Dragging: Density-Ratio Bagging

Yimin Tan
Department of Computer Sciences
University of Wisconsin-Madison

Xiaojin Zhu
Department of Computer Sciences
University of Wisconsin-Madison

June 5, 2013

Abstract

We propose density-ratio bagging (dragging), a semi-supervised extension of bootstrap aggregation (bagging) method. Additional unlabeled training data are used to calculate the weight on each labeled training point by a density-ratio estimator. The weight is then used to construct a weighted labeled empirical distribution, from which bags of bootstrap samples are drawn. Asymptotically, dragging is proved to be no worse than bagging and requires no semi-supervised learning assumptions other than *iid*-ness. We show that compared to bagging, the dragging predictor achieves less asymptotic variance, which leads to a smaller MSE. We conduct real experiments on several regression and classification tasks. The performance of dragging, bagging, semi-supervised learning with density-ratio estimator, and basic supervised learning is compared and discussed.

1 Introduction

In this paper, we propose density-ratio bagging(dragging), a semi-supervised extension of bootstrap aggregation(bagging) method. As a semi-supervised learning (SSL) method, dragging takes advantage of unlabeled training data.

There are many existing SSL algorithms including mixture models, S3VMs, manifold learning, co-training, etc. [1, 2]. For all those SSL algorithms, when their model assumption doesn't hold, SSL may perform even worse than supervised learning algorithms which simply ignore the unlabeled data. Recently, a safe SSL algorithm called "SSL with density-ratio estimator" (DR-SSL) is proposed by Kawakita et al. [3]. As a safe SSL method, it is proved to be no worse than supervised learning regardless of model assumptions. It is safe in the sense that the parameter estimator used in DR-SSL always achieves the same or smaller asymptotic variance.

Dragging applies the idea of DR-SSL to the bagging scenario. Bagging requires resampling from the empirical distribution of the training data to create multiple bags. Similarly, dragging requires sampling from the *weighted* empirical distribution. The weight is learned by a density-ratio estimator based on both labeled and unlabeled training data.

Bagging and DR-SSL are quite different ideas. As an ensemble method, Bagging bootstraps multiple bags and makes prediction by taking average of predictor for each bag. DR-SSL utilizes the unlabeled data to improve the supervised l earner. However, their success can both be justified as the result of reducing the asymptotic variance of the relevant predictor. Usually when the variance term dominates squared bias term, less asymptotic variance leads to better Mean Square Error (MSE).

Dragging combines the advantages of bagging and DR-SSL together. We prove that dragging also achieves less asymptotic variance than bagging. In other words, it is proved to be a safe extension of bagging as well. Compared to DR-SSL, dragging is more flexible in choosing its learning algorithm. While DR-SSL requests that the learning algorithm accept the weight for each training data, dragging resamples training points and give them to the learner. Therefore, one can easily take dragging as a wrapper and use existing learning algorithms as a black box.

The paper is organized as follows: In section 2, we introduce DR-SSL and give its asymptotic analysis result and explain its intuition. In section 3, we review why bagging works in theory via the asymptotic analysis in [4]. We will formally define dragging and analyze its asymptotic behavior in section 4 and present its advantages over bagging. Experiment result including regression and classification tasks on both synthetic and real dataset are listed in section 5. In section 6, we summarize the property of dragging and present future directions.

2 Review on SSL with Density-Ratio Estimator (DR-SSL)

2.1 Density Ratio Estimator

Let's first introduce the density-ratio estimator in [3]. Suppose there are two unknown probability distributions $p(x)$, $q(x)$. One observe iid data sampled from them separately: $x_i \overset{iid}{\sim} p(x)$, $x'_i \overset{iid}{\sim} q(x)$. Density ratio estimator directly estimates the density-ratio $w(x) = q(x)/p(x)$ from those iid data. Assuming a parametric model for density-ratio:

$$w(x;\theta) = \exp(\theta_1\phi_1(x) + \ldots + \theta_r\phi_r(x)) \tag{1}$$

where $\phi_i(x) : X \mapsto \mathbb{R}, i = 1 \ldots r$ are arbitrary functions of x, with the exception that $\phi_1(x) = 1$, and $\theta_1, \ldots, \theta_r$ are parameters to be estimated.

