Understanding seed selection in bootstrapping

Yo Ehara†∗ **Issei Sato**‡ **Hidekazu Oiwa**†∗ **Hiroshi Nakagawa**‡
† Graduate School of Information Science and Technology ‡ Information Technology Center
The University of Tokyo / 7-3-1 Hongo, Bunkyo-ku, Tokyo, Japan
∗ JSPS Research Fellow
Kojimachi Business Center Building, 5-3-1 Kojimachi, Chiyoda-ku, Tokyo, Japan
{ehara@r., sato@r., oiwa@r., nakagawa@}dl.itc.u-tokyo.ac.jp

Abstract

Bootstrapping has recently become the focus of much attention in natural language processing to reduce labeling cost. In bootstrapping, unlabeled instances can be harvested from the initial labeled "seed" set. The selected seed set affects accuracy, but how to select a good seed set is not yet clear. Thus, an "iterative seeding" framework is proposed for bootstrapping to reduce its labeling cost. Our framework iteratively selects the unlabeled instance that has the best "goodness of seed" and labels the unlabeled instance in the seed set. Our framework deepens understanding of this seeding process in bootstrapping by deriving the dual problem. We propose a method called expected model rotation (EMR) that works well on not well-separated data which frequently occur as realistic data. Experimental results show that EMR can select seed sets that provide significantly higher mean reciprocal rank on realistic data than existing naive selection methods or random seed sets.

1 Introduction

Bootstrapping has recently drawn a great deal of attention in natural language processing (NLP) research. We define bootstrapping as a method for harvesting "instances" similar to given "seeds" by recursively harvesting "instances" and "patterns" by turns over corpora using the distributional hypothesis (Harris, 1954). This definition follows the definitions of bootstrapping in existing NLP papers (Komachi et al., 2008; Talukdar and Pereira, 2010; Kozareva et al., 2011). Bootstrapping can greatly reduce the cost of labeling instances, which is especially needed for tasks with high labeling costs.

The performance of bootstrapping algorithms, however, depends on the selection of seeds. Although various bootstrapping algorithms have been proposed, randomly chosen seeds are usually used instead. Kozareva and Hovy (2010) recently reports that the performance of bootstrapping algorithms depends on the selection of seeds, which sheds light on the importance of selecting a good seed set. Especially a method to select a seed set considering the characteristics of the dataset remains largely unaddressed. To this end, we propose an "iterative seeding" framework, where the algorithm iteratively ranks the goodness of seeds in response to current human labeling and the characteristics of the dataset. For iterative seeding, we added the following two properties to the bootstrapping;

- criteria that support iterative updates of goodness of seeds for seed candidate unlabeled instances.
- iterative update of similarity "score" to the seeds.

To invent a "criterion" that captures the characteristics of a dataset, we need to measure the influence of the unlabeled instances to the model. This model, however, is not explicit in usual bootstrapping algorithms' notations. Thus, we need to reveal the model parameters of bootstrapping algorithms for explicit model notations.

To this end, we first reduced bootstrapping algorithms to label propagation using Komachi et al.

Proceedings of the TextGraphs-8 Workshop, pages 44–52,
Seattle, Washington, USA, 18 October 2013.

(2008)'s theorization. Komachi et al. (2008) shows that simple bootstrapping algorithms can be interpreted as label propagation on graphs (Komachi et al., 2008). This accords with the fact that many papers such as (Talukdar and Pereira, 2010; Kozareva et al., 2011) suggest that graph-based semi-supervised learning, or label propagation, is another effective method for this harvesting task. Their theorization starts from a simple bootstrapping scheme that can model many bootstrapping algorithms so far proposed, including the "Espresso" algorithm (Pantel and Pennacchiotti, 2006), which was the most cited among the Association for Computational Linguistics (ACL) 2006 papers.

After reducing bootstrapping algorithms to label propagation, next, we will reveal the model parameters of a bootstrapping algorithm by taking the dual problem of bootstrapping formalization of (Komachi et al., 2008). By revealing the model parameters, we can obtain an interpretation of selecting seeds which helps us to create criteria for the iterative seeding framework. Namely, we propose expected model rotation (EMR) criterion that works well on realistic, and not well-separated data.

The contributions of this paper are summarized as follows.

- The iterative seeding framework, where seeds are selected by certain criteria and labeled iteratively.
- To measure the influence of the unlabeled instances to the model, we revealed the model parameters through the dual problem of bootstrapping.
- The revealed model parameters provides an interpretation of selecting seeds focusing on how well the dataset is separated.
- "EMR" criterion that works well on not well-separated data which frequently occur as realistic data. .

2 Related Work

Kozareva and Hovy (2010) recently shed light on the problem of improving the seed set for bootstrapping. They defined several goodness of seeds and proposed a method to predict these measures using support vector regression (SVR) for their doubly anchored pattern (DAP) system. However, Kozareva and Hovy (2010) does not show how effective the seed set selected by the goodness of seeds that they defined was for the bootstrapping process while they show how accurately they could predict the goodness of seeds.

