The Value of Agreement, a New Boosting Algorithm[⋆]

Boaz Leskes[⋆⋆]

University of Amsterdam, ILLC, Plantage Muidergracht 24, 1018 TV Amsterdam
bleskes@science.uva.nl

Abstract. We present a new generalization bound where the use of unlabeled examples results in a better ratio between training-set size and the resulting classifier's quality and thus reduce the number of labeled examples necessary for achieving it. This is achieved by demanding from the algorithms generating the classifiers to agree on the unlabeled examples. The extent of this improvement depends on the diversity of the learners—a more diverse group of learners will result in a larger improvement whereas using two copies of a single algorithm gives no advantage at all. As a proof of concept, we apply the algorithm, named AgreementBoost, to a web classification problem where an up to 40% reduction in the number of labeled examples is obtained.

1 Introduction

One of the simplest but popular models in machine learning is the so called *supervised learning* model. This model represents a scenario where a 'learner' is required to solve a *classification problem.* The model assumes the existence of a set of possible examples $\mathcal{X}$ which are divided in some way into a set of classes $\mathcal{Y} \subseteq [-1,+1]$ (often called labels). Furthermore, it is assumed that there exists a distribution P over $\mathcal{X} \times \mathcal{Y}$, which represents the 'chance' to see a specific example and its label in real life. The learning algorithm's task is then to construct a mapping $f : \mathcal{X} \to \mathcal{Y}$, which predicts the distribution P well, i.e., minimizes $P(\{f(x) \neq y : (x,y) \in \mathcal{X} \times \mathcal{Y}\})$. The only information available to the learner to assist it in its task is a finite *training set* $S = \left\{(x_j, y_j)\right\}_{j=1}^{n_s}$, generated by repeatedly and independently sampling the distribution P .

Despite of the high abstraction level, many real life applications fall nicely within this model. Problems like OCR, web pages classification (as done in Internet directories) and detection of spam e-mail are only a few of many problems that fit into this scheme. In all the examples above and in many others, it is relatively hard to obtain a large sample of labeled examples. The sample has to be carefully analyzed and labeled by humans—a costly and time consuming task. However in many situations it is fairly easy to obtain unlabeled examples: examples from the example space $\mathcal{X}$ *without* the class that they belong to. This process can be easily mechanized and preformed by a machine, much faster then any human-plausible rate. This difference between labeled and unlabeled examples has encouraged researchers in the recent years to study the benefits that unlabeled examples may have in various learning scenarios.

[⋆] A full version of this paper is available on http://www.illc.uva.nl/Publications/ResearchReports/MoL-2005-02.text.pdf

[⋆⋆] Eligible for the "Best Student Paper" award.

P. Auer and R. Meir (Eds.): COLT 2005, LNAI 3559, pp. 95–110, 2005.

At first glance it might seem that nothing is to be gained from unlabeled examples. After all, unlabeled examples lack the most important piece of information—the class to which they belong. However, this is not necessarily the case. In some theoretical settings, it is beneficial to gain knowledge over the examples' marginal distribution $P(x)$ (for example in [7]). In these cases, having extra examples, with or without their label, provides this extra information. On the other hand, there exist situations (for example in [9]) where knowing $P(x)$ is not helpful and unlabeled examples do not help at all. The main goal of this sort of research is to determine the amount of information that can be extracted from unlabeled examples. However, unlabeled examples have also been used by algorithms in a more practical way: as a sort of a communication platform between two different learning algorithms. One such usage is the so called Co-Training model or strategy.

A typical example of Co-Training can be found in [5], a paper often cited with respect to unlabeled examples. In their paper, Blum and Mitchell provide both an algorithm and a theoretical framework where unlabeled examples are used to communicate an 'opinion' about an unlabeled example from one algorithm to another. As a case study, the algorithm is then applied to a web-page classification problem involving identifying courses' homepages out of a collection of web-pages.

In this Co-Training model, it is assumed that the example space can be split into two 'views' $\mathcal{X}^1$ and $\mathcal{X}^2$ (i.e., $\mathcal{X} = \mathcal{X}^1 \times \mathcal{X}^2$) and that both views are sufficient for learning the problem. Furthermore, the theoretical framework in [5] uses a very severe assumption: for every fixed example $\left(\hat{x}^1, \hat{x}^2\right) \in \mathcal{X}$ of non-zero probability it must hold that:

$$\begin{aligned} P\left(X^1 = \hat{x}^1 \mid X^2 = \hat{x}^2\right) &= P\left(X^1 = \hat{x}^1 \mid f^2\left(X^2\right) = f^2\left(\hat{x}^2\right)\right) \\ P\left(X^2 = \hat{x}^2 \mid X^1 = \hat{x}^1\right) &= P\left(X^2 = \hat{x}^2 \mid f^1\left(X^1\right) = f^1\left(\hat{x}^1\right)\right). \end{aligned} \tag{1}$$

In other words, that $X^1$and X^2 are conditionally independent given the label. As the authors themselves state, only four hypotheses comply with this assumption (assuming that P allows for it).

The Co-Training algorithm has been shown to produce better classifiers in the web-pages problem and in other experiments (for example, [10, 21], for more detailed analysis and limitations see [11, 14]). However, the theory presented can only be used as a motivation or a general intuition for the algorithm's success. Instead of using the training set to train only one of the learners and produce abundant newly-labeled examples for the second one, both learners are trained and subsequently label *some* unlabeled examples. These newly-labeled examples are then added to the pool of labeled examples, which is used to train the learners anew. Therefore, after being labeled, an unlabeled example assumes the same role as a labeled example: a true representation of the target function. As the authors themselves remark, this process encourages the learners to slowly *agree* on the labels of the unlabeled examples. This type of agreement is a side effect, if not a goal, of many variants of the co-training model [13, 12].

In this paper, we elaborate on this intuition and make it more precise. We show that agreement is useful and can assist in the task of learning. In other words, we present a theoretical framework where agreement between different learners has a clear advantage. Drawing upon these results, we propose a new boosting[1] algorithm—a field where our theoretical settings are especially applicable.