For SSL, $\{x_1, \ldots, x_n\}$ is the labeled data (removing the labels y) while $\{x'_1, \ldots, x'_N\}$ is the unlabeled data. Usually $n \ll N$. Note both samples come from the same marginal distribution $p(x)$ and hence the true density-ratio $w(x) = 1$ everywhere. Nonetheless, one proceeds to *estimate* the density-ratio $w(x)$ and uses the estimate in building a classifier – paradoxically, this leads to better properties. Density ratio is estimated by matching the empirical mean of certain sensing function $\eta(x;\theta) \in \mathbb{R}^r$ on the labeled and unlabeled data, i.e. solving the following equation:

$$\frac{1}{n}\sum_{i=1}^{n}\eta(x_i;\theta)w(x_i;\theta) - \frac{1}{N}\sum_{j=1}^{N}\eta(x'_j;\theta) = 0 \tag{2}$$

The optimal choice of $\eta(x;\theta)$ is given by [5]

$$\eta(x;\theta) = \frac{1}{1 + w(x;\theta) * N/n}\nabla \log w(x;\theta) = \frac{1}{1 + w(x;\theta) * N/n}\phi(x) \tag{3}$$

Let $\hat{\theta}$ be a solution of equation (2), $w(x;\hat{\theta})$ is an estimator of $w(x)$.

2.2 SSL with Density-Ratio Estimator (DR-SSL)

Density-ratio estimator is usually used in the covariate-shift situation where underlying marginal distributions over X in training and test data are different. Interestingly, Sokolovska et al. [6] showed that density-ratio can also be used in semi-supervised learning, where labeled data and unlabeled data share the same marginal distribution.

Here is the problem setting for DR-SSL. Let $X \subset \mathbb{R}^d$ and Y be a finite label space. Let p_{XY} be the unknown underlying data generation distribution, and p_X its marginal. Let $(x_1, y_1), \ldots, (x_n, y_n) \overset{iid}{\sim} p_{XY}$ be a labeled training set, and $x'_1, \ldots, x'_N \overset{iid}{\sim} p_X$ be an unlabeled data set. Typically, $N \gg n$. The goal is to estimate $p(y|x)$ which can be further used for regression or classification. Consider the MLE of $p(y|x)$ under the model $p(y|x, \beta)$

$$\frac{1}{n}\sum_{i=1}^{n}u(x_i, y_i;\beta) = 0 \tag{4}$$

where $u(x_i, y_i; \beta)$ is the score function. Let $\hat{\beta}$ be a solution to the above equation. Asymptotic analysis of Z-estimator suggests that:

$$\begin{aligned} &\sqrt{n}(\hat{\beta} - \beta^*) \xrightarrow{d} \mathbf{N}(\mathbf{0}, V_1) \\ &V_1 = \mathbf{E}[\nabla u]^{-1}\mathbf{E}[uu^{\mathrm{T}}](\mathbf{E}[\nabla u]^{-1})^{\mathrm{T}} \end{aligned} \tag{5}$$

Here, β^* is the solution of the above equation in the asymptotic sense (where finite sum is replaced by integral over the true p_{XY}). β^* is called the best asymptotics, where $p(y|x, \beta^*)$ is closest to the true $p(y|x)$ in KL-divergence sense. Note that in the well-specified case where $p(y|x, \beta^*) = p(y|x)$, V_1 becomes the inverse of Fisher information matrix because $I(\theta) = -\mathbf{E}[\nabla u] = E[uu^{\mathrm{T}}]$. The whole asymptotic analysis degenerates to the common asymptotic result of MLE estimator.

With the help of the additional unlabeled data, DR-SSL gives a better estimation $\tilde{\beta}$, which is the solution to the following equation:

$$\begin{cases} \dfrac{1}{n}\displaystyle\sum_{i=1}^{n} w(x_i; \theta)u(x_i, y_i; \beta) = 0 \\ \dfrac{1}{n}\displaystyle\sum_{i=1}^{n} \eta(x_i; \theta)w(x_i; \theta) - \dfrac{1}{N}\displaystyle\sum_{j=1}^{N} \eta(x'_j; \theta) = \mathbf{0} \end{cases} \tag{6}$$

Compared to MLE, the DR-SSL estimator $\tilde{\beta}$ has the same asymptotic mean β^*, but less asymptotic variance:

$$\begin{aligned} &\sqrt{n}(\tilde{\beta} - \beta^*) \xrightarrow{d} \mathbf{N}(\mathbf{0}, V_2) \\ &V_2 = \mathbf{E}[\nabla u]^{-1}(\mathbf{E}[uu^{\mathrm{T}}] - \mathbf{E}[\bar{u}\bar{u}^{\mathrm{T}}])(\mathbf{E}[\nabla u]^{-1})^{\mathrm{T}} \\ &\bar{u} = \mathbf{E}_{y|x}u(x, y; \beta) \end{aligned} \tag{7}$$

It is easy to verify that $V_2 \leq V_1$, thus $avar(\tilde{\beta}) \leq avar(\hat{\beta})$. The equality holds if the model is well-specified, namely $p(y|x) = p(y|x, \beta^*)$. The reader is referred to [3] for detailed asymptotic analysis.