Early work on bootstrapping includes that of (Hearst, 1992) and that of (Yarowsky, 1995). Abney (2004) extended self-training algorithms including that of (Yarowsky, 1995), forming a theory different from that of (Komachi et al., 2008). We chose to extend the theory of (Komachi et al., 2008) because it can actually explain recent graph-based algorithms including that of (Pantel and Pennacchiotti, 2006). The theory of Komachi et al. (2008) is also newer and simpler than that of (Abney, 2004).

The iterative seeding framework can be regarded as an example of active learning on graph-based semi-supervised learning. Selecting seed sets corresponds to sampling a data point in active learning. In active learning on supervised learning, the active learning survey (Settles, 2012) includes a method called expected model change, after which this paper's expected model rotation (EMR) is named. They share a basic concept: the data point that surprises the classifier the most is selected next. Expected model change mentioned by (Settles, 2012), however, is for supervised setting, not semi-supervised setting, with which this paper deals. It also does not aim to provide intuitive understanding of the dataset. Note that our method is for semi-supervised learning and we also made the calculation of EMR practical.

Another idea relevant to our EMR is an "angle diversity" method for support vector machines (Brinker, 2003). Unlike our method, the angle diversity method interprets each data point as data "lines" in a version space. The weight vector is expressed as a point in a version space. Then, it samples a data "line" whose angle formed with existing data lines is large. Again, our method builds upon different settings in that this method is only for supervised learning, while ours is for semi-supervised learning.

3 Theorization of Bootstrapping

This section introduces a theorization of bootstrapping by (Komachi et al., 2008).

3.1 Simple bootstrapping

Let $\mathcal{D} = \{(y_1, \boldsymbol{x}_1), \ldots, (y_l, \boldsymbol{x}_l), \boldsymbol{x}_{l+1}, \ldots, \boldsymbol{x}_{l+u}\}$ be a dataset. The first l data are labeled, and the following u data are unlabeled. We let $n = l + u$ for simplicity. Each $\boldsymbol{x}_i \in \mathbb{R}^m$ is an m-dimensional input feature vector, and $y_i \in C$ is its corresponding label where C is the set of semantic classes. To handle $|C|$ classes, for $k \in C$, we call an n-sized 0-1 vector $\boldsymbol{y}_k = (y_{1k}, \ldots, y_{nk})^\top$ a "seed vector", where $y_{ik} = 1$ if the i-th instance is labeled and its label is k, otherwise $y_{ik} = 0$.

Note that this multi-class formalization includes typical ranking settings for harvesting tasks as its special case. For example, if the task is to harvest animal names from all given instances, such as "elephant" and "zebra", C is set to be binary as $C = \{\text{animal}, \text{not animal}\}$. The ranking is obtained by the score vector resulting from the seed vector $\boldsymbol{y}_{\text{animal}} - \boldsymbol{y}_{\text{not animal}}$ due to the linearity.

By stacking row vectors $\boldsymbol{x}_i$, we denote $X = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)^\top$. Let X be an instance-pattern (feature) matrix where $(X)_{ij}$ stores the value of the jth feature in the ith datum. Note that we can almost always assume the matrix X to be sparse for bootstrapping purposes due to the language sparsity. This sparsity enables the fast computation.

The simple bootstrapping (Komachi et al., 2008) is a simple model of bootstrapping using matrix representation. The algorithm starts from $\boldsymbol{f}_0 \stackrel{\text{def}}{=} \boldsymbol{y}$ and repeats the following steps until $\boldsymbol{f}_c$ converges.

1. $\boldsymbol{a}_{c+1} = X^\top \boldsymbol{f}_c$. Then, normalize $\boldsymbol{a}_{c+1}$.
2. $\boldsymbol{f}_{c+1} = X\boldsymbol{a}_{c+1}$. Then, normalize $\boldsymbol{f}_{c+1}$.

The score vector after c iterations of the simple bootstrapping is obtained by the following equation.

$$\boldsymbol{f} = \left(\frac{1}{m}\frac{1}{n}XX^\top\right)^c \boldsymbol{y} \tag{1}$$

"Simplified Espresso" is a special version of the simple bootstrapping where $X_{ij} = \frac{\text{pmi}(i,j)}{\max \text{pmi}}$ and we normalize score vectors uniformly: $\boldsymbol{f}_c \leftarrow \boldsymbol{f}_c/n$, $\boldsymbol{a}_c \leftarrow \boldsymbol{a}_c/m$. Here, $\text{pmi}(i,j) \stackrel{\text{def}}{=} \log \frac{p(i,j)}{p(i)p(j)}$.

