TrBagg: A Simple Transfer Learning Method and Its Application to Personalization in Collaborative Tagging

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

ICDM2009 @ Miami, USA, 6-9/12/2009

START

Today, I'd like to talk about TrBagg, which is a simple technique for transfer learning and its application to personalization in collaborative tagging.

Outline

Collaborative Tagging (CT)

- An introduction of CT
- three properties of tags in CT
- personalization in tag recommendation

Transfer Learning

- What is transfer learning

TrBagg

- bagging, which is a base method of our TrBagg
- TrBagg algorithm

Experimental Results

- Tag personalization by using TrBagg
- Experimental Results
  vs Bagging, vs Other TL methods, and Two filtering methods

This is an outline of our talk.
We begin with talking about collaborative tagging, and personalization in tag recommendation.
Second, we briefly introduce transfer learning.
Third, we show our TrBagg together with bagging.
Finally, we talk about the tag personalization by using TrBagg, and show several experimental results.

Collaborative Tagging

Collaborative Tagging (CT)

- An introduction of CT
- three properties of tags in CT
- personalization in tag recommendation

Transfer Learning

- What is transfer learning

TrBagg

- bagging, which is a base method of our TrBagg
- TrBagg algorithm

Experimental Results

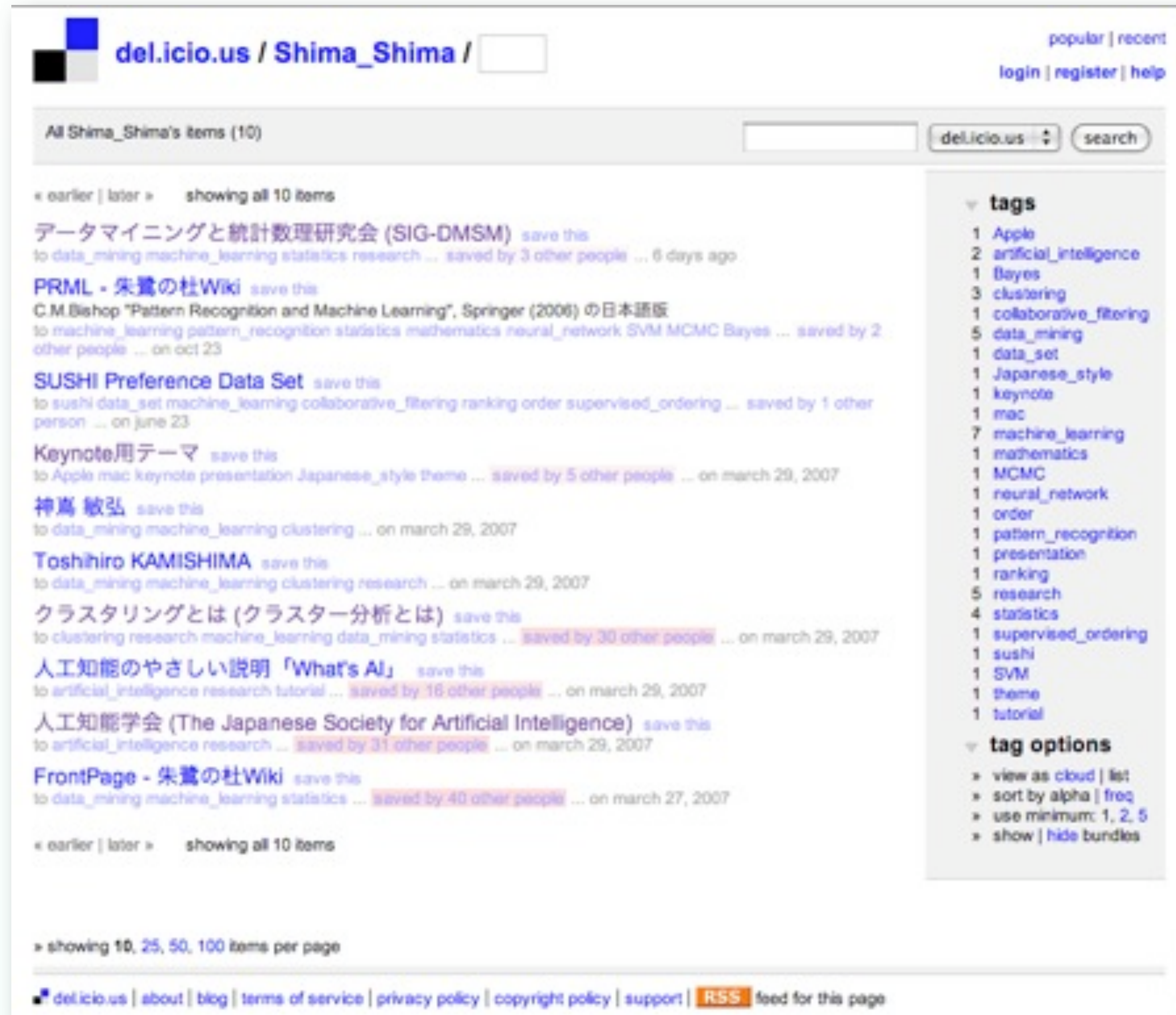
- Tag personalization by using TrBagg
- Experimental Results
  vs Bagging, vs Other TL methods, and Two filtering methods