A similar attempt can be found in [15] where a boosting algorithm is presented, based on the above intuition. However, no proof is provided that the algorithm does result in agreeing classifiers nor for the advantage of such an agreement. A proof for the latter (in a more general settings) was provided by Dasgupta et al. in [16]. Nevertheless, for the proof to hold one still has to use the strong assumption of view-independence (Equation 1). Another example for the use of unlabeled examples in boosting can be found in [19].

2 The Value of Agreement

A typical approach in the supervised learning model is to design an algorithm that chooses a hypothesis that in some way best fits the training sample. We will show that an advantage can be gained by taking several such learning algorithms and demanding that they not only best learn the training set but also ‘agree’ with each other on a set of extra unlabelled examples.

The discussion below involves several learning algorithms and their accompanying hypothesis spaces. To avoid confusion, any enumeration or index that relates to different learners or hypotheses is enumerated using superscripts (typically l). All other indices, such as algorithm iterations and different examples, are denoted using a subscript.

2.1 Preliminaries

Since the learning algorithm is only given a finite sample of examples, it can only select a hypothesis based on limited information. However, the task of the algorithm is a global one. The resulting classifier f must perform well with respect to all examples in $\mathcal{X}$. The probability of error $P(\{(x,y) : f(x) \neq y\})$ must be small. In order to transfer the success of a classifier on the training set to the global case, there exist numerous *generalization bounds* (two such theorems will be given below). Typically these theorems involve some measure of the complexity or richness of the available hypothesis space. If the hypothesis space is not too rich, any hypothesis able to correctly classify the given examples cannot be to far from the target distribution. However, if H is very rich and can classify correctly any finite sample using different functions, success on a finite sample does not necessarily imply good global behavior.

As a complexity measure, we use the *Rademacher Complexity* (see [2]), which is particularly useful in the boosting scenario.

Definition 1. *Let $X_1, \ldots, X_n$ be independent samples drawn according to some distribution P on a set $\mathcal{X}$. For a class of functions F, mapping $\mathcal{X}$ to $\mathbb{R}$, define the random variable*

$$\hat{R}_n(F) = \mathbf{E}\left[\sup_{f \in F}\left|\frac{2}{n}\sum_{i=1}^{n}\sigma_i f(X_i)\right|\right]$$

[1] For an excellent introduction to boosting, the reader is referred to [1].

where the expectation is taken with respect to $\sigma_1, \ldots, \sigma_n$ *, independent uniform* $\{\pm 1\}$*-valued random variables. Then the* Rademacher complexity *of* F *is* $R_n(F) = \mathbf{E}\hat{R}_n(F)$ *where the expectation is now taken over* $X_1, \ldots, X_n$*.*

As an example, we present the following generalization bound(adapted from Theorem 3 in [1] and proved in [3]).

Theorem 1. *Let* F *be a class of real-valued functions from* $\mathcal{X}$ *to* $[-1,+1]$ *and let* $\theta \in [0,1]$*. Let* P *be a probability distribution on* $\mathcal{X} \times \{-1,+1\}$ *and suppose that a sample of N examples* $S = \{(x_1, y_1), \ldots, (x_{n_s}, y_{n_s})\}$ *is generated independently at random according to P. Then for any integer N, with probability at least* $1-\delta$ *over samples of length* n_s*, every* $f \in F$ *satisfies*

$$P(y \neq \mathrm{sign}(f(x))) \leq \hat{L}^{\theta}(f) + \frac{2R_{n_s}(F)}{\theta} + \sqrt{\frac{\log(2/\delta)}{2n_s}}$$

where $\hat{L}^{\theta}(f) = \frac{1}{n_s}\sum_{i=1}^{n_s} \mathbf{I}(y_i f(x_i) \leq \theta)$ *and* $\mathbf{I}(y_i f(x_i) \leq \theta) = 1$ *if* $y_i f(x_i) \leq \theta$ *and* 0 *otherwise.*

Theorem 1 introduces a new concept named *margin.*

Definition 2. *The* margin *of a function* $h : \mathcal{X} \to [-1,1]$ *on an example* $x \in \mathcal{X}$ *with a label* $y \in \{\pm 1\}$ *is* $yh(x)$*.*

Margins have been used to give a new explanation to the success of boosting algorithms, such as AdaBoost [17], in decreasing the global error long after a perfect classification of the training examples has been achieved [4]. Typically, one would expect a learning algorithm to eventually over-fit the training sample, resulting in an increase in global error[2].

Theorem 1 represents a rather general type of generalization bounds. Instead of assuming that the labels are generated by one of the hypotheses in H, it gives a connection between the empirical error on samples drawn from *any* distribution and the global expected error. As can be seen, the complexity of H plays a crucial role in this relation. Hence, if one was able to reduce the hypothesis space H without harming its ability to fit the sampled data, the resulting classifier is expected to have a smaller global error.

2.2 Formal Settings

Let $H^1, \ldots, H^L$ be a set of hypothesis spaces, each with a fitting learning algorithm, A^l. Further suppose that all learning algorithms are forced to agree and output hypotheses that agree with probability 1. If it is further assumed that the hypothesis that best fits the training set belongs to every H^l (thus available to all algorithms), this scheme produces a set of hypotheses from a potentially much *smaller* hypothesis spaces which are just as

[2] AdaBoost does eventually over-fit the data, if run long enough. However this happens at a much later stage then originally expected.

good on the training sample. Hence, the generalization capability of such hypotheses, as drawn from theorems such as Theorem 1, is potentially much better than the hypotheses outputted from any algorithm operating alone.

While the above discussion would yield the expected theoretical gain, it is very hard to implement. First, demanding that the algorithms output hypothesis that agree with probability 1 entails an ability that is unlikely to be easily available. Typically the different hypothesis spaces would consist of classifiers as different as neural networks and Bayes classifiers. It is unrealistic to demand that the hypothesis spaces will have an intersection which is rich enough to be useful to correctly classify different target distributions. While this might be feasible for $L = 2$ (such as the assumption in [5]) it is highly unlikely for a bigger number of learners. We will therefore present a more relaxed agreement demand, along with a simple way of checking it: unlabeled examples.

Definition 3.