Komachi et al. (2008) pointed out that, although the scores $\boldsymbol{f}_c$ are normalized during the iterations in the simple bootstrapping, when $c \to \infty$, $\boldsymbol{f}_c$ converges to a score vector that does not depend on the seed vector $\boldsymbol{y}$ as the principal eigenvector of $\left(\frac{1}{m}\frac{1}{n}XX^\top\right)$ becomes dominant. For bootstrapping purposes, however, it is appropriate for the resulting score vector $\boldsymbol{f}_c$ to depend on the seed vector $\boldsymbol{y}$.

3.2 Laplacian label propagation

To make $\boldsymbol{f}$ seed dependent, Komachi et al. (2008) noted that we should use a power series of a matrix rather than a simple power of a matrix. As the following equation incorporates the score vectors $((-L)^c\boldsymbol{y})$ with both low and high c values, it provides a seed dependent score vector with taking higher c into account.

$$\sum_{c=0}^{\infty} \beta^c \left((-L)^c \boldsymbol{y}\right) = (I + \beta L)^{-1} \boldsymbol{y} \tag{2}$$

Instead of using $\left(\frac{1}{m}\frac{1}{n}XX^\top\right)$, Komachi et al. (2008) used $L \stackrel{\text{def}}{=} I - D^{-1/2}XX^\top D^{-1/2}$, a normalized graph Laplacian for graph theoretical reasons. D is a diagonal matrix defined as $D_{ii} \stackrel{\text{def}}{=} \sum_j (XX^\top)_{ij}$. This infinite summation of the matrix can be expressed by inverting the matrix under the condition that $0 < \beta < \frac{1}{\rho(L)}$, where $\rho(L)$ be the spectral radius of L.

Komachi et al. (2008)'s Laplacian label propagation is simply expressed as (3). Given $\boldsymbol{y}$, it outputs the score vector $\boldsymbol{f}$ to rank unlabeled instances. They reports that the resulting score vector $\boldsymbol{f}$ constantly achieves better results than those by Espresso (Pantel and Pennacchiotti, 2006).

$$\boldsymbol{f} = (I + \beta L)^{-1} \boldsymbol{y}. \tag{3}$$

4 Proposal: criteria for iterative seeding

This section describes our iterative seeding framework. The entire framework is shown in Algorithm 1.

Let g_i be the goodness of seed for an unlabeled instance i. We want to select the instance with the highest goodness of seed as the next seed added in the next iteration.

$$\hat{i} = \arg\max_i g_i \tag{4}$$

Algorithm 1 Iterative seeding framework

Require: $\boldsymbol{y}$, X, the set of unlabeled instances U, the set of classes C.
  Initialize $g_{k,i'}; \forall k \in C, \forall i' \in U$
  repeat
    Select instance $\hat{i}$ by (4).
    Label $\hat{i}$. Let k' be $\hat{i}$'s class.
    $U \leftarrow U \backslash \{\hat{i}\}$
    for all $i' \in U$ **do**
      Recalculate $g_{k',i'}$
    end for
  until A sufficient number of seeds are collected.

Each seed selection criterion defines each goodness of seed g_i. To measure the goodness of seeds, we want to measure how an unlabeled instance will affect the model underlying Eq. (3). That is, we want to choose the unlabeled instance that would most influence the model. However, as the model parameters are not explicitly shown in Eq. (3), we first need to reveal them before measuring the influence of the unlabeled instances.

4.1 Scores as margins

This section reveals the model parameters through the dual problem of bootstrapping. We show that the score obtained by Eq. (3) can be regarded as the "margin" between each unlabeled data point and the hyperplane obtained by ridge regression; specifically, we can show that the i-th element of the resulting score vector obtained using Eq. (3) can be written as $f_i = \beta\left(y_i - \langle \hat{\mathbf{w}}, \phi\left(\mathbf{x}_i\right)\rangle\right)$, where $\hat{\mathbf{w}}$ is the optimal model parameter that we need to reveal (Figure 1). ϕ is a feature function mapping $\mathbf{x}_i$ to a feature space and is set to make this relation hold. Note that, for unlabeled instances, $y_i = 0$ holds, and thus f_i is simply $f_i = -\beta \langle \hat{\mathbf{w}}, \phi\left(\mathbf{x}_i\right)\rangle$. Therefore, $|f_i| \propto \| \langle \hat{\mathbf{w}}, \phi\left(\mathbf{x}_i\right)\rangle \|$ denotes the "margin" between each unlabeled data point and the underlying hyperplane.

