Semi-Supervised Random Forests*

Christian Leistner Amir Saffari Jakob Santner Horst Bischof
Institute for Computer Graphics and Vision
Graz University of Technology
{leistner,saffari,santner,bischof}@icg.tugraz.at

Abstract

Random Forests (RFs) have become commonplace in many computer vision applications. Their popularity is mainly driven by their high computational efficiency during both training and evaluation while still being able to achieve state-of-the-art accuracy.

This work extends the usage of Random Forests to Semi-Supervised Learning (SSL) problems. We show that traditional decision trees are optimizing multi-class margin maximizing loss functions. From this intuition, we develop a novel multi-class margin definition for the unlabeled data, and an iterative deterministic annealing-style training algorithm maximizing both the multi-class margin of labeled and unlabeled samples. In particular, this allows us to use the predicted labels of the unlabeled data as additional optimization variables. Furthermore, we propose a control mechanism based on the out-of-bag error, which prevents the algorithm from degradation if the unlabeled data is not useful for the task. Our experiments demonstrate state-of-the-art semi-supervised learning performance in typical machine learning problems and constant improvement using unlabeled data for the Caltech-101 object categorization task.

1. Introduction

There has been recent interest in using Random Forests (RFs) [4] for computer vision. RFs have demonstrated to be better or at least comparable to other state-of-the-art methods in both classification [4, 2] and clustering [12]. While obtaining state-of-the-art results, RFs possess several properties that make them particularly interesting for computer vision applications. First, they are very fast in both training and evaluation. Additionally, they can easily be parallelized, which makes them interesting for multi-core and GPU implementations [16]. RFs are inherently multi-class, therefore they do not require to build several binary classifiers for a multi-class problem. Compared to boosting and other ensemble methods, RFs are more robust against label noise [4].

In contrast, RFs suffer from the same disadvantages as other popular discriminative learning methods: they need a huge amount of labeled data in order to achieve good performance. This paper addresses this particular weakness by proposing a semi-supervised learning (SSL) [19, 6] algorithm for RFs allowing the algorithm to make use of both labeled and unlabeled training data. In fact, the speed, the inherent parallelism as well as the insensitivity to label noise makes RF an interesting candidate for SSL.

Another problem with most SSL methods is that they only focus on binary problems. Multi-class problems are often decomposed to a set of binary tasks with 1-vs-all or 1-vs-1 strategies. Considering the fact that most of the state-of-the-art SSL methods have high computational complexity, such a strategy can become a problem when dealing with a large number of samples and classes. Therefore, the ability of RFs to handle multi-class tasks makes them very attractive for SSL problems.

When training RFs, the individual trees have to be kept as diverse as possible while their agreement among the labeled samples is enforced. Thus, in order to incorporate unlabeled data one of the main obstacles is to preserve the diversity of the trees while making them agree on unlabeled data in order to increase the margin. This is the main reason why

*The first and second author contributed equally to this paper. This work has been supported by the Austrian Joint Research Project Cognitive Vision under projects S9103-N04 and S9104-N04, the FFG project EVis (813399) under the FIT-IT program and the Austrian Science Fund (FWF) under the doctoral program Confluence of Vision and Graphics W1209.

traditional regularized loss functions or previous approaches for SSL with trees (*e.g.*, [9]) cannot be directly applied to RFs because regularization usually destroys the diversity. Therefore, we propose a SSL algorithm for RFs that is based on Deterministic Annealing [13]. Using this approach, labels of the unlabeled data can be efficiently treated as additional optimization variables. The margin can be used as the regularization term for the unlabeled samples. Due to the efficiency of RFs the whole algorithm is quite fast during training. Another nice feature of the resulting algorithm is that by measuring the out-of-bag-error we can monitor the usefulness of the unlabeled data (we call this "airbag mechanism"). We derive the new algorithm theoretically and demonstrate in various experiments the advantages of our method compared to other SSL algorithms such as semi-supervised Boosting and TSVMs.

In Section 2.1, we present a brief overview on semi-supervised learning methods and RFs. In Section 3, we derive our new semi-supervised learning algorithm for random forests. Experimental results on Caltech 101 and machine learning datasets, comparisons to other SSL approaches and a detailed empirical analysis of our approach are presented in Section 4. Finally, the paper concludes with Section 5.