1. *Define the* variance *of a vector in* $\mathbb{R}^L$ *by* $V(y^1,\ldots,y^L) = \frac{1}{L}\sum_{l=1}^{L}\left(y^l\right)^2 - \left(\frac{1}{L}\sum_{l=1}^{L} y^l\right)^2$.
2. *Furthermore, define the variance of a set of classifiers* $f^1,\ldots,f^L$ *to be the expected variance on examples from* X*, i.e.,* $V\left(f^1,\ldots,f^L\right) = \mathbf{E}V\left(f^1(x),\ldots,f^L(x)\right)$.

We will use the variance of a set of classifiers in the following relaxed definition of intersection as a measures of their disagreement.

Definition 4. *For any* $\nu > 0$*, define the* ν-intersection *of a set of hypothesis spaces,* $H^1,\ldots,H^L$ *to be:*

$$\nu - \bigcap_{l=1}^{L} H^l = \left\{f^1,\ldots,f^L : \forall l,\, f^l \in H^l,\ and\, V(f^1,\ldots,f^L) \le \nu\right\}.$$

In effect, the ν-intersection of $H^1,\ldots,H^L$ contains all the hypotheses whose difference with some of the members of other hypothesis spaces is hard to discover. We use this relaxed definition of intersection as the space from which the algorithms can draw their hypotheses. Note that for $\nu = 0$, the 0-intersection is precisely the set of hypotheses that might be outputted when the algorithms are required to agree with probability 1.

As mentioned before, unlabeled examples will be used to measure the level of agreement between the various learners. Therefore, let $U = \left\{u_j\right\}_{j=1}^{n_u}$ be a set of unlabeled examples, drawn independently from the same distribution P but without the label being available. We first show that if enough unlabeled examples are drawn, the disagreement measured on them is a good representative of the global disagreement. To this end we define a new hypothesis space $V(H^1,\ldots,H^L)$ and a target distribution $\tilde{P}$ to be used in a generalization bound resembling Theorem 1.

Definition 5.

1. *Let* $V\left(H^1,\ldots,H^L\right) = \left\{V \circ \left(f^1,\ldots,f^L\right) : f^1 \in H^1,\ldots,f^L \in H^L\right\}$ *where* $V \circ \left(f^1,\ldots,f^l\right) : \mathcal{X} \to [0,1]$ *is defined by:* $V \circ \left(f^1,\ldots,f^L\right)(x) = V\left(f^1(x),\ldots,f^L(x)\right)$
2. *Let* $\tilde{P}$ *be a probability distribution over* $\mathcal{X} \times [0,\infty]$ *which is defined by:*

$$\forall A \subseteq \mathcal{X} \times [0,\infty]\ \tilde{P}(A) = P\left(\{(x,y) \in \mathcal{X} \times \mathcal{Y} : (x,0) \in A\}\right).$$

In essence, $\tilde{P}$ labels all examples in $\mathcal{X}$ with 0, while giving them same marginal probability as before.

Before we can use the generalizing bound, we need to establish the Rademacher complexity of the new hypothesis space $V(H^1,\ldots,H^L)$.

Lemma 1. $R_n\left(V\left(H^1,\ldots,H^L\right)\right) \leq 8\max_l R_n\left(H^l\right)$.

Proof. Using Theorem 12 from [2], which gives some structural properties of the Rademacher complexity, the result follows from the following fact: $V\left(H^1,\ldots,H^L\right) \subseteq \frac{1}{L}\sum_{l=1}^{L}\phi\left(H^l\right)+\left[-\phi\left(\frac{1}{L}\sum_{l=1}^{L}H^l\right)\right]$ where $F^1+F^2=\left\{f^1+f^2 : f^1\in F^1, f^2\in F^2\right\}$, $\phi(F)=\{\phi\circ f : f\in F\}$ and $\phi(z)=z^2$. Note that ϕ is Lipschitz on $\mathcal{Y}\subseteq[-1,+1]$ with $L_\phi=2$.

Before proving the main theorems of the section, we present the following generalization bound (adapted from [2]). This theorem allows the use of an arbitrary loss function and does not use the concepts of margins.

Theorem 2. *Consider a loss function $\mathcal{L}:\mathcal{Y}\times\mathbb{R}\rightarrow[0,1]$ and let F be a class of functions mapping $\mathcal{X}$ to $\mathcal{Y}$. Let $\{(x_i,y_i)\}_{i=1}^{n}$ be a sample independently selected according to some probability measure P. Then, for any integer n and any $0<\delta<1$, with probability of at least $1-\delta$ over samples of length n, every $f\in F$ satisfies*

$$\mathbf{E}\mathcal{L}(Y,f(X)) \leq \hat{\mathbf{E}}_n\mathcal{L}(Y,f(X)) + R_n\left(\tilde{L}\circ F\right) + \sqrt{\frac{8\log(2/\delta)}{n}}$$

where $\hat{\mathbf{E}}_n$ is the expectation measured on the samples and

$$\tilde{L}\circ F = \{(x,y)\mapsto\mathcal{L}(y,f(x))-\mathcal{L}(y,0) : f\in F\}.$$

The scene is now set to give the first of the two main theorems of this section—a connection between function's agreement on a finite sample set and their true disagreement:

Theorem 3. *Let $H^1,\ldots,H^L$ be sets of functions from $\mathcal{X}$ to $\mathcal{Y}$ and let $U=\left\{u_j\right\}_{j=1}^{n_u}$ be a set of unlabeled examples drawn independently according to a distribution P over $\mathcal{X}\times\mathcal{Y}$. Then for any integer n and $0<\delta<1$, with probability of at least $1-\delta$ every set of functions $f^l\in H^l$, $l=1\ldots L$ satisfies:*

$$V(f^1,\ldots,f^L) \leq \hat{V}(f^1,\ldots,f^L) + 8\max_l R_{n_u}(H^l) + \sqrt{\frac{8\log(2/\delta)}{n_u}}$$

where $\hat{V}(f^1,\ldots,f^L)$ is the sampled expected variance, as measured on $U=\left\{u_j\right\}_{j=1}^{n_u}$.

Proof. The theorem follows directly from Theorem 2 when applied to the function set $V(H^1,\ldots,H^L)$ with $\tilde{P}$ as target distribution. The loss function is defined by $\mathcal{L}(y,z)=\min\{|y-z|,1\}$.

Theorem 3 allows us to use a finite set of unlabeled examples to make sure (with high probability) that the classifiers selected by the learning algorithms are indeed in the desired ν-intersection of the hypothesis spaces. This allows us to adapt generalization bounds to use smaller hypothesis spaces. As an example, we present an adapted version of Theorem 1.

