Personalized Tag Prediction Boosted by BaggTaming

A Case Study of the Hatena Bookmark

Int’l Workshop on Data-Mining and Statistical Science
25-26/9/2008

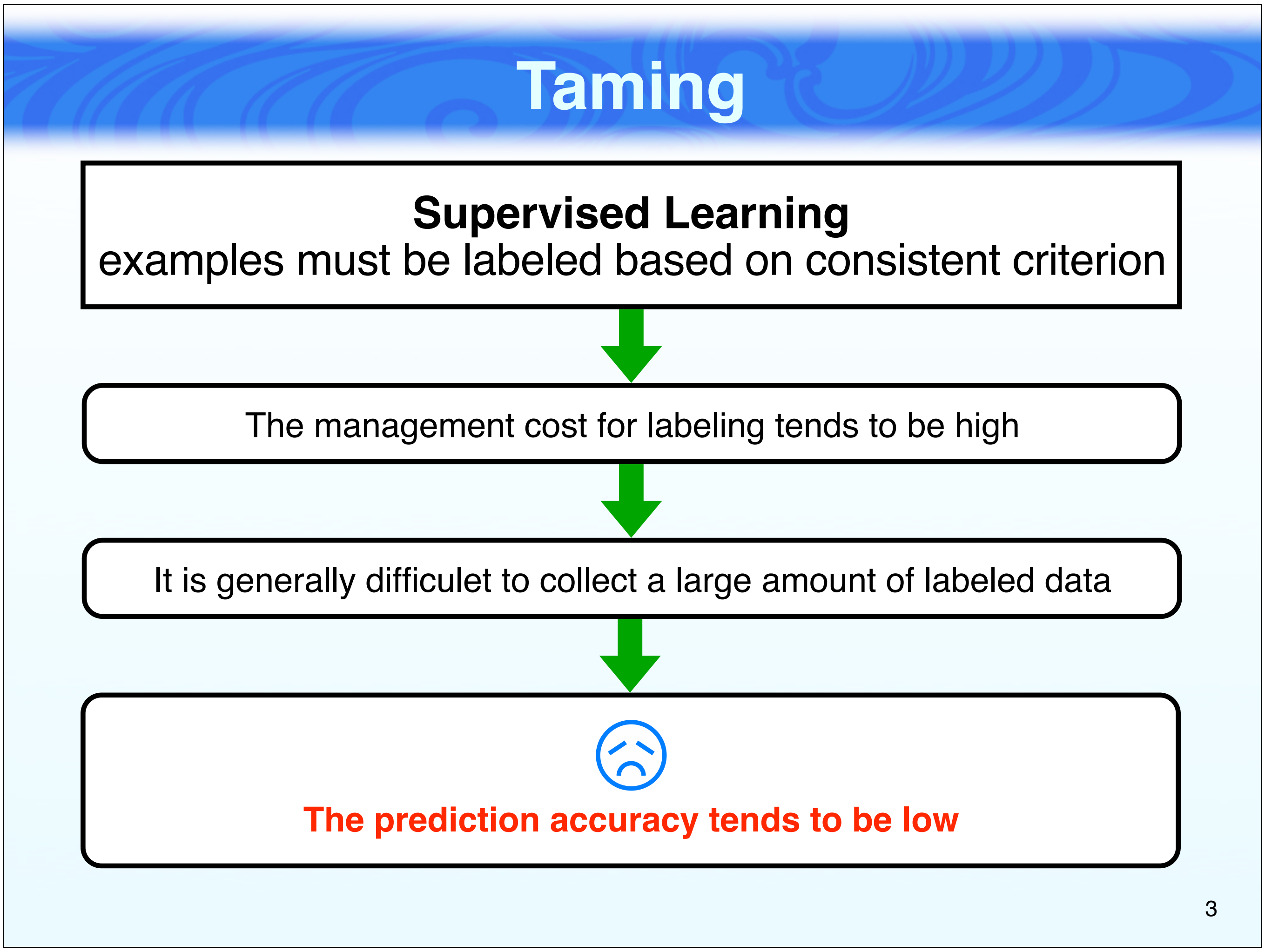
Toshihiro Kamishima, Masahiro Hamasaki, and Shotaro Akaho
National Institute of Advanced Industrial Science and Technology (AIST)
http://www.kamishima.net/

Today, I’d like to talk about a learning framework, taming, and its application to the tag personalization.