Let’s move on to collaborative tagging

Collaborative Tagging

Social Bookmarking 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

One example of collaborative tagging is a social bookmarking service.
By 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 freely assign 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:

window

a hole in the wall

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

It is difficult for a user to identify relevant pages

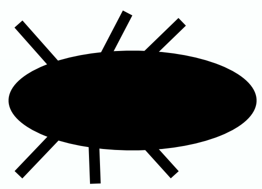
In collaborative tagging, users can freely assign tags based on their own criteria. Therefore, tags are generally inconsistent among different users. Golder et al. pointed out three causes for this inconsistency.
The first one is a polysemous word.
Polysemy refers a word having multiple related meanings. 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.
Due to confusing meanings, it is difficult for users 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 level of specificity



You find a black intruder in a kitchen

improper specificity level	Yow! Arthropod!
proper specificity level	Oops! Bugs! Terrible roach!

Both are basic level!

Users may select the different level of specificity, and different tags can be assigned to the same page

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 level of specificity. But, the basic level may not be unique.
For example, you find a black intruder in a kitchen. No one screams academic names. 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 would be referred by different tags

Semantics of tags or the tag selection criteria differ 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 the multiple words having the same meaning. For example, television and TV. In this case, the same item would be referred by different tags.
Semantics of tags or the tag selection criteria differ 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 satisfy 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 are generally small

The quality of the tag personalization is generally low

In such a case, shared tags cannot perfectly satisfy everyone's needs. So, we need 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 have been tagged by the user before. Unfortunately, the number of such tags are generally small. Due to the lack of information, the quality of the tag personalization is generally low. In a context of recommendation, this problem is called by "a cold-start problem."

Transfer Learning

- **Collaborative Tagging (CT)**
  - An introduction of CT
  - three properties of tags in CT
  - personalization in tag recommendation
- **Transfer Learning**
  - What is transfer learning
- **TrBagg**
  - bagging, which is a base method of our TrBagg
  - TrBagg algorithm
- **Experimental Results**
  - Tag personalization by using TrBagg
  - Experimental Results
    vs Bagging, vs Other TL methods, and Two filtering methods

We have introduced collaborative tagging.
We next move onto talking about transfer learning.

Transfer Learning

Not formally defined. But, broadly speaking ...

Source domain (data, knowledge)		Target domain
the task related to the target task	↔	**the task that we want to solve**
containing **the knowledge related to the target** task	↔	The amount of knowledge or data is **insufficient**

Improving the prediction performance in the target domain
by exploiting the knowledge or data of the related source domain

Broad definition of transfer learning is as follows:
We consider two domains of tasks: target and source.
We want to solve the task of the target domain. On the other hand, we don't need to solve the task of the source domain, but the source task is related to that of the target domain.
Transfer learning refers to the problem of Improving the prediction performance in the target domain by exploiting the knowledge or data of the related source domain.

Our Assumption

Improving the prediction performance in the target domain
by exploiting the knowledge or data of the related source domain

Related domain, what?

Each transfer learning method assumes its own relatedness

Our assumption

Source Domain

non-target data

target data

Mixture of non-target and target data

Target Domain

target data

Purely consist of target data

Here is one question: what's the related domain? Each transfer learning method assumes its own relatedness.
So, we clarify our assumption.
We assume that a data set of the target domain purely consists of target data.
On the other hand, A data set of the source domain is a mixture of non-target and target data.