Theorem 4. *Let $H^1,\dots,H^L$ be a class of real-valued functions from $\mathcal{X}$ to $[-1,+1]$ and let $\theta \in [0,1]$. Let P be a probability distribution on $\mathcal{X} \times \{-1,+1\}$ and suppose that a sample of n_s labeled examples $S = \{(x_j,y_j)\}_{j=1}^{n_s}$ and n_u unlabeled examples $U = \{u_j\}_{j=1}^{n_u}$ is generated independently at random according to P. Then for any integer n_s, $\nu > 0$, $0 < \delta < 1$ and n_u such that $8\max_l R_{n_u}(H^l) + \sqrt{\frac{8\ln(4/\delta)}{n_u}} \leq \frac{\nu}{2}$, with a probability at least $1-\delta$, every $f^1 \in H^1,\dots,f^L \in H^L$ whose disagreement $\hat{V}$ on U is at most $\frac{\nu}{2}$ satisfies*

$$\forall l\, P(y \neq \mathrm{sign}(f^l(x))) \leq \hat{L}^\theta(f^l) + \frac{2R_{n_s}(\nu - \bigcap_{\hat{l}} H^{\hat{l}})}{\theta} + \sqrt{\frac{\log(4/\delta)}{2n_s}}$$

where $\hat{L}^\theta(f^l) = \frac{1}{n_s}\sum_{j=1}^{n_s} \mathbf{I}\left(y_i f^l(x_i) \leq \theta\right)$.

Proof. By using Theorem 3 to reduce the hypothesis space, Theorem 1 can be applied to $v - \bigcap_{\hat{l}} H^{\hat{l}}$. By the union bound, the probability that the procedure fails is at most $\frac{\delta}{2} + \frac{\delta}{2} = \delta$.

To conclude this section, we note that the proposed settings has the following desired property: it doesn't help to have duplicate copies of the same hypothesis space. To have any advantage, $\nu - \bigcap_{\hat{l}} H^{\hat{l}}$ must be considerably smaller then any of the base hypothesis spaces. Therefore, using only duplicate copies of the same hypothesis space $H = H^1,\dots H^L$ gives $\nu - \bigcap_{\hat{l}} H^{\hat{l}} = H$ and hence no improvement. Furthermore, any duplicates within the set of different hypothesis spaces can be removed without changing the results.

2.3 Reduction of Labeled Examples

The previous section presented a formal setting where agreement was used to reduce the complexity of the set of possible hypotheses. The immediate implication is that training error serves as a better approximation for global true error. Therefore, for a given number of labeled examples, if the learning algorithm has produced a classifier with a low training error one can expect a lower global error. However this reduction in complexity can be also viewed from a different, though very related, point of view.

Since when given the right hypothesis space most algorithms can reduce the training error to a very low level, increasing the number of labeled examples gives a mean to decrease the two other terms in generalization bounds: the complexity of the hypothesis space and the certainty in the success of the whole procedure (δ). Using more labeled examples allows using a lower δ value without hindering the expected error of the resulting classifier (for example, Theorem 1 involves a $\sqrt{\frac{\log(2/\delta)}{2n_s}}$ term). The second result of increasing the number of labeled examples is reduction in the Rademacher complexity (or similar complexity terms). Therefore, decreasing the term relating to hypothesis space complexity, enables to use *less* labeled examples while achieving the same bound.

Algorithm 1. Agreement Boost

```
Denote $F\left(g^1,\dots,g^L\right)=\sum_{l=1}^{L}\sum_{j=1}^{n_s} er\left(-y_j g^l(x_j)\right)+\eta L\sum_{j=1}^{n_u} er\left(V\left(u_j\right)\right)$ where
$V(u)=\frac{1}{L}\sum_{l=1}^{L} g^l(u)^2-\left[\frac{1}{L}\sum_{l=1}^{L} g^l(u)\right]^2$, $\eta\in\mathbb{R}^+$ is some positive real number and $er:\mathbb{R}\to\mathbb{R}$ is
some convex, strictly increasing function with continuous second derivative.

1. Set $g^l\equiv 0$ for $l=1\dots L$.
2. Iterate until done (counter $t$):
   (a) Iterate over $l=1\dots L$:
       i. Set $w(x_j)=er'\left(-y_j g^l(x_j)\right)y_j/Z$ for all $\left(x_j,y_j\right)\in S$ and
          $w(u_j)=2\eta\left|\frac{1}{L}\sum_{\hat{l}=1}^{L} g^{\hat{l}}(u_j)-g^l(u_j)\right| er'\left(V(u_j)\right)/Z$ for all $u_j\in U$ where $Z$ is a
          renormalization factor s.t. $\sum_{x_j} w(x_j)+\sum_{u_j} w(u_j)=1$.
          Use $y(u_j)=sign\left(\frac{1}{L}\sum_{\hat{l}=1}^{L} g^{\hat{l}}(u_j)-g^l(u_j)\right)$ as pseudo-labels for $u_j$.
      ii. Receive hypothesis $f_t^l$ from learner $l$ using the above weights and labels.
     iii. Find $\alpha_t^l\geq 0$ that minimizes $F\left(g^1,\dots,g^l+\alpha_t^l f_t^l,\dots,g^L\right)$.
      iv. Set $g^l=g^l+\alpha_t^l f_t^l$.
3. Output classifier $sign(g^l)$ whose error on the samples is minimal out of the $L$ classifiers.
```

To illustrate this consider Blumer et al. (Theorem 2.1.ii in [6]) concerning the simple case of consistent learners. With high probability, a sample of size $\max\left\{\frac{4}{\varepsilon}\log\frac{2}{\delta},\frac{8d}{\varepsilon}\log\frac{13}{\varepsilon}\right\}$ is sufficient to disqualify any function in H that is too 'far' from the target $\hat{f}$. If H is made smaller, the number of functions which need to be excluded is reduced. Therefore, less labeled examples are needed in order to exclude high error functions.