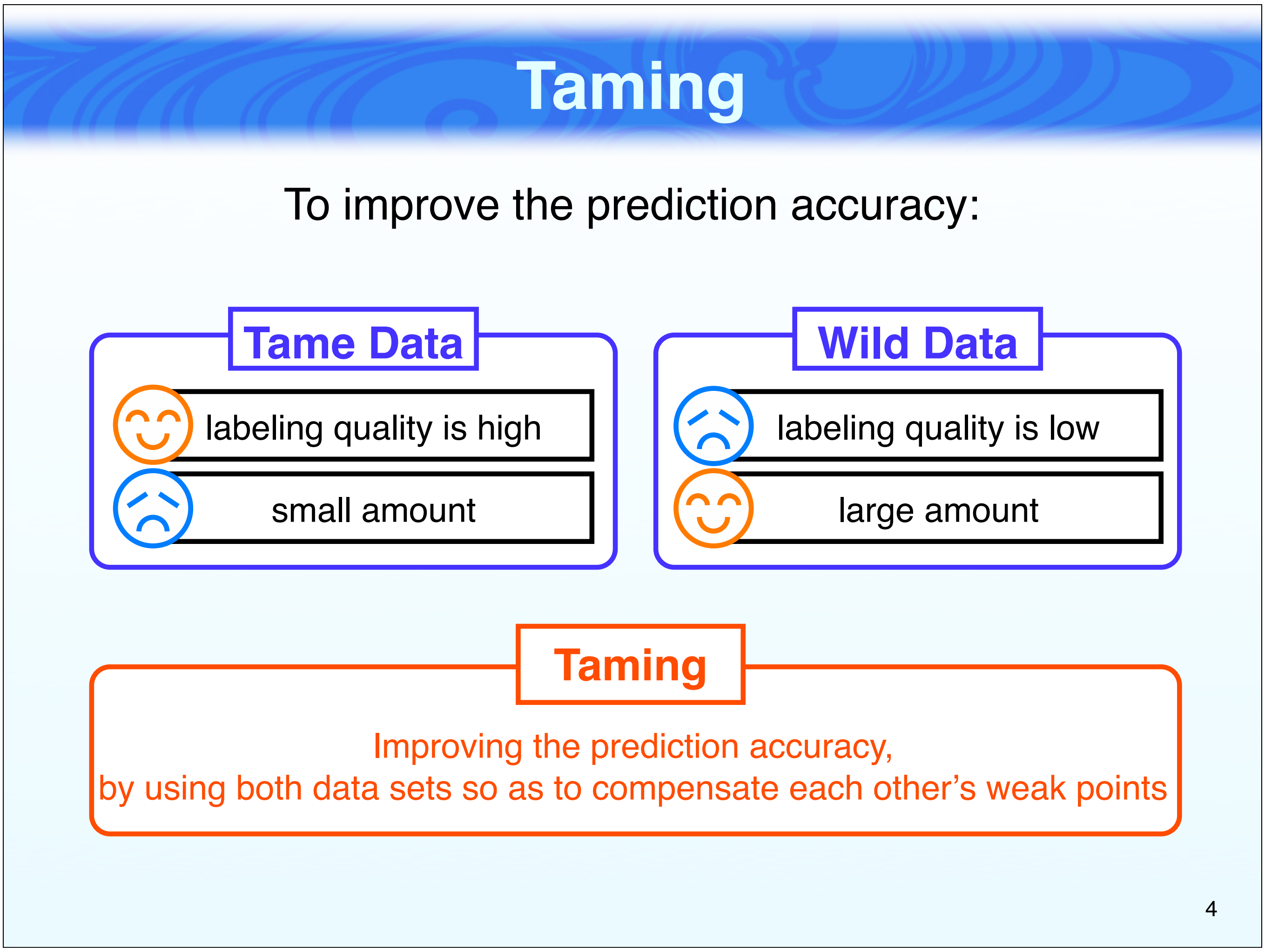
Overview

- **Taming**
  Improve the prediction accuracy by using a small reliable data set together with abundant reliable data
- **BaggTaming**
  Learning algorithm for taming, which is a variant of bagging
- **Collaborative Tagging**
  By applying BaggTaming, improving the prediction accuracy of the tag that is personalized for a specific user in the social bookmarking service, the Hatena bookmark

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, which is a variant of a bagging.
Finally, we apply this BaggTaming to the tag personalization task for the social bookmarking service, the Hatena bookmark.