Transfer Learning

Collaborative Tagging (CT)

- An introduction of CT
- three properties of tags in CT
- personalization in tag recommendation

Transfer Learning

- What is transfer learning

TrBagg

- bagging, which is a base method of our TrBagg
- TrBagg algorithm

Experimental Results

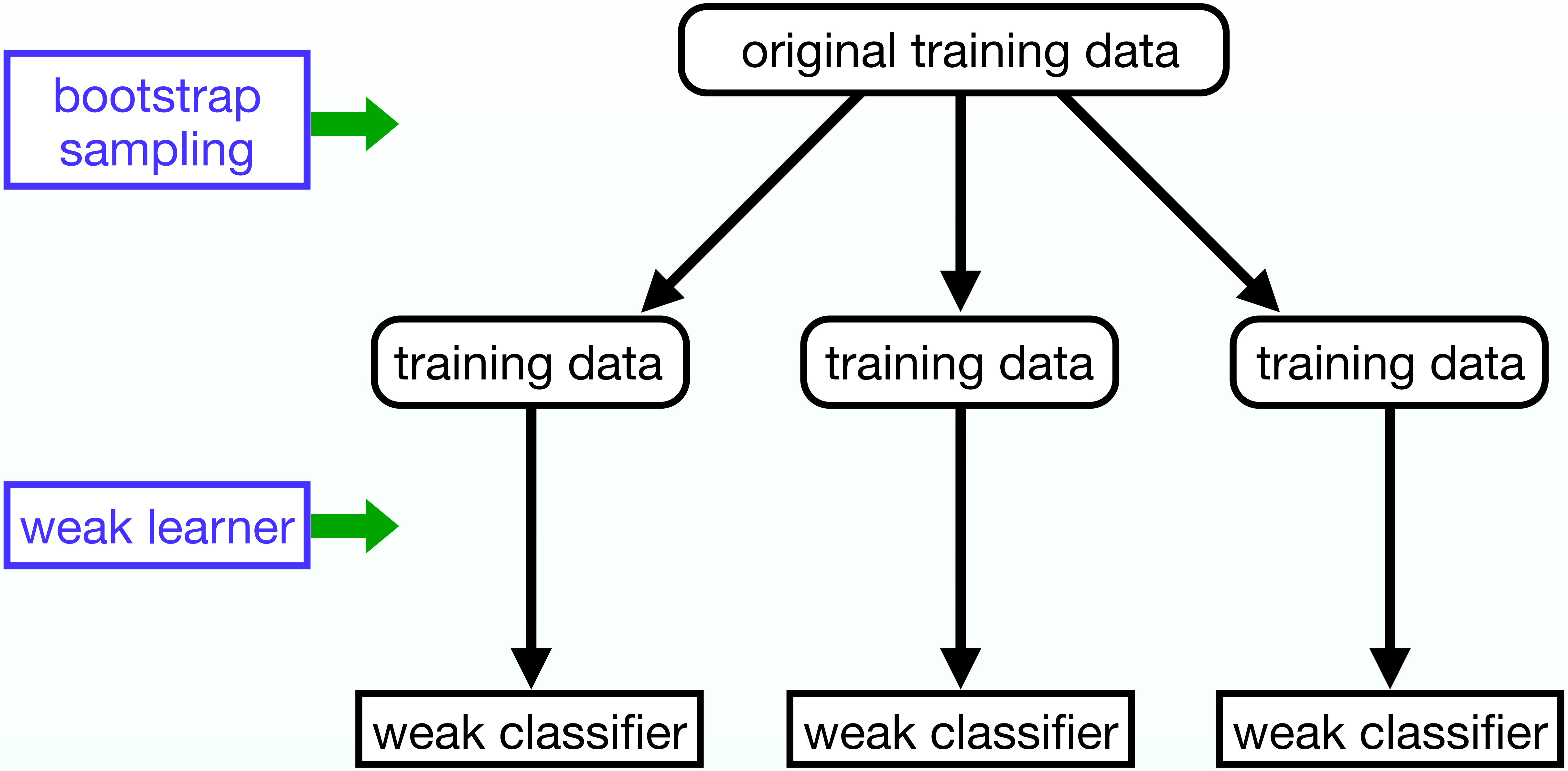
- Tag personalization by using TrBagg
- Experimental Results
  vs Bagging, vs Other TL methods, and Two filtering methods

Now, we have shown what transfer learning is.
Next, we move onto our main topic: TrBagg.