Generalization bounds such as those presented before typically deal with over-fitting using the following idea: if the algorithm is given enough labeled examples it will not over-fit. Since the training sample is representative enough of target function, specializing in it does no harm. In the extreme, this leads to theorems such as the one of Blumer et al. concerning consistent learning algorithms. In the setting proposed here, the learning algorithm needs not only fit its training data but also agree with a couple of other algorithms. If the algorithms are sufficiently different, forcing them to agree inhibits their specialization on the training data, allowing to use a less representative training sample, or less labeled examples.

3 The Algorithm

In this section, we propose a new boosting algorithm named AgreementBoost (Algorithm 1), which exploits the benefits suggested by the theory presented in the previous section. Like AdaBoost, the algorithm is designed to operate in Boolean scenarios where each example can belong to one of two possible classes denoted by ± 1.

As in many boosting algorithms, AgreementBoost creates combined classifiers or ensembles. However, instead of just one such classifier, AgreementBoost creates L ensembles, one for each hypothesis space. The ensembles are constructed using L underlying learning algorithms, one for each of the L hypothesis spaces $\left\{H^l\right\}_{l=1}^{L}$. At each iteration, one of the learning algorithms is presented with a weighing of both labeled and unlabeled examples in the form of a weight vector $w(x)$ and pseudo-labels for the unlabeled examples $(y(u))$. The underlying learner is then expected to return a hypothesis f_t^l with a near-optimal[3] edge: $\gamma = \sum_{(x_j,y_j)\in S} w(x_j)y_j f^l(x_j) + \sum_{u_j\in U} w(u_j)y(u_j)f^l(u_j)$.

The proposed AgreementBoost can be described as a particular instance of AnyBoost [18], a boosting algorithm allowing for arbitrary cost functions. AgreementBoost's cost function F has been chosen to incorporate the ensembles' disagreement into the normal margin terms. This is achieved using a weighted sum of two terms: an error or margin-related term $(\sum_{l=1}^{L}\sum_{j=1}^{n_s} er\left(-y_j g^l(x_j)\right))$ and a disagreement term $\sum_{j=1}^{n_u} er\left(V\left(u_j\right)\right)$. Despite of the fact that the these terms capture different notions, they are very similar. Both terms use the same underlying function, $er(x)$, to assign a cost to some example-related measure: The first penalizes low (negative) margins while the second condemns high variance (and hence disagreement). AgreementBoost allows for choosing any function as $er(x)$, as long as it is convex and strictly increasing. This freedom allows for using different cost schemes and thus for future cost function analysis (as done, for example, in [18]). In the degenerate case where no unlabeled examples are used ($n_u = 0$) and e^x is used as $er(x)$, AgreementBoost is equivalent to L independent runs of AdaBoost (using the L underlying learners).

4 Proof of Convergence

In this section, we give a convergence proof for Algorithm 1. The proof considers two scenarios. The first assumes that the intersection of all $conv\left(H^l\right)$ is able to correctly classify all labeled examples using classifiers which agree on all unlabeled examples. Under this assumption, we show that the algorithm will produce classifiers, which in the limit are fully correct and agree on all unlabeled examples. In other cases, where this assumption is not valid, the algorithm will produce ensembles which minimize a function representing a compromise between correctness and agreement.

Both Mason et al. [18] and Rätsch et al. [20] provide similar convergence proofs for AnyBoost-like algorithms. While both proofs can be used (with minor modifications) in our settings, they do not fully cover both scenarios. The proof in [20] demands that the sum of the α_t^l coefficients will be bounded and thus cannot be used in cases where the theoretical assumptions hold. This can be seen easily in the case of AdaBoost, where a fully correct hypothesis will be assigned an infinite weight. While AgreementBoost will never assign an infinite weight to a hypotheses (due to the disagreement term), it is easy to come up with a similar scenario where the coefficient sum grows to infinity. In [18], Mason et al. present a theorem very similar to Theorem 5 below. However, they assume that the underlying learner performs perfectly and always returns the *best* hypothesis from the hypothesis space. Such a severe assumption is not needed in the

[3] For the exact definition of 'near-optimal', see Section 4.

proof presented here. Furthermore, due to the generality of AnyBoost, the result in [18] apply to the cost function alone and is not translated back to training error terms.

The proofs below are based on two assumptions concerning the learning algorithms and the hypothesis spaces. It is assumed that when presented with an example set S and a weighing $w(x)$, the underlying learning algorithms return a hypothesis f^l whose edge is at least $\delta \max_{\hat{f} \in H^l} \left(\sum_{x_j \in S} w(x_j) y_j \hat{f}(x_i) \right)$, for some $\delta > 0$. The second assumption concerns the hypothesis spaces: it is assumed that for every l and every $f^l \in H^l$ the negation of f^l is also in H^l i.e.: $f \in H^l \Rightarrow -f \in H^l$. This allows us to use absolute value in the previous assumption:

$$\sum_{x_j \in S} w(x_j) y_j f^l(x_j) \geq \delta \max_{\hat{f} \in H^l} \left| \sum_{x_j \in S} w(x_j) y_j \hat{f}(x_i) \right| \text{ for some } \delta > 0.$$

In the Lemmas and Theorems to follow, we will sometimes assume that the hypothesis spaces are finite. Due to the fact that there is only finite amount of ways to classify a finite set of examples with a ±1 label, if some of the hypothesis spaces are infinite it will be indistinguishable when restricted to S and U. Therefore, without loss of generality, one can assume that the number of hypotheses is finite.

The convergence of the algorithm is proven taking a different point of view to the ensembles built by the algorithm. The ensembles can be seen as a mix of all possible functions in the hypothesis spaces rather then as an accumulation of hypotheses:

Definition 6.