Let Φ be defined as $\Phi \stackrel{\text{def}}{=} \left(\phi\left(\mathbf{x}_1\right), \ldots, \phi\left(\mathbf{x}_n\right)\right)^\top$. The score vector $\mathbf{f}$ can be written using Φ as in (6). If we set Φ as Eq. (6), Eq. (5) is equivalent to Eq. (3).

$$\mathbf{f} = \left(I + \beta\Phi\Phi^\top\right)^{-1}\mathbf{y} \tag{5}$$

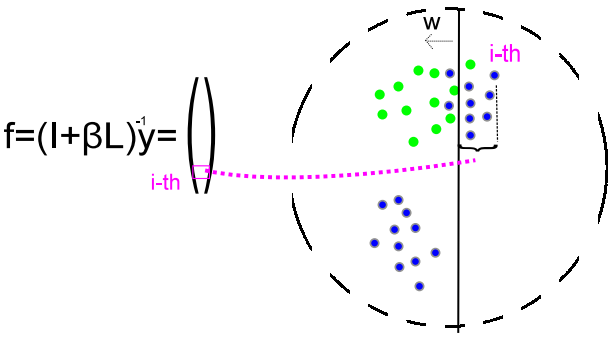

Figure 1: Scores as margins. The absolute values of the scores of the unlabeled instances are shown as the margin between the unlabeled instances and the underlying hyperplane in the feature space.

$$\Phi\Phi^\top = L = I - D^{-\frac{1}{2}} X X^T D^{-\frac{1}{2}} \tag{6}$$

By taking the diagonal of $\Phi\Phi^\top$ in Eq. (6), it is easy to see that $\|\phi\left(\mathbf{x}_i\right)\|^2 = \langle \phi\left(\mathbf{x}_i\right), \phi\left(\mathbf{x}_i\right)\rangle \leq 1$. Thus, the data points mapped into the feature space are within a unit circle in the feature space shown as the dashed circles in Figure 1-3. The weight vector is then represented by the classifying hyperplane that goes through the origin in the feature space. The classifying hyperplane views all the points positioned left of this hyperplane as the green class, and all the points positioned right of this hyperplane as the blue gray-stroked class. Note that all the points shown in Figure 1 are unlabeled, and thus the classifying hyperplane does not know the true classes of the data points. Due to the lack of space, the proof is shown in the appendix.

4.2 Margin criterion

Section 4.1 uncovered the latent weight vector for the bootstrapping model Eq. (3). A weight vector specifies a hyperplane that classifies instances into semantic classes. Thus, weight vector interpretation easily leads to an iterative seeding criterion: an unlabeled instance closer to the classifying hyperplane is more uncertain, and therefore obtains higher goodness of seed. We call this criterion the "margin criterion" (Figure 2).

First, we define $g_{k,i'} \stackrel{\text{def}}{=} |(\mathbf{f}_k)_{i'}|/s_k$ as the goodness of an instance i' to be labeled as k. s_k is the number of seeds labeled as class k in the current seed set. In the margin criterion, the goodness of the seed i' is then obtained by the difference between

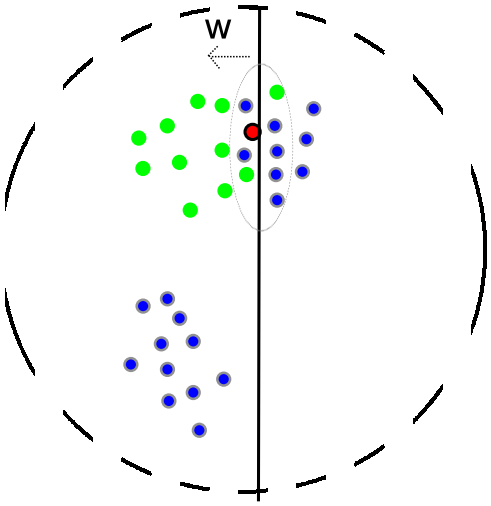

Figure 2: Margin criterion in binary setting. The instance closest to the underlying hyperplane, the red-and-black-stroked point, is selected. The part within the large gray dotted circle is not well separated. Margin criterion continues to select seeds from this part only in this example, and fails to sample from the left-bottom blue gray-stroked points. Note that all the points are unlabeled and thus the true classes of data points cannot be seen by the underlying hyperplane in this figure.

the largest and second largest $g_{k,i'}$ among all classes as follows:

$$g_i^{\text{Margin}} \stackrel{\text{def}}{=} -\left(\max_k g_{k,i'}^{\text{Margin}} - 2^{\text{nd}}\text{largest}_k g_{k,i'}^{\text{Margin}}\right). \tag{7}$$

The shortcoming of Margin criterion is that it can be "stuck", or jammed, or trapped, when the data are not well separated and the underlying hyperplanes goes right through the not well-separated part. In Figure 2, the part within the large gray dotted circle is not well separated. Margin criterion continues to select seeds from this part only in this example, and fails to sample from the left-bottom blue gray-stroked points.

4.3 Expected Model Rotation

To avoid Margin criterion from being stuck in the part where the data are not well separated, we propose another more promising criterion: the "Expected Model Rotation (EMR)". EMR measures the expected rotation of the classifying hyperplane (Figure 3) and selects the data point that rotates the underlying hyperplane "the most" is selected. This selection method prevents EMR from being stuck in the area where the data points are not well separated. Another way of viewing EMR is that it selects the data point that surprises the current classifier the most. This makes the data points influential to the classification selected in early iteration in the iterative seeding framework. A simple rationale of EMR is that important information must be made available earlier.