Bagging (learning)

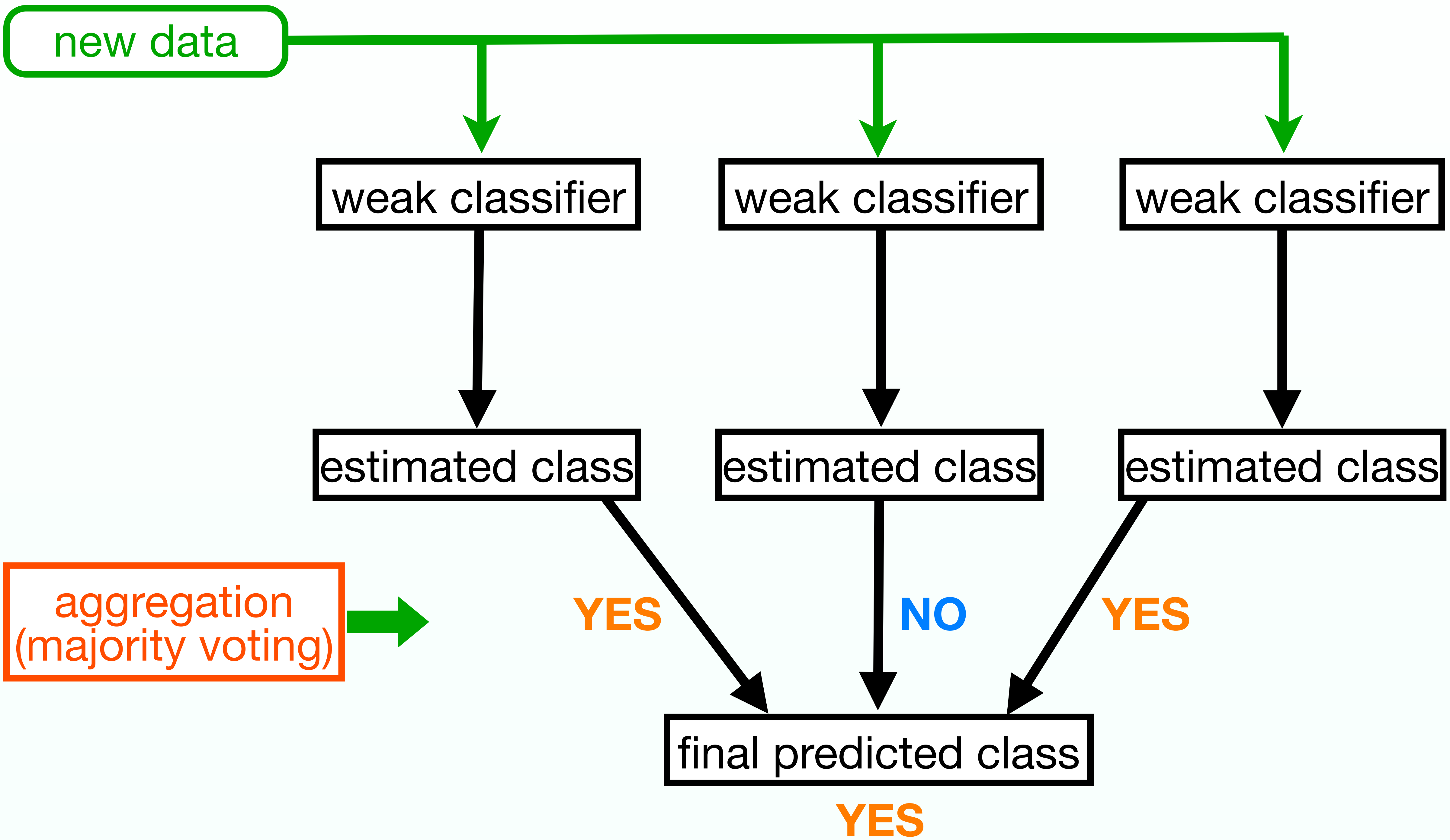
Bagging Bootstrap AGGregatING

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



We begin with bagging, because our TrBagg is a variant of this bagging.
Briefly speaking, multiple weak classifiers are learned from bootstrapped training sets, and the predictions of these classifiers are aggregated.
More specifically, 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 acquired. Any supervised learning methods, such as naive Bayes or SVM, can be used as a weak learner.