We can regard DR-SSL as weighted MLE, where the weight is calculated by density-ratio estimator beforehand. The intuition is as follows: First denote the underlying marginal distribution of label data as $p(x)$, the underlying marginal distribution of unlabeled data as $q(x)$. Note for common SSL settings $p(x) = q(x)$, here we distinguish them only for notation convenience. MLE only uses labeled data, while DR-SSL uses unlabeled data to improve it. However, DR-SSL can't use unlabeled data in MLE directly since the MLE is to estimate $p(y|x)$, but we don't have the labels y for unlabeled data. We can instead estimate $p(x)$ and $q(x)$ separately by kernel density estimation,denoted as $\hat{p}(x)$,$\hat{q}(x)$. Since unlabeled data is plentiful, $\hat{q}(x)$ should be quite close to $q(x)$. For each labeled point (x_i, y_i), if $\hat{q}(x_i)/\hat{p}(x_i) > 1$, it suggests that (x_i, y_i) is under-represented, so we boost its weight up by $\hat{q}(x_i)/\hat{p}(x_i)$. Similarly, if $\hat{q}(x_i)/\hat{p}(x_i) < 1$, we decrease its weight by $\hat{q}(x_i)/\hat{p}(x_i)$. Density-ratio SSL has the similar intuition mentioned above. However, instead of calculating density estimators $\hat{p}(x)$ and $\hat{q}(x)$ separately, density-ratio estimator estimates their ratio $w(x_i, \theta)$ directly, and then plugs it in MLE as the weight.

3 Review on Bootstrap Aggregation (Bagging)

3.1 Definition of Bagging

Here is the theoretical definition of Bootstrap aggregation (bagging). Again consider a labeled training set $(x_1, y_1), \ldots, (x_n, y_n) \overset{iid}{\sim} p_{XY}$. Bagging constructs bootstrap examples by sampling with replacement from the empirical distribution of the labeled training set, where the bootstrap examples has the same size as the original labeled training set (i.e., n). After that, bootstrap examples are used to learn predictors. The bagged predictor is defined as the expectation of the bootstrap predictors. In practice, Monte Carlo method is used to compute the expectation. First M different bags of bootstrap examples are sampled. The bagged predictor is the average of the M bootstrap predictors which are trained separately based on each bag.

We use the predictor trained with all training data together (called the base learner) as the baseline comparison for bagging. Empirically, bagging has been well acknowledged to achieve better performance than the base learner, especially when the base learner is unstable (w.r.t. random training data). Meanwhile, theoretical investigations such as [4] give theoretical explanations on how bagging reduces the asymptotic variance and mean squared error for some nonsmooth and unstable predictors.

3.2 Asymptotic Analysis of Bagging

Here is a brief summary of the analysis for variable selection via testing in linear model in [4]. Consider a 1-dimension linear regression model $Y_i = \beta X_i + \epsilon_i$ with $E[X_i^2] = 1, \{\epsilon_i\}$ iid and independent from $\{X_i\}$, $E[\epsilon_i] = 0$, $Var(\epsilon_i) = \sigma^2$. The predictor is given as $\hat{\theta}_n(x) = \hat{\beta}\mathbb{1}_{[|\hat{\beta}|>u_n]}x$ where u_n is suggested by t-test at significance level β. Further assume the threshold $u_n = u_n(c) = c\sigma n^{-1/2}$, true parameter $\beta = \beta_n(b) = b\sigma n^{-1/2}$. In other words, we first compute the MLE $\hat{\beta}$, then plug it in to get the predictor $\hat{\theta}_n(x)$.

The bagged predictor is defined as $\hat{\theta}_{n;B} = E_{\hat{\beta}_b}\left[\hat{\beta}_b\mathbb{1}_{[|\hat{\beta}_b|>u_n]}x\right]$ where $\hat{\beta}_b$ is the MLE of parameter β for each bag. Proposition 2.2 in [4] gives that the asymptotic distribution of the base predictor $\hat{\theta}_n(x)$ is:

$$n^{1/2}\sigma^{-1}\hat{\theta}_n(x) \overset{d}{\to} g(Z_b) = (Z_b - Z_b\mathbb{1}_{[|Z_b|\le c]})x \tag{8}$$

The asymptotic distribution of the bagged predictor $\hat{\theta}_{n;B}(x)$ is:

$$n^{1/2}\sigma^{-1}\hat{\theta}_{n;B}(x) \overset{d}{\to} g_B(Z_b) = (Z_b - \{Z_b\Phi(c - Z_b) - \phi(c - Z_b) - Z_b\Phi(-c - Z_b) + \phi(-c - Z_b)\})x \tag{9}$$

where $Z_b = b + Z$, $Z \sim N(0,1)$, and $\Phi(x), \phi(x)$ are cdf and pdf of $N(0,1)$, respectively.

Simulations were performed to analyze the bias, variance, and MSE of the above two asymptotic distributions. It was observed that for $1 < b < 3$, the MSE reduction is substantial due to the variance reduction of bagging. For most values of b, the bias effect plays negligible role in terms of MSE.