2. Related work & Notations

2.1. SSL Overview

In supervised learning one deals with a labeled dataset $\mathcal{X}_l \subseteq \mathcal{X} \times \mathcal{Y} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_{|\mathcal{X}_l|}, y_{|\mathcal{X}_l|})\}$, $\mathbf{x}_i \in \mathcal{X} \subseteq \mathbb{R}^M$ and $y_i \in \mathcal{Y} = \{1, \cdots, K\}$, where K is the number of classes. In contrast, unsupervised methods aim to find an interesting (natural) structure in $\mathcal{X}$ using only unlabeled input data $\mathcal{X}_u \subseteq \mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_{|\mathcal{X}_u|}\}$. In semi-supervised learning (SSL), the algorithm is provided with both labeled $\mathcal{X}_l$ and unlabeled $\mathcal{X}_u$ data (usually $|\mathcal{X}_l| \ll |\mathcal{X}_u|$). Since the unlabeled data can be obtained significantly easier than labeled samples, the main goal is to exploit the statistics of $\mathcal{X}_u$ in order to decrease the number of required labeled samples.

Many different approaches to SSL (*e.g.*, see [19, 6] for an overview) have been proposed, most methods assume an underlying structure that correlates the unlabeled data with some class label and, hence, makes them informative.

Many semi-supervised learning algorithms use the unlabeled samples to regularize the supervised loss function in the form of:

$$\sum_{(\mathbf{x},y)\in\mathcal{X}_l} \ell(y, h(\mathbf{x})) + \lambda \sum_{\mathbf{x}\in\mathcal{X}_u} \ell_u(h(\mathbf{x})), \quad (1)$$

where $h(\cdot)$ is a binary classifier, and $\ell_u(\cdot)$ encodes the regularizer based on the unlabeled samples. The regularization paradigm can be further subdivided into two main approaches:

Manifold Assumption Some algorithms try to infer the *cluster* or *manifold* structure of the feature space with unlabeled samples and use it as an additional cue for the supervised learning process, for example *cluster kernels* [7], *label propagation* [20] or *Laplacian SVMs* [1]. The latter two are graph-based methods where ℓ_u has the form of

$$\ell_u(h(\mathbf{x})) = \sum_{\substack{\mathbf{x}'\in\mathcal{X}_u \\ \mathbf{x}'\neq\mathbf{x}}} s(\mathbf{x}, \mathbf{x}') \|h(\mathbf{x}) - h(\mathbf{x}')\|^2, \quad (2)$$

where $s(\mathbf{x}, \mathbf{x}')$ is a similarity function. Using this regularization term, one can enforce the classifier to predict similar labels if the samples are similar. While graph-based methods are quite powerful, the pair-wise terms increase their computational complexity.

Another interesting approach, termed expectation regularization (ER), was proposed by Mann *et al.* [11] and developed further for semi-supervised boosting by Saffari *et al.* [14, 15], which improves both computational efficiency and robustness compared to previous methods. ER is a method for exponential-family parametric models where the basic idea is to augment the label-likelihood objective function with a term that encourages the model predictions on unlabeled data to match prior expectations (*e.g.*, label priors).

Large Margin Approaches Another class of methods such as *Transductive Support Vector Machines* (TSVM) [8, 17], tries to maximize the margin of the unlabeled samples by avoiding dense regions of the feature space for the decision boundary. For example, variants of Transductive Support Vector Machines (TSVM) [8] maximize the margin for the unlabeled samples by

$$\ell_u(h(\mathbf{x})) = \max(0, 1 - |h(\mathbf{x})|). \quad (3)$$

2.2. Random Forests

A Random Forest (RF) [4] is an ensemble of decision trees. Each tree in the forest is built and tested independently from other trees, hence the overall training and testing procedures can be performed in parallel. During the training, each tree receives a new bootstrapped training set generated from the original training set by subsampling with replacement. We refer to those samples which are not included during the training of a tree as the *Out-Of-Bag* (OOB) samples of that tree. These samples can be used to compute the *Out-Of-Bag-Error* (OOBE) of the tree as well as for the ensemble which is an unbiased estimate of the generalization error [3].

During training, each decision node of the tree creates a set of random tests and then selectes the best among them according to some quality measurement (*e.g.*, information gain or Gini index). The trees are usually grown to their full size without pruning.

