for a new task. In our opinion, our taming differs from inductive transfer in this point. Inductive transfer uses the data of other related domain or task. The labeling may be inconsistent in the target domain, but is consistent in the related domain. In taming, wild data consists of a mixture of data for the target and unknown irrelevant tasks.
Conclusion

Summary

- **Stating the taming approach**
  Prediction accuracy was improved by using a small set of reliable tame data together with less reliable abundant wild data

- **Developing BaggTaming algorithm**
  Ensemble learning sampling from wild data, and weak classifiers are filtered out by exploiting tame data

- **Application to collaborative tagging data**
  Personalized tags are more precisely predicted by adopting our BaggTaming technique

Homepage

- http://www.kamishima.net/ (errata of Table 1 can be downloaded)
Future Work

- **Using formal ontology**
  Using the labels in formal ontology will realize highly consistent tags, but it is difficult to label so many documents. By adopting a taming approach together with collaboratively tagged documents, one can classify much more documents by using vocabulary of a formal ontology.

- **Improvement of efficiency**
  Our current sampling technique is highly brute forced and inefficient. Adaptive sampling will contribute to alleviate this inefficiency.

- **Multi-label technique**
  Constructing classifiers for every tags is computationally expensive. A multi-label classification technique would be useful to remove this drawback.