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.

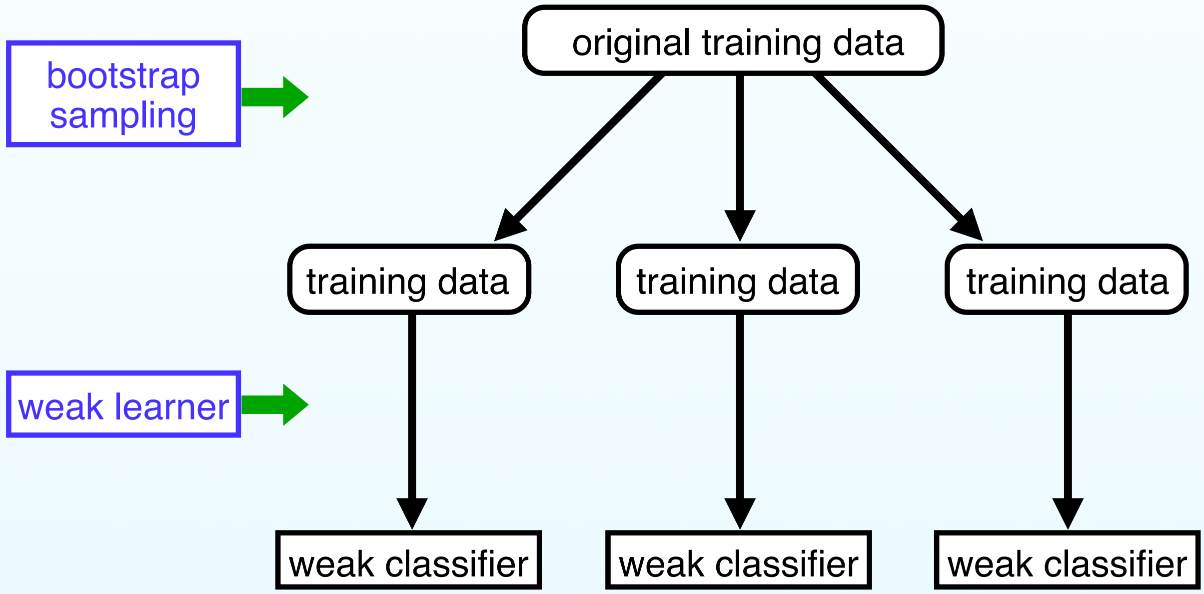


To relieve this difficulty, we propose a learning framework, taming.
We employ two types of training data sets, tame and wild.
The tame data are carefully labeled and so these labels are highly consistent with the target concept. However, due to their labeling cost, a relatively small amount of data are available.
On the other hand, the labeling quality of wild data is low, but these data are much more abundant.
We try to improve the prediction accuracy by using both data sets so as to compensate each other's weak points.
Next, we will talk about a BaggTaming algorithm that is designed for this taming.