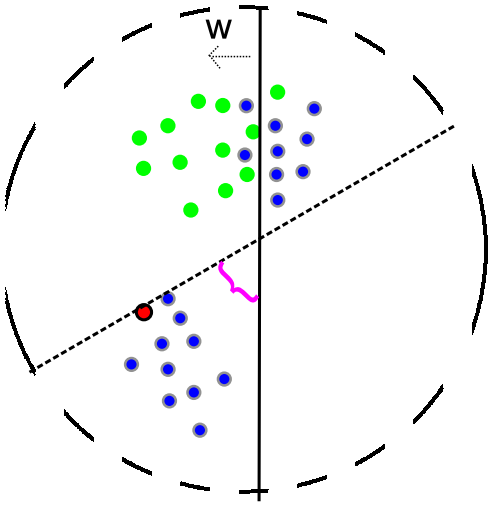

Figure 3: EMR criterion in binary setting. The instance that would rotate the underlying hyperplane the most is selected. The amount denoted by the purple brace "{" is the goodness of seeds in the EMR criterion. This criterion successfully samples from the left bottom blue points.

To obtain the "expected" model rotation, in EMR, we define the goodness of seeds for an instance i', $g_{i'}$ as the sum of each per-class goodness of seeds $g_{k,i'}$ weighted by the probability that i' is labeled as k. Intuitively, $g_{k,i'}$ measures how the classifying hyperplane would rotate if the instance i' were labeled as k. Then, $g_{k,i'}$ is weighted by the probability that i' is labeled as k and summed. The probability for i' to be labeled as k can be obtained from the i'-th element of the current normalized score vector $p_{i'}(k) \stackrel{\text{def}}{=} \frac{\left|(\mathbf{f}_k)_{i'}/s_k\right|}{\sum_{k\in C}\left|(\mathbf{f}_k)_{i'}/s_k\right|}$, where s_k is the number of seeds labeled as class k in the current seed set.

$$g_{i'}^{\text{EMR}} \stackrel{\text{def}}{=} \sum_{k\in C} p_{i'}(k)\, g_{k,i'}^{\text{EMR}} \tag{8}$$

The per-class goodness of seeds $g_{k,i'}$ can be calculated as follows:

$$g_{k,i'}^{\text{EMR}} \stackrel{\text{def}}{=} 1 - \left|\frac{\mathbf{w}_k^\top}{||\mathbf{w}_k||}\frac{\mathbf{w}_{k,+i'}}{||\mathbf{w}_{k,+i'}||}\right|. \tag{9}$$

From Eq. (17) in the proof, $\mathbf{w} = \Phi^\top \mathbf{f}$. Here, $\mathbf{e}_{i'}$ is a unit vector whose i'-th element is 1 and all other elements are 0.

$$\mathbf{w}_k = \Phi^\top \mathbf{f}_k = \Phi^\top (I + \beta L)^{-1} \mathbf{y}_k \quad (10)$$

$$\mathbf{w}_{k,+i'} = \Phi^\top \mathbf{f}_{k,+i'} = \Phi^\top (I + \beta L)^{-1} (\mathbf{y}_k + \mathbf{e}_{i'}) \quad (11)$$

Although Eqs. (10) and (11) use Φ, we do not need to directly calculate Φ. Instead, we can use Eq. (6) to calculate these weight vectors as follows:

$$\mathbf{w}_k^\top \mathbf{w}_{k,+i'} = \mathbf{f}_k^\top \left(I - D^{-\frac{1}{2}} X X^T D^{-\frac{1}{2}}\right) \mathbf{f}_{k,+i'} \quad (12)$$

$$||\mathbf{w}|| = \sqrt{\mathbf{f}^\top \left(I - D^{-\frac{1}{2}} X X^T D^{-\frac{1}{2}}\right) \mathbf{f}}. \quad (13)$$

For more efficient computation, we cached $(I + \beta L)\, \mathbf{e}_{i'}$ to boost the calculation in Eqs. (10) and (11) by exploiting the fact that $\mathbf{y}_k$ can be written as the sum of $\mathbf{e}_i$ for all the instances in class k.

5 Evaluation

We evaluated our method for two bootstrapping tasks with high labeling costs. Due to the nature of bootstrapping, previous papers have commonly evaluated each method by using running search engines. While this is useful and practical, it also reduces the reproducibility of the evaluation. We instead used openly available resources for our evaluation.