The crucial part of the analysis is based on two asymptotic normality, which will be extensively cited in later section:

$$\sqrt{n}(\hat{\beta} - \beta) \overset{d}{\to} N(0, \sigma^2) \tag{10}$$

$$\sqrt{n}(\hat{\beta}_b - \hat{\beta}) \overset{d}{\to} N(0, \sigma^2) \text{ in probability} \tag{11}$$

Recall that β is the true parameter, $\hat{\beta}$ is the MLE, β_b is the estimator for each bag.

The first asymptotic normality follows the property of MLE, while the second one is usually mentioned as the asymptotic normality of bagged estimator (or simply “bagging works”). The asymptotic normality of bagged estimator are established for different models separately, see [7, 8]. Here is the intuition why bagging works : For unstable predictors, the bagged predictor replaces them with a smoothed predictor and the smoothed predictor leads to smaller variance. In the example above, the hard-threshold indicator function is replaced by a soft-threshold function, with the help of convolution with a Gaussian distribution.

4 Density-Ratio Bootstrap Aggregation (Dragging)

4.1 Definition of Dragging

We are now prepared to define our dragging estimator. While bagging creates bootstrap samples by sampling from the empirical distribution, dragging samples from the weighted empirical distribution. The weight is the same weight used by weighted MLE in density-ratio SSL, which is calculated by the density-ratio estimator [3]. Similar to density-ratio SSL, the weight is used to correct the labeled empirical distribution with the help of the unlabeled data.

The setting for dragging is the same as density-ratio SSL: Let $X \subset \mathbb{R}^d$ and Y be a finite label space. Let p_{XY} be the unknown underlying data generation distribution, and p_X its marginal. Let $(x_1, y_1), \ldots, (x_n, y_n) \overset{iid}{\sim}$

p_{XY} be a labeled training set, and $x'_1, \ldots, x'_N \overset{iid}{\sim} p_X$ be an unlabeled data set. Typically, $N \gg n$. Our dragging procedure is defined in 4 steps:

1. Learn the density ratio estimator $w(x, \theta)$ in the same way as the second line of equation (6).
2. Create M bags of bootstrap samples $B_k = \{B_{ki}(x, y), i = 1 \ldots n\}, k = 1 \ldots M$ from the labeled training set by sampling with replacement. Instead of sampling from the labeled empirical distribution, dragging samples each point $B_{ki}(x, y)$ from the weighted labeled empirical distribution $\hat{P}_{L,w}(x, y)$, where

$$\hat{P}_{L,w}(x, y) = \frac{\sum_{i=1}^{n} \mathbb{1}(x = x_i, y = y_i) w(x_i, \theta)}{\sum_{i=1}^{n} w(x_i, \theta)} \tag{12}$$

3. Learn M predictors based on the M bags separately.
4. The dragging predictor is the average over the M predictors.

Note that the density-ratio estimator is calculated only once before the bags are created in step 1, so all M bags sample from the same weighted empirical distribution.

Dragging and density-ratio SSL share the same key idea. As the first step, Density-ratio SSL and Dragging both calculate the density-ratio $w(x, \theta)$,where unlabeled data are utilized. While density-ratio SSL uses the weight to formulate weighted MLE, dragging uses the weighted labeled empirical distribution as the source probability for bagging sampling. In other words, dragging is a semi-supervised extension of bagging in the same way that density-ratio SSL extends MLE. Since density-ratio SSL has been proven to outperform MLE, we could expect dragging to outperform bagging estimator as well. On the other hand, because bagging is not necessarily always better than the MLE learner, we don't always expect dragging to outperform either density-ratio SSL or MLE.

4.2 Asymptotic Properties of Dragging

There are two relevant asymptotic normality result for dragging. First, the asymptotic normality of density-ratio SSL [3]. Based on equation (7), we have:

$$\sqrt{n}(\tilde{\beta} - \beta^*) \to N(0, V_2). \tag{13}$$

Second, the dragging asymptotic normality ("Dragging works") :

$$\sqrt{n}(\tilde{\beta}_b - \tilde{\beta}) \to N(0, V_1) \tag{14}$$

where $\tilde{\beta}$ is DR-SSL, $\tilde{\beta}_b$ is the dragging estimator for each bag, and β^* is the best asymptotics.

Let's further discuss the second one, "dragging asymptotic normality" (dragging works). Regarding the asymptotic mean, because each bag in dragging is sampled from the weighted empirical labeled distribution, the asymptotic mean should be DR-SSL $\tilde{\beta}$. Regarding the asymptotic variance, interested readers may ask why the asymptotic variance here is V_1 instead of V_2 ($V_2 \leq V_1$). Here is one explanation. The asymptotic variance is introduced by the uncertainty of labeled training data. In DR-SSL, the weight for each labeled point is corrected by the density-ratio estimator,which leads to variance reduction. However, in the dragging procedure, density-ratio is calculated before the bags are created. So variance reduction only happen in equation (10) but not in equation (11).