Bagging (prediction)



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 own estimated class.
The final class is determined by majority voting of these estimated classes.

Bias-Variance Theory

Bias-Variance Theory

Generalization Error = Bias + Variance + Noise

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

↓

How does bagging reduce the generalization error?

- **Bias**: this type of error cannot be reduced without changing the model of weak learners
- **Noise**: impossible to remove by definition

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

Breiman showed the reason why the prediction accuracy is improved by bagging based on the bias-variance theory.
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.
Bagging cannot reduce bias and noise because of these reasons, but training weak learners by diverse types of data contributes to reduce the variance.
In summary, bagging is a technique to reduce variance without sacrificing bias.

TrBagg (idea)

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

↓

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

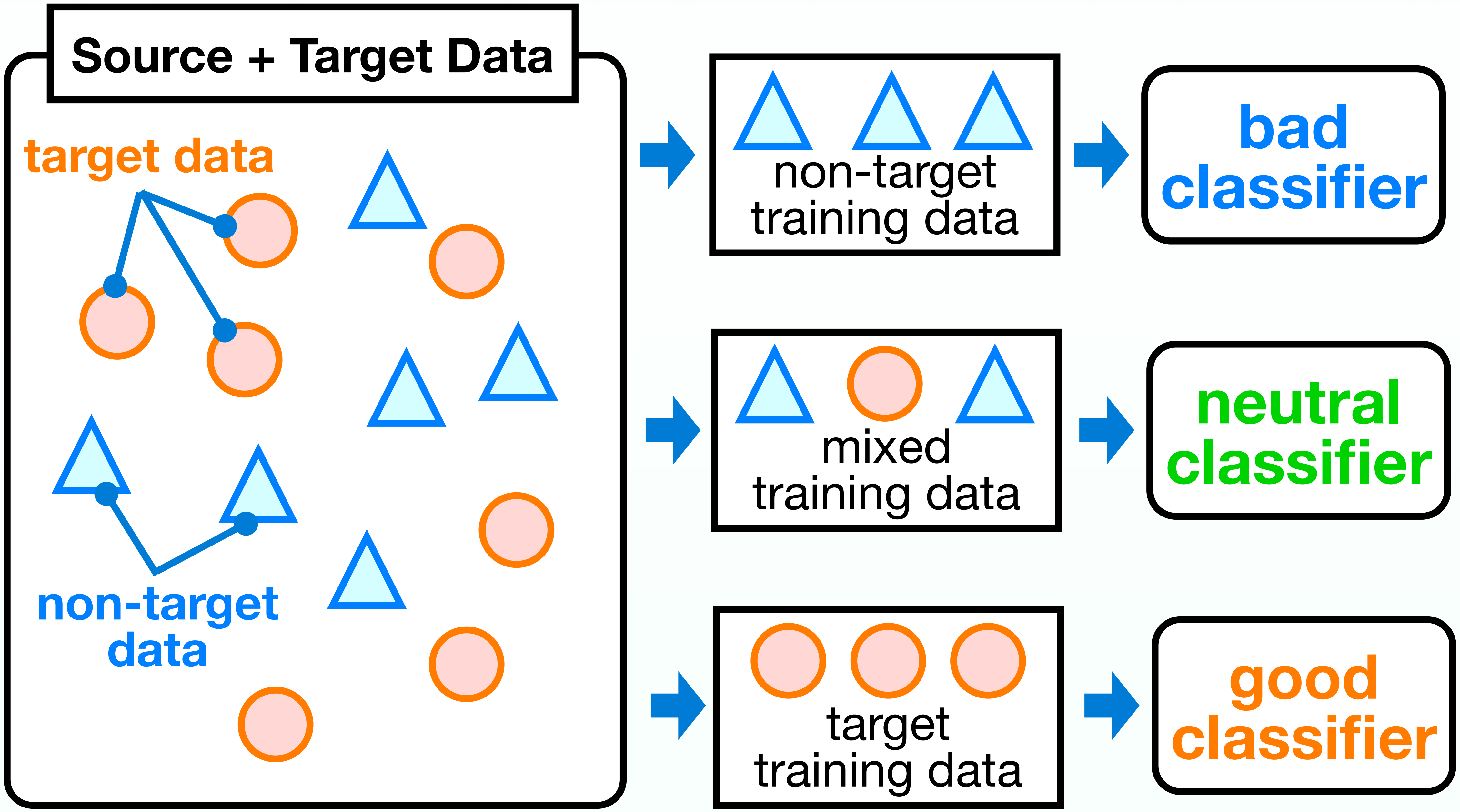
Because the source data set contains many non-target data,
these non-target data have to be filtered out