First, we want to focus on the separatedness of the dataset. To this end, we prepared two datasets: one is "Freebase 1", a not well-separated dataset, and another is "sb-8-1", a well-separated dataset. We fixed $\beta = 0.01$ as Zhou et al. (2011) reports that $\beta = 0.01$ generally provides good performance on various datasets and the performance is not keen to β except extreme settings such as 0 or 1. In all experiments, each class initially has 1 seed and the seeds are selected and increased iteratively according to each criterion. The meaning of each curve is shared by all experiments and is explained in the caption of Figure 4.

"Freebase 1" is an experiment for information extraction, a common application target of bootstrapping methods. Based on (Talukdar and Pereira, 2010), the experiment setting is basically the same as that of the experiment Section 3.1 in their paper[1]. As 39 instances have multiple correct labels, however, we removed these instances from the experiment to perform the experiment under multi-class setting. Eventually, we had $31,143$ instances with $1,529$ features in 23 classes. The task of "Freebase 1" is bootstrapping instances of a certain semantic class. For example, to harvest the names of stars, given {Vega, Altair} as a seed set, the bootstrapping ranks Sirius high among other instances (proper nouns) in the dataset. Following the experiment setting of (Talukdar and Pereira, 2010), we used mean reciprocal rank (MRR) throughout our evaluation [2].

"sb-8-1" is manually designed to be well-separated and taken from 20 Newsgroup subsets[3]. It has $4,000$ instances with $16,282$ features in 8 classes.

Figure 4 and Figure 5 shows the results. We can easily see that "EMR" wins in "Freebase 1", a not well-separated dataset, and "Margin" wins in "sb-8-1", a well-separated dataset. This result can be regarded as showing that "EMR" successfully avoids being "stuck" in the area where the data are not well separated. In fact, in Figure 4, "Random" wins "Margin". This implies that the not well-separated part of this dataset causes the classifying hyperplane in "Margin" criterion to be stuck and make it lose against even simple "Random" criterion.

In contrast, in the "sb-8-1", a well-separated balanced dataset, "Margin" beats the other remaining two. This implies the following: When the dataset is well separated, uncertainty of a data point is the next important factor to select a seed set. As "Margin" exactly takes the data point that is the most uncertain to the current hyperplane, "Margin" works quite well in this example.

Note that all figures in all the experiments show the average of 30 random trials and win-and-lose relationships mentioned are statistically tested using Mann-Whitney test.

While "sb-8-1" is a balanced dataset, realistic data like "freebase 1" is not only not-well-separated, but also imbalanced . Therefore, we performed experiments "sb-8-1", an imbalanced well-separated dataset, and "ol-8-1", an imbalanced not-well sepa-

[1] *Freebase-1 with Pantel Classes*, `http://www.talukdar.net/datasets/class_inst/`

[2] MRR is defined as $MRR \stackrel{\text{def}}{=} \frac{1}{|Q|} \sum_{i \in Q} \frac{1}{r_i}$, where Q is the test set, $i \in Q$ denotes an instance in the test set Q, and r_i is the rank of the correct class among all $|C|$ classes.

[3] `http://mlg.ucd.ie/datasets/20ng.html`

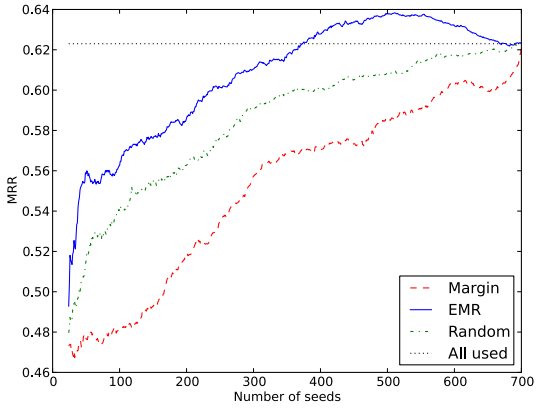

Figure 4: Freebase 1, a NOT well-separated dataset. Average of 30 random trials. "Random" and "Margin" are baselines. "Random" is the case that the seeds are selected randomly. "Margin" is the case that the seeds are selected using the margin criterion described in §4.2. "EMR" is proposed and is the case that the seeds are selected using the EMR criterion described in §4.3. At the rightmost point, all the curves meet because all the instances in the seed pool were labeled and used as seeds by this point. The MRR achieved by this point is shown as the line "All used". If a curve of each method crosses "All used", this can be intepretted as that iterative seeding of the curve's criterion can reduce the cost of labeling all the instances to the crossing point of the x-axis. "EMR" significantly beats "Random" and "Margin" where x-axis is 46 and 460 with p-value < 0.01.