Actually, If one really wants it to be V_2, the weight of the labeled training data for each bag needs to be corrected separately (which means M density-ratio estimator should be calculated separately). However, we design dragging *not* to reduce asymptotic variance in equation (11) on purpose. The reason is that such variance is crucial for bagging's success. Recall the intuition for why bagging work – we need the expectation over this asymptotic distribution in equation (11) to create the soft-threshold predictor.

4.3 Case Study of Dragging Predictors

Now let's see how these two asymptotic normalities lead to superior performance of dragging over bagging in a particular task, namely variable selection via testing in linear model in Section 3.2. We use the same model for prediction, but now we assume the ground-truth model is no longer linear. In other words, our model assumption is mis-specified. Thus density-ratio SSL is guaranteed to have less asymptotic variance than the MLE (i.e. $V_2 < V_1$).

Consider the 1-dimension linear regression model $Y_i = \beta X_i + \epsilon_i$, with $E[X_i^2] = 1, \{\epsilon_i\}$ iid and independent from $\{X_i\}$, $E[\epsilon_i] = 0$, and $Var(\epsilon_i) = \sigma^2$. Similar to proposition 2.2 in [4], we have the following result for DR-SSL

$$\tilde{\theta}_n(x) = \tilde{\beta}\mathbb{1}_{[|\tilde{\beta}|>u_n]}x$$

and dragging predictor

$$\tilde{\theta}_{n;B} = E_{\tilde{\beta}_b}\left[\tilde{\beta}_b\mathbb{1}_{[|\tilde{\beta}_b|>u_n]}x\right].$$

Next, we will compare the asymptotic distribution of four predictors :

1. MLE or base predictor $\hat{\theta}_n(x)$
2. bagged predictor $\hat{\theta}_{n;B}(x)$
3. DR-SSL $\tilde{\theta}_n(x)$
4. dragging $\tilde{\theta}_{n;B}(x)$.

Recall the asymptotic variance for MLE (base predictor) and DR-SSL, respectively:

$$V_1 = \mathbf{E}[\nabla u]^{-1}\mathbf{E}[uu^{\mathrm{T}}](\mathbf{E}[\nabla u]^{-1})^{\mathrm{T}} \tag{15}$$

$$V_2 = \mathbf{E}[\nabla u]^{-1}(\mathbf{E}[uu^{\mathrm{T}}] - \mathbf{E}[\bar{u}\bar{u}^{\mathrm{T}}])(\mathbf{E}[\nabla u]^{-1})^{\mathrm{T}} \tag{16}$$

Note that for the mis-specified model, $V_1 \neq \sigma^2$. u_n is suggested by t-test at significance level α. Further assume the threshold $u_n = u_n(c) = c\sqrt{V_1}n^{-1/2}$, best fit parameter $\beta^* = \beta_n(b) = b\sqrt{V_1}n^{-1/2}$. The asymptotic distribution of the density-ratio predictor $\tilde{\theta}_n(x)$ is:

$$n^{1/2}(\sqrt{V1})^{-1}\tilde{\theta}_n(x) \xrightarrow{d} g(Z_b) = (Z_b - Z_b\mathbb{1}_{[|Z_b|\leq c]})x \tag{17}$$

The asymptotic distribution of the dragging predictor $\tilde{\theta}_{n;B}(x)$ is:

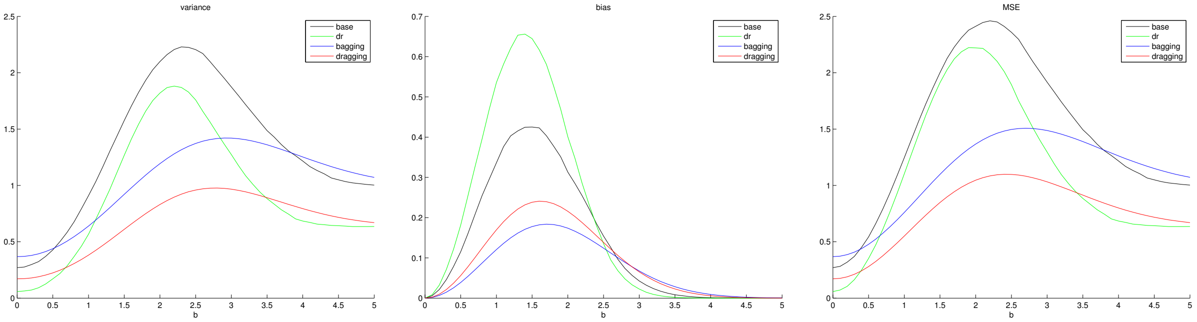
$$n^{1/2}(\sqrt{V1})^{-1}\tilde{\theta}_{n;B}(x) \xrightarrow{d} g_B(Z_b) = (Z_b - \{Z_b\Phi(c - Z_b) - \phi(c - Z_b) - Z_b\Phi(-c - Z_b) + \phi(-c - Z_b)\})x \tag{18}$$