We denote the n^{th} tree of the ensemble as $f_n(\mathbf{x}) = f(\mathbf{x}, \theta_n) : \mathcal{X} \rightarrow \mathcal{Y}$, where θ_n is a random vector capturing the various stochastic elements of the tree (such as the randomly subsampled training set or selected random tests at its decision nodes). We also denote the entire forest as $\mathcal{F} = \{f_1, \cdots, f_N\}$, where N is the number of trees in the forest. We can write the estimated probability for predicting class k for a sample as

$$p(k|\mathbf{x}) = \frac{1}{N} \sum_{n=1}^{N} p_n(k|\mathbf{x}), \tag{4}$$

where $p_n(k|\mathbf{x})$ is the estimated density of class labels of the leaf of the n^{th} tree. The final multi-class decision function of the forest is defined as

$$C(\mathbf{x}) = \arg\max_{k \in \mathcal{Y}} p(k|\mathbf{x}). \tag{5}$$

Breiman [4] defined the classification *margin* of a labeled sample $(\mathbf{x}, y)$ as

$$m_l(\mathbf{x}, y) = p(y|\mathbf{x}) - \max_{\substack{k \in \mathcal{Y} \\ k \neq y}} p(k|\mathbf{x}). \tag{6}$$

It is obvious that for a correct classification $m_l(\mathbf{x}, y) > 0$ should hold. Therefore, the generalization error is given by

$$GE = E_{(\mathbf{X},Y)}(m_l(\mathbf{x}, y) < 0), \tag{7}$$

where the expectation is measured over the entire distribution of $(\mathbf{x}, y)$. It has been shown by Breiman [4] that this error has an upper bound in form of

$$GE \leq \bar{\rho} \frac{1 - s^2}{s^2}, \tag{8}$$

where $\bar{\rho}$ is the mean correlation between pairs of trees in the forest [1] and s is the strength of the ensemble (*i.e.*, the expected value of the margin over the entire distribution).

3. Semi-Supervised Learning with Random Forests

As discussed in Section 2.1, there exist two main regularization approaches to semi-supervised learning. Since we target applications with a large amount of data and manifold regularization leads to algorithms that are at least quadratic, *i.e.* $\mathcal{O}(n^2)$, in terms of number of samples, in this work, we chose to use a maximum margin approach. In particular, we propose to exploit the additional unlabeled data in order to maximize the margin over the entire random forest. Since RFs are multi-class classifiers, in the following we will define the margin maximizing properties of decision trees for multi-class problems and subsequently define the margin over the unlabeled samples. Based on that, we propose to optimize a regularized loss function with the usage of Deterministic Annealing.

3.1. Margin for Multi-Class Classification

Recently, Zou *et al.* [21] extended the concept of Fisher-consistent loss functions [10] from binary classification to the domain of multi-class problems. This concept provides an understanding about the success of margin-based loss functions and their statistical characteristics.

Let $\mathbf{g}(\mathbf{x}) = [g_1(\mathbf{x}), \cdots, g_K(\mathbf{x})]^T$ be a multi-valued function. $\mathbf{g}(\mathbf{x})$ is called a margin vector, if

$$\forall \mathbf{x} : \sum_{i=1}^{K} g_i(\mathbf{x}) = 0. \tag{9}$$

In such a case, one can define the margin for the i^{th} class as $g_i(\mathbf{x})$ and the true margin as $g_y(\mathbf{x})$. For a Fisher consistent loss function, this quantity is naturally linked to the optimal Bayes decision rule [21].

We define a loss function $\ell(g_y(\mathbf{x}))$ to be a margin maximizing loss if $\ell'(g_y(\mathbf{x})) \leq 0$ for all values of g_y. Therefore, an optimization based on this kind of loss

[1] The correlation is measured in terms of the similarities of the predictions.

functions will lead to maximizing the true margin. In this respect, the exponential loss of boosting, the hinge loss of SVM, and the negative log-likelihood loss of statistical models are all margin maximizing loss functions.

For decision trees, local tests at each node are selected based on a score which measures the purity of the node. Usual choices are the entropy ($\mathcal{L}(\mathcal{R}_j) = -\sum_{i=1}^{K} p_i^j \log(p_i^j)$) or the Gini index ($\mathcal{L}(\mathcal{R}_j) = \sum_{i=1}^{K} p_i^j(1-p_i^j)$), where p_i^j is the label density of class i in node j. In the following Theorem, we relate such scores to multi-class margin maximizing loss functions and show that one can pick any margin maximizing loss function and derive a local score measurement.