1. *Let $H^l = \{f_i^l\}_{i \in I^l}$ be an enumeration of functions in H^l. One can rewrite the ensembles g^l built by AgreementBoost as functions from $X \times \mathbb{R}^{|H^l|}$ to $\mathbb{R}$: $g^l(x, \beta^l) = \sum_i \beta_i^l f_i^l(x)$ for $\beta^l = (\beta_1^l, \beta_2^l, \ldots) \in \mathbb{R}^{|H^l|}$ and $l = 1 \ldots L$. Further denote $\beta = (\beta^1, \ldots, \beta^L)$. Note that β_i^l is the sum of all α_t^l such that $f_t^l \equiv f_i^l$.*
2. *Let the* variance *of $g^1, \ldots, g^L$ on an example u be $V(u, \beta) = V\left(g^1(u, \beta^1), \ldots, g^L(u, \beta^L)\right)$.*
3. *Whenever it is clear from context what are the β parameters, $V(u)$ and $g^l(u)$ will be used for brevity.*
4. *Let $er : \mathbb{R} \to \mathbb{R}^+$ be a convex monotonically increasing function. Denoting $E(\beta) = \sum_{l=1}^{L} \sum_{j=1}^{n_s} er\left(-y_j g^l(x_j)\right)$ and $D(\beta) = L \sum_{j=1}^{n_u} er(V(u_j))$, the function F becomes $F(\beta) = E(\beta) + \eta D(\beta)$ for some $\eta > 0$.*

$F(\beta)$ represents a weighing between correctness and disagreement. $E(\beta)$, being a sum of loss functions penalizing negative margins, relates to the current error of the ensemble classifiers. $D(\beta)$ captures the ensembles' disagreement over the unlabeled examples.

Using the above notations and the new point of view, the edge of hypotheses becomes proportional to the partial derivative of $F(\beta)$ with respect to the corresponding coefficient. Replacing the examples' weight and labels according to the definition of AgreementBoost, we have that $\sum_{x_j \in S} w(x_j) y_j f_i^l(x_j) + \sum_{u_j \in U} w(u) y(u_j) f_i^l(u_j) = -\frac{1}{Z} \frac{\partial F}{\partial \beta_i^l}(\beta)$.

Therefore the underlying learners return hypotheses whose corresponding partial derivatives are bounded by $-\frac{\partial F}{\partial \beta_i^l}(\beta) \geq \delta \max_{\hat{i}} -\frac{\partial F}{\partial \beta_{\hat{i}}^l}(\beta) = \delta \max_{\hat{i}} \left| \frac{\partial F}{\partial \beta_{\hat{i}}^l}(\beta) \right|$. Note that this ensures that the partial derivative with respect to the returned function coefficient is non-positive and hence the choice of α_t^l in step 2.a.ii of Algorithm 1 is in fact the global optimum[4] over all $\mathbb{R}$. Since in every iteration only one coefficient is changed to a value which minimizes $F(\beta)$, Algorithm 1 is equivalent to a coordinate descent minimization algorithm (for more information about minimization algorithms see, for example, [8]).

As a last preparation before the convergence proof, we show that $F(\beta)$ is convex. Apart from having other technical advantages, this guarantees that the algorithm will not get stuck in a local minimum.

Lemma 2. *The function $F(\beta)$ is convex with respect to β.*

Lemma 3. *Let $\{\beta_n\}$ be a sequence of points generated by an iterative linear search algorithm A, i.e., $\beta_{n+1} = A(\beta_n)$ minimizing a non-negative convex function $F \in C^2$. Denote the direction in which the algorithm minimizes F in every step by $v_n = \frac{\beta_{n+1}-\beta_n}{\|\beta_{n+1}-\beta_n\|_\infty}$ and $F_n(\alpha) = F(\beta_n + \alpha v_n)$ (i.e., A minimizes $F_n(\alpha)$ in every iteration by a linear search). Then, if $\exists M, m > 0 \in \mathbb{R}^+$ such that $(\forall n) \left[m \leq \frac{d^2 F_n}{d\alpha^2}(\alpha) \leq M \right]$ for every 'feasible' α (i.e., when $F_n(\alpha) \leq F(\beta_n)$) then $\lim_{n\to\infty} \frac{dF_n}{d\alpha}(0) = 0$ and $\lim_{n\to\infty} \|\beta_{n+1} - \beta_n\|_\infty = 0$.*

Proof. The results is obtained by using a first order Tailor expansion of $F_n(\alpha)$ and bounding the remainder with m and M.

Theorem 5. *For some non-empty sets of labeled examples S and unlabeled examples U, suppose that the underlying learners are guaranteed to return a hypothesis $\hat{f}$ such that $\sum_x w(x) y_x \hat{f}(x) \geq \delta \left(\max_f \left| \sum_x w(x) y_x f(x) \right| \right)$ for some constant $\delta > 0$ and every weighing $w(x)$ of their examples. Further let $er : \mathbb{R} \to \mathbb{R}^+$ be a non constant convex monotonically increasing function such that:*

1. *$er \in C^2$ and $er'(0) > 0$.*
2. *$\exists M \in \mathbb{R}^+$ for which $er(x) \leq \max \left\{ L(|S| + \eta |U|) er(0), \frac{1}{\eta}(|S| + \eta |U|) er(0) \right\}$ implies that $er''(x) < M$.*

Then it holds that $\lim_{n\to\infty} \|\nabla F(\beta_n)\|_\infty = 0$.

Proof. We first transform the assumption with respect to $er(x)$ into the bounds necessary for Lemma 3, obtaining that $\lim_{n\to\infty} \frac{\partial F}{\partial \beta_{i_n}^{l_n}} = 0$ where l_n and i_n are the indices of the hypothesis returned by the underlying learner at iteration n. Using the assumptions with respect to the underlying learner, it follows that $\left(\forall i \in I^{l_n}\right) \left[\lim_{n\to\infty} \left| \frac{\partial F}{\partial \beta_i^{l_n}}(\beta_n) \right| = 0 \right]$. The proof is concluded using an induction on the distance of the hypothesis spaces from l_n, i.e., $H^{l_n}, H^{(l_n-1) \bmod L}, \ldots, H^{(l_n-L+1) \bmod L}$.

[4] This involves the convexity of $F(\beta)$ that will be discussed below.