where $Z_b = b + Z$, $Z \sim N(0, \rho), \rho = \frac{V_2}{V_1} < 1$ and $\Phi(x), \phi(x)$ are cdf and pdf of $N(0, 1)$, respectively. Recall that the asymptotic distribution for the base predictor $\hat{\theta}_n$ has almost the same formula as density-ratio SSL predictor $\tilde{\theta}_n$, but with $\rho = 1$. Similarly, the asymptotic distribution for the bagging predictor $\hat{\theta}_{n,B}$ has almost the same formula as dragging predictor $\tilde{\theta}_{n,B}$, but with $\rho = 1$.

Now we run simulation to compare the asymptotic distributions of these four predictors in terms of their bias, variance and MSE. Figure 1 simulates the predictors for $x = 1, c = 1.96, V_1 = 1, \rho = 0.8$. Here are some observations:

1. Dragging outperforms bagging: The dragging predictor always has smaller variance than the bagging predictor. In terms of squared bias, the dragging predictor is slightly worse than bagging. In terms of MSE, since variance dominates the squared bias, dragging outperforms bagging.

2. Dragging outperforms MLE and DR-SSL: Most of time ($1 \leq b \leq 3$), the dragging predictor has the smallest variance among the four predictors. Dragging also outperforms the base and density-ratio SSL predictor in squared bias. For $1 \leq b \leq 3$, dragging has the smallest MSE among the 4 predictors. However, this is probably owing to the fact that "variable selection via testing in linear model" is a bagging-friendly task. In general, we don't expect dragging to always beat MLE and DR-SSL.



variance squared bias MSE

Figure 1: Simulations of asymptotic distributions

5 Experiments

5.1 Datasets and Algorithms

We want to compare the base learner, density-ratio (if possible), bagging, and dragging predictors. Our tasks include both regression and classification. Our datasets include:

1. Bagging friendly synthetic datasets: Friedman 1,2,3
2. Density-ratio friendly synthetic dataset: DR-SSL1 in Section 7.1 of [3]
3. SSL friendly datasets: g241.c ,g241.n, Digit1
4. UCI datasets: Boston housing, breast cancer Wisconsin, diabetes, ionosphere, spambase, abalone, cmc

We use multiple regression and classification algorithms for the base learner, including

1. Classification: SVM, Classification tree, logistic regression, Random forest
2. Regression: Linear regression, Regression tree, Random forest

Note that random forest is used as base learner, we can still have bagging as the wrapper as usual. We adopt Kernel unconstrained least-squares importance fitting (KulSIF) [9] as the standard density-ratio estimator and use it in the density-ratio SSL predictor and the dragging predictor.

5.2 Procedure

For simplicity, we only consider one base learner Alg, and one dataset D. We experimented with different training set sizes for the labeled and unlabeled data. The procedure is as follows.

1. Input: a set of labeled-set sizes $\{L_1, ..., L_{max}\}$, unlabeled-set size U
2. Randomly sample training set D_{train} of size $L_{max} + U$ from the data set. The rest of the data are used as test data D_{test}
3. For each L, U size pair:
   (a) Run M trials. For each trial:
      i. Sample L data from D_{train}, denoted as D_L
      ii. Sample U data from $D_{train} - D_L$, denoted as D_U

iii. With training data D_L and D_U, learn base learner, density-ratio, bagging, and dragging, separately. Inside training, cross validation is used to choose parameters of algorithms.

iv. Evaluate the four learners on D_{test},i.e. calculate Loss(D_{test}, learner). We use 0/1 loss for classification, MSE for regression

(b) Take the average of Loss in M trials

For the following regression and classification tasks, we used $M = 10$ (for classification), 100(for regression); $L = 10$ and 100; $U = 1000$ (if the dataset is not large enough, $U = 300$). We made sure that $|D_{test}| > 200$. We used 5-fold cross-validation.

5.3 Results

Table 1 presents the regression experiment results. It contains 5 datasets and 3 algorithms. For each algorithm-dataset pair, we ran experiments with label data size = 10 and 100. We list the regression mean square error (MSE) for base learner(base), DR-SSL(dr), bagging(bag), and dragging(drag). Each number is the average over 10 trials. We mark the smallest MSE among the four in bold. In addition, we list the p-value of one-side t-test between dragging and bagging. Note that smaller p-value means dragging is better than bagging.

Table 2 presents the classification experiment results. It contains 9 datasets and 4 algorithms. The number listed are test set classification error, the structure is the same as in Table 1.

Here are some observations and discussion:

1. DR-SSL vs base: We don't always see DR-SSL being better than the base learner. But DR-SSL is better in the smaller L situation. This coincides with our intuitive explanation of DR-SSL, where labeled empirical distribution is corrected by unlabeled data. With smaller L, it makes more sense to make such a correction. Recall that the proof of DR-SSL uses asymptotics. So in practice, we may observe that DR-SSL is worse than the base learner.