↓

Weak classifiers are filtered out
if they are poorly performed on the target data

Based on this bias-variance theory, we discuss the idea of our TrBagg.
In order to more drastically reduce the variance, classifiers should be learned from more diverse types of examples. For this purpose, training examples are sampled from the source data, because it contains more diverse data.
But, we now face one difficulty. Because the source data set contains many non-target data, these data have to be filtered out. For this purpose, we use the target data set. Weak classifiers are filtered out if they are poorly performed on the target data.

TrBagg (training)



Assumption: source data is a mixture of target and non-target

We don’t know **which is good**

Let me talk about the training phase of our TrBagg.
Training data sets for weak classifiers are bootstrap-sampled from the merger of the source and target data.
If the training data set happens to consist of non-target data, bad classifiers would be learned.
Contrarily, if the training data set happens to consist of target data, good classifiers would be learned.
But now, we face the difficulty: we don’t know which is GOOD!

TrBagg (filtering)

Filtering Stage

Bad classifiers are filtered out
based on the prediction performance on the target data

MVT (Majority Voting on the Target set)

By using greedy search, find a set of classifiers to maximize the accuracy on the target data **including** those used for training

- Minimize empirical error on the target data and risk of over-fitting
- perfoming well for small target data sets in a practical use.

MVV (Majority Voting on the Validation set)

By using greedy search, find a set of classifiers to maximize the accuracy on the target data **excluding** those used for training

- Minimize generalization error on the test data, which we want to reduce
- Theoretical advantage

In the filtering stage, bad classifiers are filtered out based on the prediction performance on the target data.
We tested two types of filtering methods: MVT and MVV.
In a case of MVT, we find a set of classifiers to maximize the accuracy on the target target data including those used for training.
In a case of MVV, data used for training are excluded.
In other words, MVT tries to minimize empirical error on the target data, while MVV tries to minimize generalization error on the test data.
MVV is theoretically superior, but in a practical use, MVT performed well for the small set of the target data.

Transfer Learning

Collaborative Tagging (CT)

- An introduction of CT
- three properties of tags in CT
- personalization in tag recommendation

Transfer Learning

- What is transfer learning

TrBagg

- bagging, which is a base method of our TrBagg
- TrBagg algorithm

Experimental Results

- Tag personalization by using TrBagg
- Experimental Results
  vs Bagging, vs Other TL methods, and Two filtering methods

Now, we have introduced our TrBagg algorithm.
We finally show tag personalization by using TrBagg and experimental results.

TrBagg Can Improve Tag Personalization

The quality of the tag personalization is generally low

To relieve this difficulty, our TrBagg and other users' tags are used

Source Data

Tags assigned by the non-target user

Many users assigned tags to abundant pages, but the tags can be inconsistent.