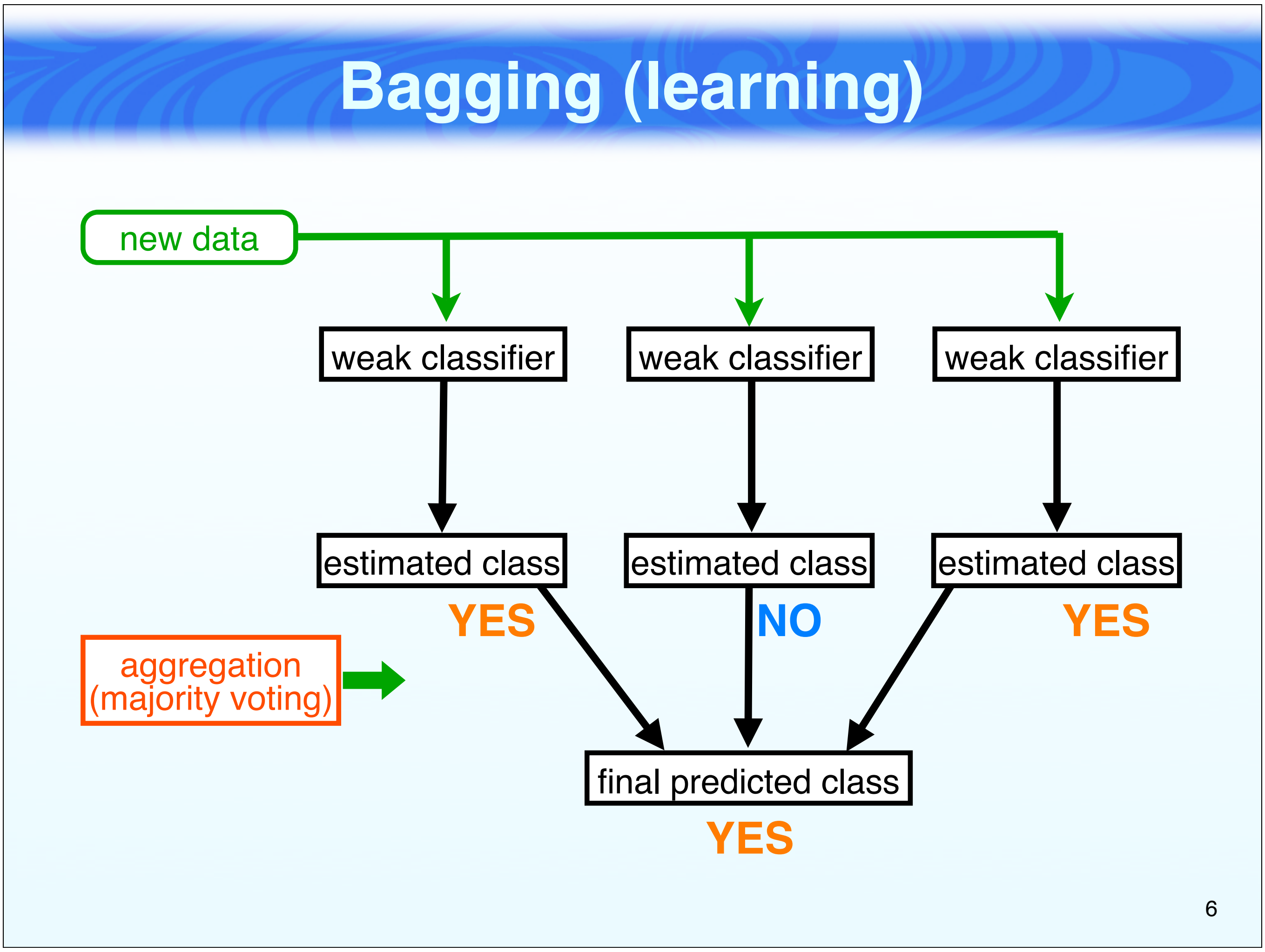
Bagging (learning)

Bagging Bootstrap AGGregatING

Multiple weak classifiers are learned from a bootstrapped training data sets.
The predictions of these classifiers are aggregated.



We begin with bagging, because our BaggTaming is a variant of a bagging.
Briefly speaking, multiple weak classifiers are learned from bootstrapped training data sets, and the predictions of these classifiers are aggregated.
More specifically, multiple training data sets are first generated by bootstrap sampling from original training data.
Each training set is inputed 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.

Bias-Variance Trade-off

Bias-Variance Trade-off
Generalization Error = Bias + Variance + Noise

- **Bias:** error depending on the model complexity
- **Variance:** error resulted from the sampling of training data
- **Noise:** intrinsically irreducible

How does bagging reduce the generalization error?

- **Bias:** controled by the complexity of weak learners
- **Noise:** impossible to remove by definition

Training weak learners by various types of data
contributes to reduce the variance

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 the error depending on the model complexity.
The variance is the error resulted from the sampling of training data, and the noise is intrinsically irreducible.
Generally speaking, by increasing the model complexity, the bias can be reduced, but the variance is increased at the same time, and vice versa.
How does bagging reduce the generalization error?
Bagging cannot reduce the bias and noise because of these reasons: Bias is controlled by the complexity of weak learners and noise is impossible to remove by definition.
However, training weak learners by various types of data contributes to reduce the variance. In summary, bagging is a technique to reduce the variance without sacrificing the bias.