2. Bagging vs base: Bagging is not always better than the base learner. For certain algorithms (regression tree, classification tree), bagging is much better. This follows from the theory that bagging helps unstable learner.

3. Dragging vs bagging: From the p-value listed, we don't always see dragging being better than bagging. But we note that whether dragging is better than bagging highly depends on whether DR-SSL is better than the base learner. In the regression experiments, among the 12 cases where DR-SSL is better than the base learner, there are 10 cases where dragging is better than bagging. On the other hand, among the 8 cases where DR-SSL is worse than the base learner, there are 3 cases where dragging is worse than bagging. However, in classification experiments, such a dependence is not observed.

4. Dragging vs all other three: It is hard to say that dragging is the best among the four algorithms. Dragging is a combination of density-ratio SSL and bagging ideas. Although these two ideas have some theoretical guarantee in certain situation, but in practice we don't always see either of them work. Therefore, we don't see dragging always works either.

6 Conclusion

This paper proposes density-ratio aggregation (dragging), a semi-supervised ensemble method. Dragging improves bagging by sampling from the weighted empirical distribution, where the weight is calculated by density-ratio estimator with the help of unlabeled data. Asymptotic normality of dragging is discussed and used to prove the asymptotic behavior of a particular learning task. We see how dragging reduces the asymptotic variance of bagging, and thus leads to smaller MSE. Real experiments on a wide range of classification and regression tasks indicate the success of dragging compared to bagging to a limited degree.

		Regression Tree					Linear regression					Random Forest			
Dataset	l	base	dr	bag	drag	p-value	base	dr	bag	drag	p-value	base	bag	drag	p-value
Friedman1	10	31.240	31.611	21.700	**21.582**	0.140	21.974	21.979	19.363	**19.361**	0.486	**22.761**	23.359	23.308	0.099
	100	14.070	13.990	**9.169**	9.221	0.883	7.342	**7.310**	7.369	**7.335**	0.001	**12.660**	13.385	13.408	0.967
Friedman2	10	2.997e5	2.986e5	2.245e5	**2.182e5**	0.002	2.192e5	**2.173e5**	2.394e5	2.356e5	0.219	2.197e5	2.214e5	**2.194e5**	0.125
	100	1.551e5	1.566e5	**1.271e5**	1.275e5	0.822	1.246e5	**1.245e5**	1.248e5	1.245e5	0.043	1.382e5	1.382e5	**1.382e5**	0.402
Friedman3	10	1.009	1.012	0.828	**0.819**	0.076	0.839	**0.829**	0.836	0.848	0.886	0.758	**0.750**	0.755	0.872
	100	0.686	0.685	0.665	**0.665**	0.214	**0.644**	0.644	0.646	0.645	0.225	0.649	0.646	**0.646**	0.277
Boston housing	10	76.617	82.301	58.792	**58.421**	0.303	92.517	92.923	**84.667**	98.310	0.953	**57.908**	60.115	61.268	0.941
	100	29.195	29.399	20.683	**20.712**	0.584	26.376	25.978	26.288	**25.940**	0.000	**27.631**	29.198	29.256	0.888
DR-SSL1	10	1.946	1.912	1.325	**1.229**	0.000	0.359	0.352	0.315	**0.246**	0.000	**1.039**	1.125	1.072	0.000
	100	0.505	0.494	**0.352**	0.355	0.842	0.217	0.211	0.211	**0.151**	0.000	**0.320**	0.334	0.336	0.965

Table 1: Benchmark regression comparison results. All numbers are averages over 10 trials.