- non-personalized
- large amount

Target Data

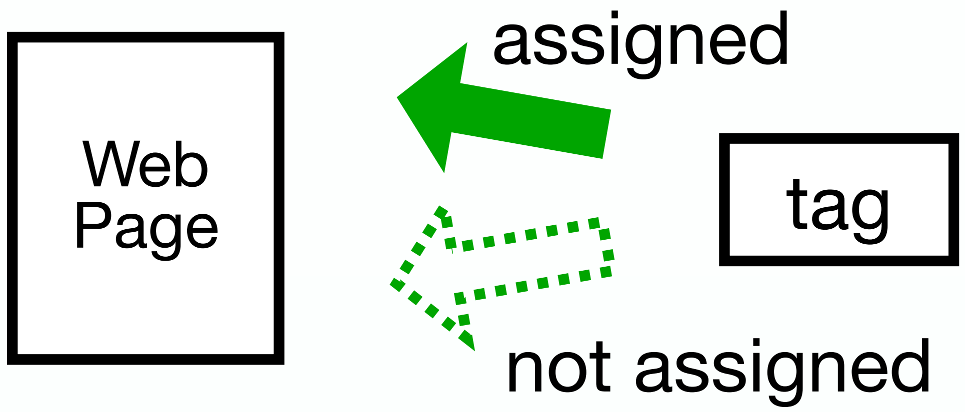
Tags assigned by the target user

Tags fully satisfy the tagging criterion of the target user, but the number of tagged pages are small.

- fully personalized
- small amount

We have shown that the quality of the tag personalization is generally low due to the lack of information.
To relieve this difficulty, we use our TrBagg and other users' tags.
Tags assigned by the target user are considered as the target data. Tags fully satisfy the tagging criterion of the target user, but the number of tagged pages are small.
On the other hand, tags assigned by the non-target user are considered as source data. Many users assign tags to abundant pages, but the tags can be inconsistent.

Tag Prediction



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

classification problem

- **Class**: a specific tag should be assigned or not for a specific user
- **Features**: the number of non-target tags assigned to the target Web page

NOTE: We currently didn't use text information of Web pages to avoid the difficulty in cleaning terms, but such information will be able to use straightforwardly

NOTE: As weak classifiers, we used naive Bayes classifiers using multinomial distributions

The personalized tag prediction task is formalized as classification. Specifically, a class indicates whether a specific tag should be assigned or not for a specific user. In other words, we adopted one-vs-rest encoding. As features, we adopted the number of non-target tags assigned to the target Web page.

Tag Prediction

For each tag in a set of candidate tag,
a set of weak classifiers are learned

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

NOTE: Learning classifiers for every candidate tags is computationally expensive. We plan to introduce multi-label classifiers to remedy this drawback

For each tag in a set of candidate tag, weak classifiers are learned. Once weak classifiers are learned, the system can predict whether each tag is appropriate for a given Web page in the target user's view by performing the filtering stage of TrBagg.

Merits of This Approach

Compared to other approaches for tag personalization, our approach using transfer learning is rather computationally intensive

Suited for Parallel Computation

Once sub-samples are given, weak classifiers can be learned in parallel

↓

Our approach is suited for **non-centerized environments**, that are composed of sparsely connected and distributed machines

Private Tag Personalization

Once a common set of weak classifiers are learned, a filtering stage can be performed purely based on user's own data

↓

Tags can be **privately personalized** within local machines

Compared to other approaches for tag personalization, our approach using transfer learning is rather computationally intensive. However, our approach has these merits.
Firstly, ours are suited for parallel computation, because once sub-samples are given, weak classifiers can be learned in parallel.
Secondly, tags can be privately personalized, because Once a common set of weak classifiers are learned, a filtering stage can be performed purely based on user's own data.

Experimental Procedure

- Picking up the 20 most popular tags

blog | design | reference .... free | webdesign

- For each target tag

blog → **Target User** — The top user of this tag

blog → **Source Users** — The 2nd — 20th user of this tag

Pages assigned by the target user — **Target Data** → **TrBagg**

Pages assigned by the source users — **Source Data** → **TrBagg**

Prediction accuracies of the TrBagg is compared with those of the **standard bagging** whose weak classifiers are **trained by using only target data**

We next show our 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 target user, and the second to twentieth users are source users.
Pages assigned by the target user and source users are treated as target and source data, respectively.
Prediction accuracies of the TrBagg is compared with those of the standard bagging whose weak classifiers are trained by using target data.