BaggTaming (idea)

To more drastically reduce variance,
Classifiers should be learned from more various types of examples

↓

Training examples are sampled from the wild set,
because it contains more diverse data

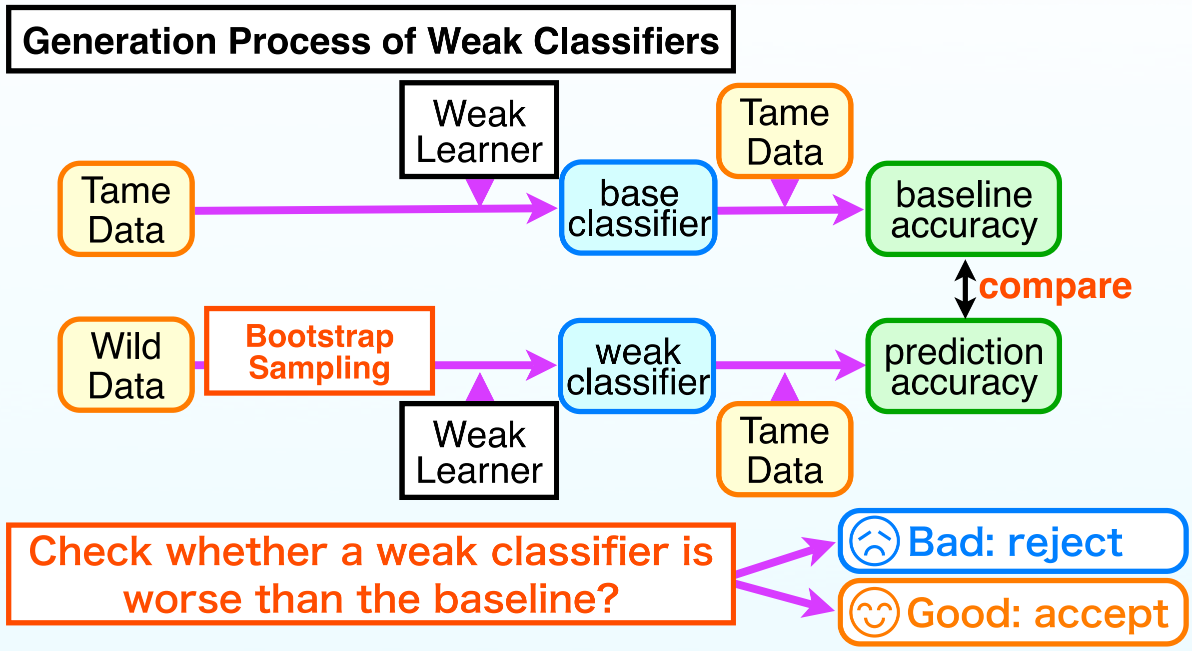
Because the wild set may contain many non-target data,
these non-target data have to be filtered out

↓

Weak classifiers are filtered out,
if the prediction accuracy on the tame set is low

Based on the theory of bias-variance trade-off, we discuss the idea of our BaggTaming.
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.
However, we now face another difficulty. Because the wild set may contain many non-target data, these non-target data have to be filtered out.
For this purpose, we use tame data. Weak classifiers are filtered out if the prediction accuracy on the tame set is low.

Weak Classifiers of BaggTaming



- By repeating this process, multiple classifiers are genarated
- If no classifier is accepted, we use the base classifier as default

We show the detail of the generation process of weak classifiers in our BaggTaming. Before learning weak classifiers, we compute the baseline accuracy. Weak learner acquires a base classifier from the tame 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.
This accuracy is compared with the baseline. If it is statistically worse than the baseline, the candidate weak classifier is rejected; otherwise, it is accepted.
By repeating this process, multiple classifiers are generated. If no classifier is accepted, we use the base classifier as default.
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.

Collaborative Tagging

Social Bookmark 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**

Shared tags can be exploited
for classifying or retrieving Web pages

We first talk about collaborative tagging, such as a social bookmarking service.
In this service, users can register their favorite Web pages. To these Web pages, users can assign 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 freely

Tags are generally inconsistent among 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: window

a hole in the wall	the pane of glass that resides within a hole in the wall

These tags make it difficult to find relevant pages

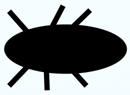
In collaborative tagging, users can assign any tags freely. Therefore, tags are generally inconsistent among users. Golder et al. pointed out three causes for this inconsistency.
The first one is a polysemous word.
Polysemy refers a word having multiple related meanings. For example, one use a word “window” to refer a hole in the wall. But, another refers the pane of glass that resides within the hole.
Because the related meanings confuse users,these tags make it difficult to find relevant pages.

Inconsistent Tags

level of the specificity

The term that describes an item vary along a hierarchy of specificity ranging from very general to very specific
basic level: the level that most users choose as the specificity level

Example:



You find a black intruder in a kitchen

NOT basic level	Yow! Arthropod (節足動物)!
basic level	Oops! Bugs! Terrible roach!