Theorem 6. *Under the assumptions of Theorem 5 with the additional assumption that $er(x)$ is strictly monotonic and that all underlying hypothesis spaces are able to correctly classify the data using finite ensemble classifiers from the intersection of the hypothesis spaces, both the error and the disagreement of the ensemble classifiers constructed by Algorithm 1 converge to 0.*

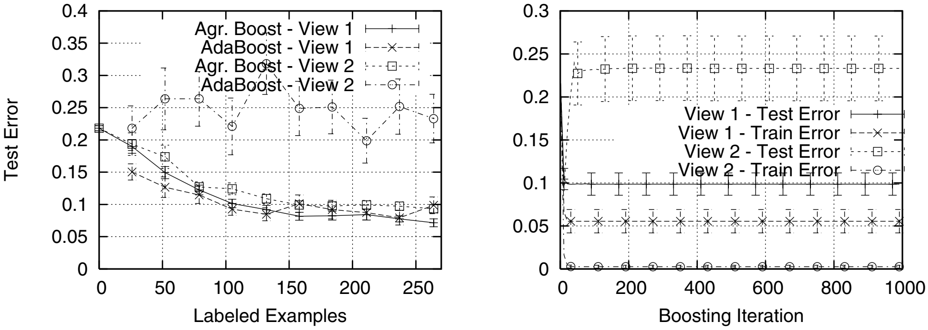
Proof. Denote the correct classifiers as $\tilde{g}^l = \sum \tilde{\beta}^l_i f^l_j$ and by $\tilde{\beta}$ the corresponding coefficient vector $(\tilde{\beta}^1_1, \ldots, \tilde{\beta}^L_{|H^L|})$. The correctness of the constructed classifiers is established by looking at the directional derivative $\frac{\partial F}{\partial \tilde{\beta}}$. Note that by the agreement of $\tilde{g}^l$, $\frac{\partial D}{\partial \tilde{\beta}} = 0$ and therefore $\frac{\partial F}{\partial \tilde{\beta}} = \frac{\partial E}{\partial \tilde{\beta}}$. The convergence of the disagreement term $D(\beta)$ is shown by deriving a contradiction. This is done by bounding the distance of β_n from the agreement group $B = \{\beta : \forall u \in U, V(u, \beta) = 0\}$. Suppose that a subsequence $D(\beta_{n_i}) > \varepsilon$ for some $\varepsilon > 0$. Since the tangent to a convex function is always an under estimator, the tangent to $D(\beta_{n_i})$ in the direction of B has to drop at least ε between β_{n_i} and the nearest point in B. This implies that it must have a negative slope that is bounded away from 0. However, Theorem 5 implies that the slope must converge to 0, which gives the contradiction.

5 Experiments

In this section we present a few experiments, testing the algorithm (and theory) presented in the previous sections. In these experiments, e^x was used as the loss function $er(x)$. This gives an algorithm which is very similar to AdaBoost, with the additional agreement requirement. In order to have a reference point, we compare the proposed AgreementBoost algorithm to AdaBoost, which is run separately on each of the underlying learning algorithms. In all experiments done, the η parameter is set using the following formula: $\eta = \frac{n_s}{n_u} c$, where c is some constant. This keeps the relative influence of the disagreement and training error terms in F roughly constant within a single series of experiments. This compensates for the fact that the number of labeled and unlabeled examples changes.

As a test case, we return to the problem of classifying web pages from the WebKb database presented in [5]. The WebKb database contains 1051 web pages, collected from the websites of computer science faculties of four different universities. For each web page, the database contains both the words contained in the page itself (referred to as View 1 in [5]) and words appearing in links referring to that web pages (View 2). The web pages are split into two classes: homepages of courses (230) and non-course pages (821). The goal of the learning algorithms presented in this section is to correctly classify web pages into these two classes.

In order to determine the quality of the resulting classifiers, 25% of the examples in the database were randomly selected in each experiment and held out as a test group. The experiments were repeated 20 times for each parameter set. All figures show the average result and its standard error of the mean.



(a) Test error: Agreement Boost vs AdaBoost

(b) Overfitting, AdaBoost, $n_s = 264$

Fig. 1. WebKb database, Naive Bayes applied to content and links

$\eta = 1 * \frac{n_s}{264}$, $n_u = 525$

5.1 Agreeing with the Village Fool...

The first set of experiments on the WebKb database mimics the experiments performed in [5]. The Naive Bayes algorithm is used as a single underlying learning algorithm, applied to each of the so called views: Page content and words on incoming links. This is done in a similar fashion to the toy problem, where AgreementBoost is run using the same learning algorithm on two different aspects of an example. AgreementBoost was allowed to run for 1000 iterations, using 525 unlabeled examples and setting $\eta = \frac{n_s}{264}$.

As can be seen in Figure 1, the classifiers built by AgreementBoost are roughly as good as the better of the two AdaBoost classifiers. Both AgreementBoost classifiers perform roughly the same as the AdaBoost classifier that uses the web pages' content.

One of the main assumptions used in Section 2 was that the underlying learners are all capable to produce a good classifier. However, as Figure 1(b) show, this is not the case in this experiment. While learning the links pointing to the pages produces a classifier with very low training error, it highly over-fits the data and has a very large test error. It is therefore not surprising that such a classifier has nothing to contribute. Nevertheless, AgreementBoost does seem to be able to 'choose' the better classifier. Despite of the fact that the two classifiers are forced to agree, the resulting consensus is as good as the better independent classifier.

5.2 Using a Better Learner

In light of the performance of the underlying links-based algorithm, it was replaced by a another learning algorithm which learns the web pages' content. This new underlying learner is based on a degenerate version of decision trees called tree stumps. Tree stumps consist of only one decision node, classifying an example only according to a single test. In these experiments, the web pages are classified by testing the number of instances of a single word within them. If the word has more instances then a given threshold, the web page is classified to one class and otherwise to the other.

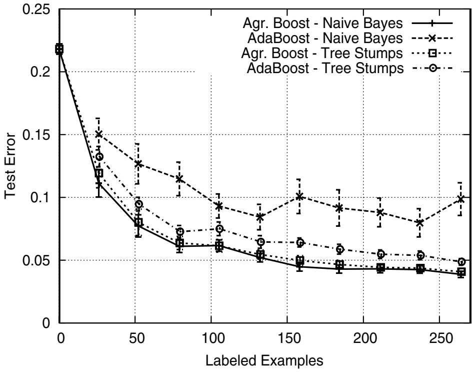


Fig. 2. Using Naive Bayes and Tree Stumps $\eta = 1 * \frac{n_s}{264}$, $n_u = 525$

The results of the experiments performed with the Tree Stumps algorithm are presented in Figure 2. In these experiments, 526 examples were used as unlabeled examples, allowing for up to 264 labeled examples. To perform a fair competition and to avoid over-fitting, the AdaBoost was run for only 300 iterations. As can be seen, AgreementBoost produces substantially better classifiers. On average, using the full 264 labeled example set, the tree stumps ensemble produced by AgreementBoost had 0.04 error on the test set. The naive Bayes classifier performed even better with a 0.038 test error. In comparison, the tree stumps ensemble constructed by AdaBoost, which was better than the corresponding naive Bayes classifier, had a test error of 0.049.