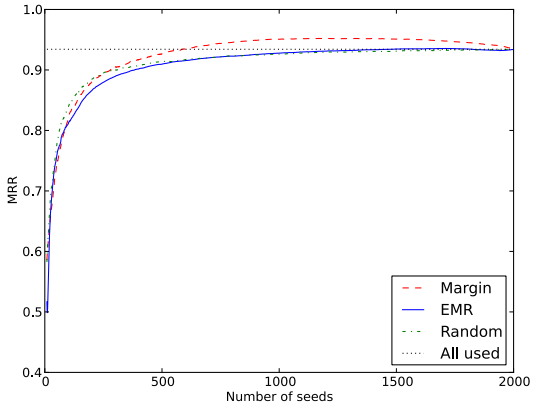

Figure 5: sb-8-1. A dataset manually designed to be well separated. Average of 30 random trials. Legends are the same as those in Figure 4. "Margin" beats "Random" and "EMR" where x-axis is 500 with p-value < 0.01.

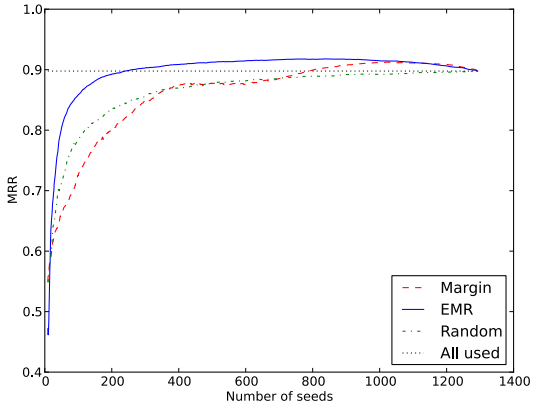

Figure 6: sl-8-1. An imbalanced well separated dataset. Average of 30 random trials. Legends are the same as those in Figure 4. "EMR" significantly beats "Random" and "Margin" where x-axis is 100 with p-value < 0.01.

rated dataset under the same experiment setting used for "sb-8-1". "sl-8-1" have $2,586$ instances with $10,764$ features. "ol-8-1" have $2,388$ instances with $9,971$ features. Both "sl-8-1" and "ol-8-1" have 8 classes.

Results are shown in Figure 6 and Figure 7. In Figure 6, "EMR" beats the other remaining two even though this is a well-separated data set. This implies that "EMR" can also be robust to the imbalancedness as well. In Figure 7, although the MRR of "Margin" eventually is the highest, the MRR of "EMR" rises far earlier than that of "Margin". This result can be explained as follows: "Margin" gets "stuck" in early iterations as this dataset is not well separated though "Margin" achieves best once it gets out of being stuck. In contrast, as "EMR" can avoid being stuck, it rises early achieving high performance with small number of seeds, or labeling. This result suggests that "EMR" is pereferable for reducing labeling cost while "Margin" can sometimes be preferable for higher performance.

6 Conclusion

Little is known about how best to select seed sets in bootstrapping. We thus introduced the iterative seeding framework, which provides criteria for selecting seeds. To introduce the iterative seeding framework, we deepened the understanding of the seeding process in bootstrapping through the dual problem by further extending the interpretation of bootstrapping as graph-based semi-supervised learning (Komachi et al., 2008), which generalizes

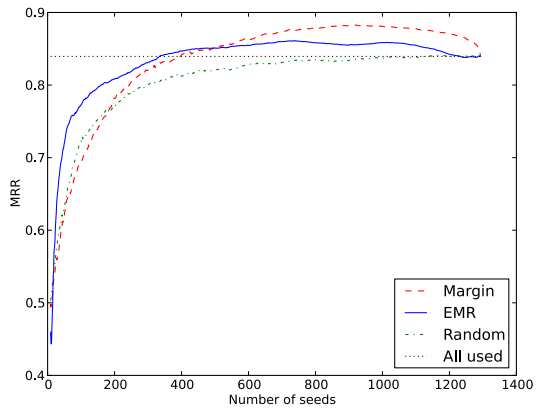

Figure 7: ol-8-1. An imbalanced NOT well separated dataset. Average of 30 random trials. Legends are the same as those in Figure 4. "EMR" significantly beats "Random" and "Margin" where x-axis is 100 with p-value < 0.01. "Margin" significantly beats "EMR" and "Random" where x-axis is $1,000$ with p-value < 0.01.

and improves Espresso-like algorithms.

Our method shows that existing simple "Margin" criterion can be "stuck" at the area when the data points are not well separated. Note that many realistic data are not well separated. To deal with this problem, we proposed "EMR" criterion that is not stuck in the area where the data points are not well separated.

We also contributed to make the calculation of "EMR" practical. In particular, we reduced the number of matrix inversions for calculating the goodness of seeds for "EMR". We also showed that the parameters for bootstrapping also affect the convergence speed of each matrix inversion and that the typical parameters used in other work are fairly efficient and practical.

Through experiments, we showed that the proposed "EMR" significantly beats "Margin" and "Random" baselines where the dataset are not well separated. We also showed that the iterative seeding framework with the proposed measures for the goodness of seeds can reduce labeling cost.