Users may select the different level of the specificity

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!” It cannot be considered as the basic level.
However, one would say “Oops! Bugs!”, or “Terrible roach!” Both terms can be considered as the basic level.
Users may select the different level of the specificity, and different tags can be used to attribute the same aspect of the 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 a synonymous word. 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.

Tag Personalization

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

⬇

Shared tags cannot perfectly meet everyone’s needs

⬇

Tag Personalization

Assignment of tags designed for a specific user

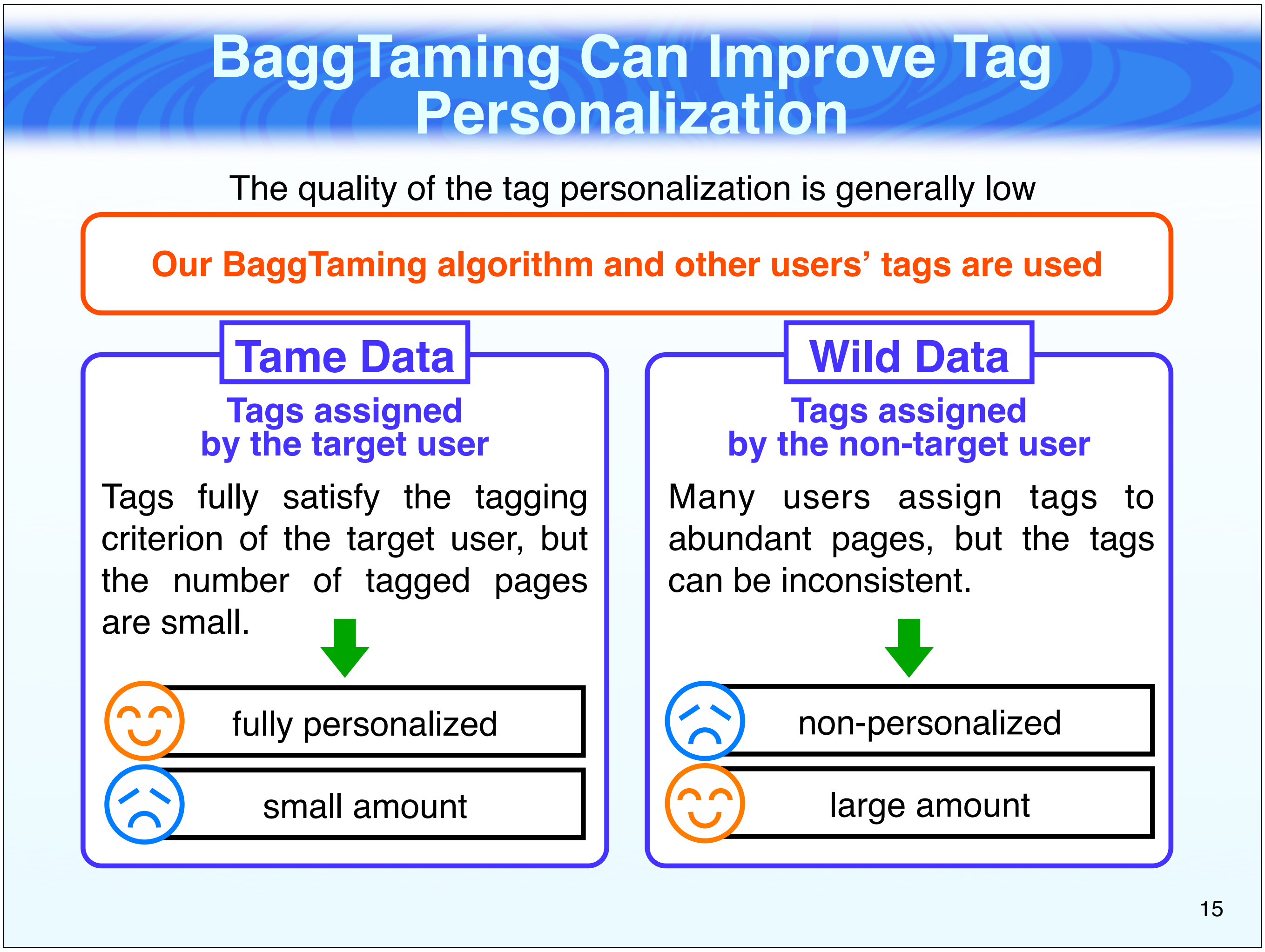
To find the preference pattern of a specific user,
analyzing the tags that have been tagged by the user before