Tag Data Overview

Sizes of the target and source data sets of the 20 target tags of the **delicious** data

Tag	Target	Source	Tag	Target	Source
blog	603	24201	web2.0	917	25256
design	1405	25353	politics	5455	21857
reference	6323	19512	news	67	28385
software	3512	30264	howto	6359	23335
music	6311	22914	linux	1151	24288
programming	4498	25931	blogs	3472	18437
web	1291	31024	tutorial	3518	28593
tools	3493	23625	games	3218	22588
video	1870	30334	free	3509	23543
art	6258	16574	webdesign	1098	25427

We also tested data from **hatena** (Japanese social bookmark service)
This data set contain more target data than delicious data

We collected tag data from two social bookmarking sites: delicious and hatena.
This table shows the sizes of the target and source data sets of the 20 target tags of the delicious data.
You can see that the number of the target data is much smaller than that of the source data.
The hatena data set contain more target data than delicious data

Results: vs baseline Bagging

of target tags for which TrBagg won/lost against baseline bagging

# of target data	many ↔ small					
delicious	3/0	6/0	7/0	8/0	10/0	11/0
hatena	2/0	4/1	7/1	7/3	9/2	10/2

“X/Y” shows # of counts that the prediction accuracies of our TrBagg is higher/lower than those of a standard bagging among 20 tags

While fixing the size of the source data sets, the size of target sets are reduced by subsampling

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

This is a summary of experimental results.
We show the number of counts that the prediction accuracies of our TrBagg is higher/lower than those of a standard bagging among 20 tags. For example, “3/0” means that our TrBagg is significantly superior to a standard bagging in three tags, significantly inferior in zero tags, and tied in 17 tags.
Further, while fixing the size of the source data sets, the size of target sets are reduced by subsampling.
It would be reasonable to say that these two conclusions:
Our BaggTaming is constantly superior to the bagging.
The advantage of our TrBagg becomes clearer as the number of tame data lessen.

Results: vs Other TL Methods

	delicious		hatena	
	Large Target Data	Small Target Data	Large Target Data	Small Target Data
TrAdaBoost	9/1	11/0	8/6	13/3
frustratingly easy	10/0	9/0	14/4	17/3

# of counts that TrBagg wins/loses against two transfer learning methods among 20 tags

- **TrAdaBoost**: AdaBoost modified for transfer learning
  This method worked well, if target data are abundant and source data are well related.
- **frustratingly easy**: simple approach of converting feature vectors
  This method assumes that part of features of source data are useful for target task

We next compared our TrBagg with other transfer learning methods.
Again, we showed the win and lose counts of our TrBagg.
TrAdaBoost is AdaBoost modified for transfer learning. This method worked well, if target data are abundant and source data are well related. But, in tag prediction task, the target data are generally insufficient.
A frustratingly easy method is a simple approach of converting feature vectors. This method assumes that part of features of source data are useful for target task. But, this assumption is not the case for tag prediction task.

Results: Two Filtering Methods

	delicious		hatena	
	Large Target Data	Small Target Data	Large Target Data	Small Target Data
MVT/MVV	1/0	3/1	2/3	7/4

Two filtering methods, MVT and MVV, are compared

- **MVT**: find a set of classifiers to maximize the accuracy on the target data **including** those used for training
- **MVV**: find a set of classifiers to maximize the accuracy on the target data **excluding** those used for training

The MVT method is advantageous for the small target data sets, because rare target data must be separated for validation in a case of MVV.

We have presented two filtering methods. In a case of MVT, we find a set of classifiers to maximize the accuracy on the target target data including those used for training, but these are excluded in a case of MVV.
The MVT method is advantageous for the small target data sets, because rare target data must be separated for validation in a case of MVV.

Conclusion

Summary

- **Assumption about the relatedness of transfer learning**
  Prediction accuracy was improved by using a small set of reliable target data together with the abundant source data, which are a mixture of reliable and unreliable data
- **Developing TrBagg algorithm**
  Ensemble learning sampling from both source and target data, and weak classifiers are filtered out by exploiting target data
- **Application to collaborative tagging data**
  Personalized tags are more precisely predicted by adopting our TrBagg technique

Our conclusions are as follows

Future Work

- **Multi-label Classifiers**
  Constructing classifiers for every tags is computationally expensive. Introducing multi-label classifiers would be useful to remedy this drawback.
- **On-Line Learning**
  For the efficient update of classifiers, we'd like to consider the on-line version of TrBagg.
- **Privacy-Preserving**
  In this framework, tags can be privately recommended, once weak classifiers are given. We'd like to develop methods for learning weak classifiers under privacy-preserving condition.

These are our future works.
That's all we have to say. Thank you for attention.