Appendix: Proof Consider a simple ridge regression of the following form where $0 < \beta < 1$ is a positive constant.

$$\min_{\mathbf{w}} \frac{\beta}{2} \sum_{i=1}^{n} \|y_i - \langle \mathbf{w}, \phi(\mathbf{x}_i) \rangle\|^2 + \|\mathbf{w}\|^2 . \quad (14)$$

We define $\xi_i = y_i - \langle \mathbf{w}, \phi(\mathbf{x}_i) \rangle$. By using ξ_i, we can rewrite Eq. (14) into an optimization problem with equality constraints as follows:

$$\min_{\mathbf{w}} \frac{\beta}{2} \sum_{i=1}^{n} \xi_i^2 + \|\mathbf{w}\|^2 \quad (15)$$

$$\text{s.t.} \forall i \in \{1, \ldots, n\}; y_i = \mathbf{w}^\top \phi(\mathbf{x}_i) + \xi_i . \quad (16)$$

Because of the equality constraints of Eq. (16), we obtain the following Lagrange function h. Here, each bootstrapping score f_i occurs as Lagrange multipliers: $h(\mathbf{w}, \xi, \mathbf{f}) \stackrel{\text{def}}{=} \frac{1}{2}\|\mathbf{w}\|^2 + \frac{\beta}{2}\sum_{i=1}^{n} \xi_i^2 - \sum_{i=1}^{n} (\langle \mathbf{w}, \phi(\mathbf{x}_i) \rangle + \xi_i - y_i) f_i$.

By taking derivatives of h, we can derive $\hat{\mathbf{w}}$ by expressing it with the sum of each f_i and $\phi(\mathbf{x}_i)$.

$$\frac{\partial h}{\partial \mathbf{w}} = 0 \Rightarrow \hat{\mathbf{w}} = \sum_{i=1}^{n} f_i \phi(\mathbf{x}_i) \quad (17)$$

$$\frac{\partial h}{\partial \xi_i} = 0 \Rightarrow f_i = \beta(\xi_i = \beta y_i - \langle \hat{\mathbf{w}}, \phi(\mathbf{x}_i) \rangle) \quad (18)$$

Substituting the relations derived in Eqs. (17) and (18) to the equation $\frac{\partial h}{\partial f_i} = 0$ results in Eq. (19).

$$\frac{\partial h}{\partial f_i} = 0 \Rightarrow \sum_{j=1}^{n} f_j \phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j) + \frac{1}{\beta} f_i = y_i \quad (19)$$

Equation (19) can be written as a matrix equation using Φ defined as $\Phi \stackrel{\text{def}}{=} (\phi(\mathbf{x}_1), \ldots, \phi(\mathbf{x}_n))^\top$. From Eq. (20), we can easily derive the form of Eq. (3) as $\left(\Phi\Phi^\top + \frac{1}{\beta} I\right)^{-1} \mathbf{y} \propto \left(I + \beta\Phi\Phi^\top\right)^{-1} \mathbf{y}$.

$$\left(\Phi\Phi^\top + \frac{1}{\beta} I\right) \mathbf{f} = \mathbf{y} \quad (20)$$

□

References

Steven Abney. 2004. Understanding the yarowsky algorithm. *Computational Linguistics*, 30(3):365–395.

Klaus Brinker. 2003. Incorporating diversity in active learning with support vector machines. In *Proc. of ICML*, pages 59–66, Washington D.C.

Zelling S. Harris. 1954. Distributional structure. *Word*.

Marti A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proc. of COLING*, pages 539–545.

Mamoru Komachi, Taku Kudo, Masashi Shimbo, and Yuji Matsumoto. 2008. Graph-based analysis of semantic drift in Espresso-like bootstrapping algorithms. In *Proc. of EMNLP*, pages 1011–1020, Honolulu, Hawaii.

Zornitsa Kozareva and Eduard Hovy. 2010. Not all seeds are equal: Measuring the quality of text mining seeds. In *Proc. of NAACL-HLT*, pages 618–626, Los Angeles, California.

Zornitsa Kozareva, Konstantin Voevodski, and Shanghua Teng. 2011. Class label enhancement via related instances. In *Proc. of EMNLP*, pages 118–128, Edinburgh, Scotland, UK.

Patrick Pantel and Marco Pennacchiotti. 2006. Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In *Proc. of ACL-COLING*, pages 113–120, Sydney, Australia.

Burr Settles. 2012. *Active Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers.

Partha Pratim Talukdar and Fernando Pereira. 2010. Experiments in graph-based semi-supervised learning methods for class-instance acquisition. In *Proc. of ACL*, pages 1473–1481, Uppsala, Sweden.

David Yarowsky. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *Proc. of ACL*, pages 189–196, Cambridge, Massachusetts.

Xueyuan Zhou, Mikhail Belkin, and Nathan Srebro. 2011. An iterated graph laplacian approach for ranking on manifolds. In *Proc. of KDD*, pages 877–885.