⬇

The number of such tags is generally small

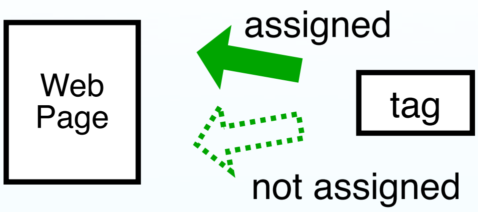
The quality of the tag personalization is generally low

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 we have to do is analyzing the tags that have been tagged by the user before. Unfortunately, the number of such tags is generally small. Consequently, the quality of the tag personalization is generally low. In the context of the recommendation, this problem is called by “a cold-start problem.”



To relieve this difficulty, our BaggTaming algorithm and other users’ tags are used.
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 are not personalized, but are much more abundant.
Next, we empirically show the improvement of the prediction by employing our BaggTaming.

Tag Prediction



Given a Web page and a tag, identify whether the tag should be assigned to the Web page

classification problem

For a specific user and a tag, this task can be stated as classification

- **Class:** binary, the tag should be assigned / not assigned
- **Features:** the number of other tags assigned to the target Web page

NOTE: We didn't use texts of Web pages to avoid the difficulty in cleaning terms that are irrelevant to the content of pages

NOTE: As a weak classifier, we adopt a naive Bayes with Multinomial model

The tag personalization is formalized as a classification task. Given a Web page and a tag, identify whether the tag should be assigned to the Web page. A class is binary, whether a specific tag is assigned or not-assigned. As features, we adopted the number of other tags assigned to the target Web page.

Tag Prediction

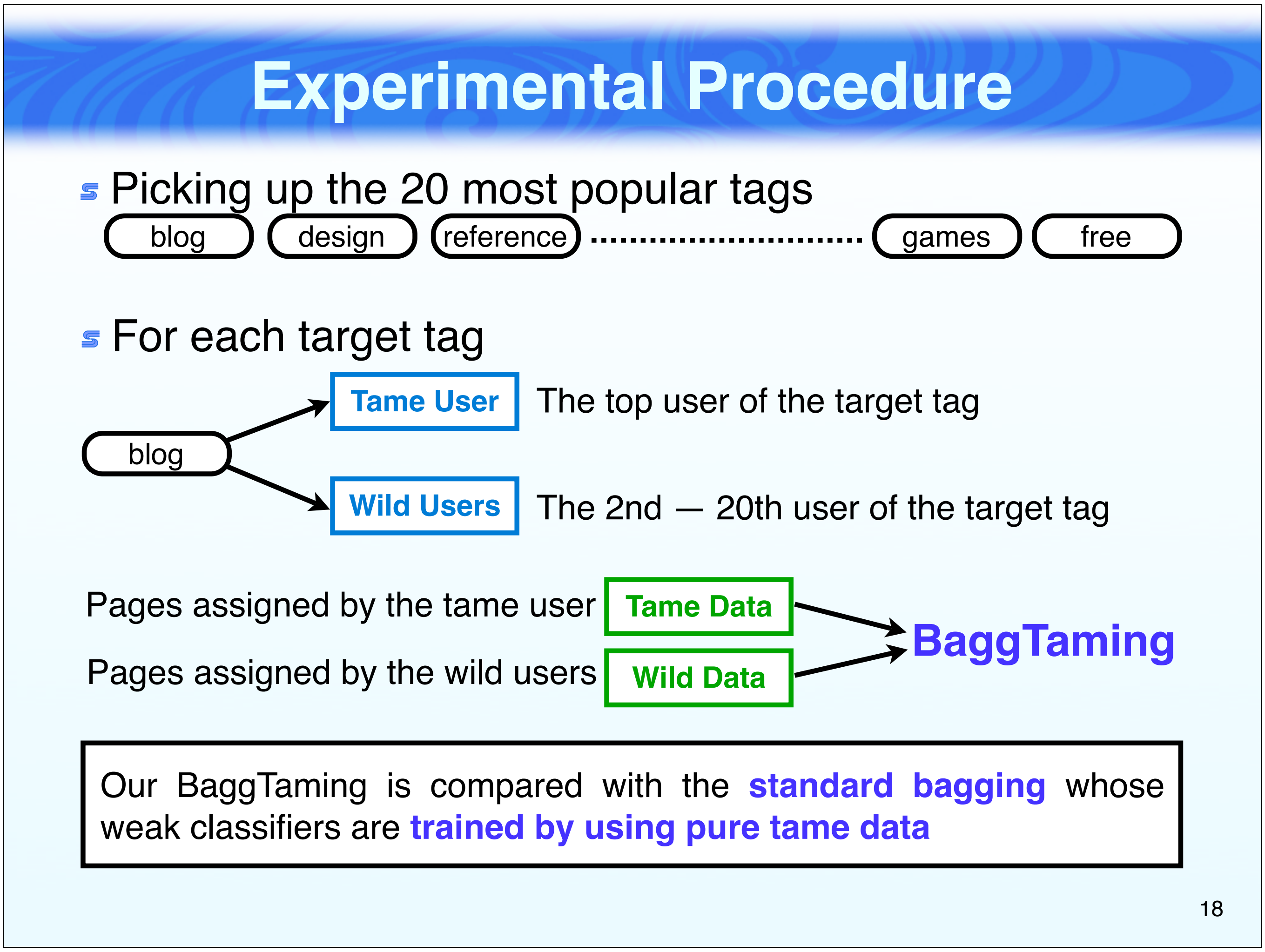
For **the target user** and
for each tag in **a set of candidate tag**,
weak classifiers are learned by using a BaggTaming technique