In terms of labeled examples reduction, AgreementBoost has also produced good results. The final test error achieved by AdaBoost using the full labeled exampled set (264 examples), was already achieved by AgreementBoost's classifiers using 158 labeled examples—a reduction of 40%.

6 Conclusions and Discussion

In the first section of this paper, we have proven a new generalization bound where unlabelled examples are used to reduce the penalty corresponding to hypothesis space complexity. Demanding from the underlying learners to agree limits the amount of hypotheses at their disposal and thus reduces the complexity of their effective hypothesis spaces. However, the theorems do not allow to foresee nor to estimate the magnitude of the improvement. In the set of experiments which we have performed, a reduction of up to 40% was observed in the number of labeled examples necessary in order to achieve a desired classification error. More theoretical and experimental work is needed to better quantify this advantage.

While agreement successfully reduces the number of labeled examples, it is not without a price. Increasing the importance assigned to the learners' agreement causes a reduction in the algorithm's convergence speed. Since AgreementBoost constructs its ensembles iteratively, this results in larger and computationally more expensive classifiers. The exact trade-off between agreement weight and convergence speed is yet to be established.

When designing AgreementBoost, we have opted for simplicity and thus avoided using many of the possible improvements and modifications, many of which are non-trivial and justify new research projects. We name a few:

1. Many of the improvements of AdaBoost suggested in the literature can adapted for AgreementBoost. Modifications like regularization terms for the hypotheses weights and soft margins will probability improve that algorithm's performance.
2. For simplicity, we have kept the agreement weight (η) constant along the run. However, we suspect that changing it during the algorithm's run might lead to superior results.
3. Following previous work, we have performed all experiments using only two underlying learners. However, the theoretical framework is quite more general, allowing for an arbitrary number of underlying learners. Further experimental study involving more learners is required.

Acknowledgement. I am grateful to Leen Torenvliet, Peter Grünwald, Pieter Adriaans and Peter van Emde Boas for their time, comments and helpful insights, leading to the final version of this work.

References

1. R.Meir and G. Rätsch: An Introduction to Boosting and Leveraging. *Advanced lectures on machine learning*. Pages: 118 - 183. ISBN:3-540-00529-3.
2. P.L. Bartlett and S. Mendelson: Rademacher and Gaussian Complexities: Risk bounds and Structural Results. *The Journal of Machine Learning Research*, Vol 3. 2003. Pages 463-482.
3. V. Koltchinskii and D. Panchenko: Empirical margin distributions and bounding the generalization error of combined classifiers. *The Annals of Statistic*s, 30(1), February 2002.
4. Robert E. Schapire, Yoav Freund, Peter Bartlett, and Wee Sun Lee: Boosting the margin: A new explanation for the effectiveness of voting methods. In *Machine Learning: Proceedings of the Fourteenth Fourteenth International Conference* 1997.
5. A. Blum and T. Mitchell: Combining labeled and unlabeled data with co-training. In *Proceedings of the 11th Annual Conference on Computational Learning Theory*. ACM, 1998.
6. Anselm Blumer and A. Ehrenfeucht and David Haussler and Manfred K. Warmuth: Learnability and the Vapnik-Chervonenkis dimension. *Journal of the ACM* Vol. 36, issue 4 1989.
7. K. Nigam, A. McCallum, S. Thrun, and T. Mitchell: Learning to classify text from labeled and unlabeled documents. In *Proc. of the* 5^{th} *National Conference on Artificial Intelligence*. AAAI Press, 1998.
8. D. Luenberger: Introduction to Linear and Nonlinear Programming. Addison-Wesley publishing company. 1973. ISBN 0-201-04347-5.
9. T. Zhang and F. Oles, A probability analysis on the value of unlabeled data for classification problems. In *Proc. of the Int. Conference on Machine Learning,* 2000.
10. Seong-Bae Park and Byoung-Tak Zhang: Co-trained support vector machines for large scale unstructured document classification using unlabeled data and syntactic information. In *Information Processing and Management: an International Journal* Vol. 40(3), 2004.
11. Kamal Nigam and Rayid Ghani: Analyzing the effectiveness and applicability of co-training. In *Proc. of the* 9^{th} *int. conference on Information and knowledge management* 2000.
12. Sally Goldman and Yan Zhou: Enhancing supervised learning with unlabeled data. In *International Joint Conference on Machine Learning*, 2000.
13. R. Hwa, M. Osborne, A. Sarkar, M. Steedman: Corrected Co-training for Statistical Parsers. In the *Proc. of the Workshop on the Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining*, International Conference of Machine Learning, Washington D.C., 2003.

14. D. Pierce and C. Cardie: Limitations of Co-Training for Natural Language Learning from Large Datasets. In *Proc. of the Conference on Empirical Methods in Natural Language Processing* 2001.
15. Michael Collins and Yoram Singer: Unsupervised models for named entity classification. In *Proc. of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, 1999.
16. S. Dasgupta, Michael L. Littman, David A. McAllester: PAC Generalization Bounds for Co-training. In *Advances in Neural Information Processing Systems 14* (2001).
17. Yoav Freund and Robert E. Schapire: A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1), 1997.
18. L. Mason, J. Baxter, P. L. Bartlett, and M. Frean. Boosting algorithms as gradient descent in function space. *Technical report, RSISE*, Australian National University 1999.
19. K. P. Bennett and A. Demiriz and R. Maclin: Exploiting unlabeled data in ensemble methods. In *Proceedings of the eighth ACM SIGKDD int. conference on Knowledge discovery and data mining,* 2002.
20. G. Rätsch, S. Mika, and M.K. Warmuth. On the convergence of leveraging. *NeuroCOLT2 Technical Report 98*, Royal Holloway College, London, August 2001.
21. A. Levin, P. Viola and Y. Freund: Unsupervised Improvement of Visual Detectors using Co-Training. *Int. Conference on Computer Vision* (ICCV), Oct 2003, Nice, France.