		SVM					Classification Tree					Logistic regression					Random Forest			
Dataset	l	base	dr	bag	drag	p-value	base	dr	bag	drag	p-value	base	dr	bag	drag	p-value	base	bag	drag	p-value
breast cancer	10	0.049	0.050	**0.047**	0.048	0.861	**0.111**	0.111	0.135	0.114	0.172	0.196	**0.192**	0.208	0.222	0.716	0.048	**0.047**	0.050	0.946
	100	0.043	0.046	**0.042**	0.044	0.959	0.080	0.077	**0.055**	0.055	0.429	**0.126**	0.143	0.130	0.151	0.996	0.044	**0.041**	0.044	0.974
diabetes	10	0.333	0.336	0.331	**0.327**	0.068	0.398	0.355	0.357	**0.314**	0.139	**0.405**	0.424	0.417	0.423	0.771	0.337	0.331	**0.322**	0.207
	100	0.328	0.268	0.258	**0.254**	0.166	0.294	0.286	0.265	**0.254**	0.065	0.335	0.335	0.335	**0.333**	0.161	0.250	0.248	**0.247**	0.383
ionosphere	10	**0.306**	0.306	0.315	0.314	0.339	0.256	0.256	**0.227**	0.230	0.828	**0.301**	0.301	0.302	0.309	0.992	**0.283**	0.283	0.285	0.714
spambase	10	**0.322**	0.343	0.324	0.353	0.823	0.288	0.287	**0.284**	0.326	0.955	**0.320**	0.320	0.331	0.365	0.972	**0.253**	0.253	0.298	0.941
	100	0.137	0.166	**0.134**	0.155	1.000	0.170	0.192	**0.120**	0.134	0.980	**0.146**	0.161	0.147	0.170	0.998	**0.097**	0.101	0.103	0.854
abalone	10	**0.380**	0.381	0.386	0.382	0.308	0.358	**0.339**	0.347	0.348	0.542	0.380	0.380	0.382	**0.375**	0.280	0.350	0.342	**0.326**	0.168
	100	**0.250**	0.251	0.251	0.253	0.848	0.298	0.294	**0.263**	0.264	0.542	0.255	0.254	**0.252**	0.253	0.698	0.261	**0.259**	0.260	0.842
cmc	10	0.421	0.422	0.420	**0.419**	0.310	**0.399**	0.402	0.406	0.405	0.466	0.437	0.434	0.426	**0.415**	0.064	**0.411**	0.411	0.412	0.536
	100	0.366	0.367	**0.350**	0.355	0.791	0.367	0.369	**0.336**	0.339	0.846	0.416	0.412	0.410	**0.408**	0.324	**0.352**	0.352	0.356	0.940
g241.c	10	0.447	**0.444**	0.485	0.482	0.172	0.486	0.494	**0.469**	0.471	0.697	**0.377**	0.378	0.386	0.386	0.500	**0.473**	0.478	0.477	0.296
	100	0.247	0.246	**0.232**	0.235	0.790	0.427	0.434	**0.344**	0.350	0.768	0.239	0.239	**0.237**	0.237	0.500	0.290	0.288	**0.285**	0.285
g241.n	10	**0.487**	0.487	0.489	0.487	0.199	**0.480**	0.483	0.491	0.487	0.045	**0.454**	0.454	0.456	0.455	0.052	**0.492**	0.498	0.494	0.117
	100	0.288	0.287	0.273	**0.270**	0.161	0.453	0.463	0.392	**0.391**	0.479	0.276	0.276	0.268	**0.267**	0.390	0.338	**0.335**	0.336	0.541
Digit1	10	0.324	0.324	0.326	**0.320**	0.048	**0.452**	0.452	0.457	0.460	0.956	**0.267**	0.267	0.272	0.267	0.142	**0.428**	0.433	0.441	0.901
	100	0.094	0.095	0.086	**0.085**	0.375	0.247	0.251	**0.145**	0.148	0.688	0.090	0.090	0.085	**0.084**	0.348	**0.076**	0.086	0.085	0.468

Table 2: Benchmark classification comparison results. All numbers are averages over 10 trials.

Further work includes (i) giving a theoretical proof of the dragging asymptotic normality assumption, and (ii) a more thorough investigation about when dragging outperforms bagging in practice.

Acknowledgments

This work is support by National Science Foundation grant IIS-0916038. We thank Sijian Wang for helpful discussions.

References

[1] Olivier Chapelle, Alexander Zien, and Bernhard Schölkopf, editors. *Semi-supervised learning*. MIT Press, 2006.

[2] Xiaojin Zhu. Semi-supervised learning. In Claude Sammut and Geoffrey Webb, editors, *Encyclopedia of Machine Learning*. Springer, first edition, 2010.

[3] Masanori Kawakita and Takafumi Kanamori. Semi-supervised learning with density-ratio estimation. *Mach. Learn.*, 91(2):189–209, May 2013.

[4] Peter Bühlmann and Bin Yu. Analyzing bagging. *Annals of Statistics*, 30:927–961, 2002.

[5] Jing Qin. Inferences for case-control and semiparametric two-sample density ratio models. *Biometrika*, 85(3):619–630, 1998.

[6] Nataliya Sokolovska, Olivier Cappé, and François Yvon. The asymptotics of semi-supervised learning in discriminative probabilistic models. In William W. Cohen, Andrew McCallum, and Sam T. Roweis, editors, *ICML*, volume 307 of *ACM International Conference Proceeding Series*, pages 984–991. ACM, 2008.

[7] David A Freedman. Bootstrapping regression models. *The Annals of Statistics*, pages 1218–1228, 1981.

[8] Evarist Giné and Joel Zinn. Bootstrapping general empirical measures. *The Annals of Probability*, pages 851–869, 1990.

[9] Takafumi Kanamori, Taiji Suzuki, and Masashi Sugiyama. Statistical analysis of kernel-based least-squares density-ratio estimation. *Machine Learning*, 86(3):335–367, 2012.