The system can predict whether each tag is appropriate for a given Web page in the target user's view.

- A user can retrieve or categorize Web pages based on tags personalized to the user
- When a user try to assign tags to a new Web page, candidate tags tailored to the user can be suggested

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.

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.
These personalized tags can be used for the purpose like these:
A user can retrieve or categorize Web pages based on tags personalized to the user
When a user try to assign tags to a new Web page, candidate tags tailored to the user can be suggested



We next show the 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. Tags are personalized by using both data sets together with our BaggTaming.
Our BaggTaming is compared with those of the standard bagging whose weak classifiers are trained by using pure tame data.
In our previous work, we showed the effectiveness of our approach on the delicious.com data. In this presentation, we performed the same experiment on the data collected from the Hatena bookmark.

Tag Data Overview

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

Tag	Tame	Wild	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

Compared with the delicious data, data sets are much more abundant

These are 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.
Compared with the delicious data, both types of data sets are much more abundant. This is likely because the Hatena provides the functionality of suggesting the tags that have already been assigned.

Experimental Results

Size of tame data	ALL	1/2	1/4	1/8	1/16	1/32
Hatena bookmark (BT/Bagg)	0/2	2/3	6/5	7/5	8/3	12/4
delicious.com (BT/Bagg)	5/2	8/3	8/2	10/2	11/1	

- **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**

Each cell shows the number of tags that our BaggTaming wind/loses among 20 tags

While fixing the size of the wild sets, the sizes of tame sets are gradually reduced from “ALL” to “1/32”

- Compared with the delicious case, our BaggTaming is less efficient in this Hatena case due to the abundance of tame data
- Our approach is clearly useful when the sizes of tame data sets are small

This is a summary of experimental results. The prediction accuracies on the tame data sets are compared.
BT and Bagg mean the results of our BaggTaming and a standard bagging. Each cell shows 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.
Compared with the delicious case, our BaggTaming is less efficient due to the abundance of tame data
However, our approach is clearly useful when the sizes of tame data sets are small. This is exactly the situation where our approach is required.

Inductive Transfer

Taming is one of variants of inductive transfer

Inductive transfer refers to the problem of retaining and applying the knowledge learned in one or more tasks to efficiently develop an effective hypothesis for a new task

Examples of techniques for inductive transfer

- Learn a hyper prior that are common for all relevant tasks
- Neural networks having a hidden layer shareed by multiple tasks
- Less weighing the relevant sub tasks than the target main task
- Building a mixture model of the main and relevant tasks

- **inductive transfer:** using the data of other related domain or task. The labeling may be inconsistent in the target domain, but is onsistent in the related domain.
- **taming:** wild data consists of a mixture of data in the target domain and unknown irrelevant domain.

NOTE: We also tested mixture model approach, but failed

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 standard 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

Misc

- Homepage: http://www.kamishima.net/
- 朱鷺の杜Wiki: http://ibisforest.org/index.php?FrontPage

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.