Theorem 3.1. *Given a margin maximizing loss function $\ell(g_y(\mathbf{x}))$, the local score for a decision node $\mathcal{R}_j$ is defined as $\mathcal{L}(\mathcal{R}_j) = \sum_{i=1}^{K} p_i^j \ell(p_i^j - \frac{1}{K})$.*

Proof. We can write the empirical loss at this node as

$$\mathcal{L}(\mathcal{R}_j) = \frac{1}{|\mathcal{R}_j|} \sum_{(\mathbf{x},y)\in\mathcal{R}_j} \ell(g_y(\mathbf{x})). \tag{10}$$

Defining the margin vector as $\mathbf{g}^j(\mathbf{x}) = [p_1^j - \frac{1}{K}, \cdots, p_K^j - \frac{1}{K}]^T$, we can develop the empirical loss as

$$\begin{aligned}
\mathcal{L}(\mathcal{R}_j) =& \frac{1}{|\mathcal{R}_j|} \sum_{(\mathbf{x},y)\in\mathcal{R}_j} \sum_{i=1}^{K} \mathbb{I}(y=i)\ell(p_i^j - \frac{1}{K}) \\
=& \frac{1}{|\mathcal{R}_j|} \sum_{i=1}^{K} \ell(p_i^j - \frac{1}{K}) \sum_{(\mathbf{x},y)\in\mathcal{R}_j} \mathbb{I}(y=i) \\
=& \sum_{i=1}^{K} p_i^j \ell(p_i^j - \frac{1}{K}).
\end{aligned}$$

□

Using the results of Theorem 3.1, we can see that the entropy score minimizes the calibrated negative log-likelihood while the Gini index is related to the hinge loss function. Thus, we can conclude that traditional decision trees greedily optimize multi-class maximum margin criteria.

3.2. Margin for the Unlabeled Data

Now that we have a better understanding with respect to the maximum margin behaviour of the decision trees, the extension of this concept to unlabeled data is straight-forward. In the absence of a label, there is no known true margin, therefore, we define the margin as:

$$m_u(\mathbf{x}_u) = \max_{i\in\mathcal{Y}} g_i(\mathbf{x}_u). \tag{11}$$

Note that the predicted label for an unlabeled sample is $C(\mathbf{x}) = \arg\max_{i\in\mathcal{Y}} g_i(\mathbf{x}_u)$. Hence, this is equivalent to the margin of the labeled samples, given in Eq (6), but only using the predicted label.

3.3. Learning

Similar to the traditional regularization-based semi-supervised learning methods, we also regularize the loss for the labeled samples with a loss over the unlabeled samples. Based on the definition of the margin for unlabeled samples, we use the same loss function used to grow the trees in a forest also to be the loss for the unlabeled samples. We can write the overall loss as

$$\begin{aligned}
\mathcal{L}(\mathbf{g}) =& \frac{1}{|\mathcal{X}_l|} \sum_{(\mathbf{x},y)\in\mathcal{X}_l} \ell(g_y(\mathbf{x})) + \\
&+ \frac{\alpha}{|\mathcal{X}_u|} \sum_{\mathbf{x}\in\mathcal{X}_u} \ell(m_u(\mathbf{x})),
\end{aligned} \tag{12}$$

where α defines the contribution rate of the unlabeled samples.

Note that this loss is convex when training only on labeled data (standard RF algorithm). Using additional unlabeled data makes the loss non-convex because also the labels of the unlabeled samples have to be optimized. We need a (global) optimization method over the forest that is highly resistant to local minima.

In this work, we formulate the optimization process in a Deterministic Annealing (DA) framework. DA is a homotopy method where an eventually difficult combinatorial optimization problem is rewritten in an easier form and then gradually deformed to its original version [13]. In particular, in a first step the discrete variables are treated as random variables over which a space of probability distributions is defined. In the second step, the original problem is replaced by a continuous optimization term. Although DA cannot guarantee a global optimal solution, the method has proven to be a both fast and robust optimization technique. Note that Sindhwani *et al.* [17] used a similar approach for developing a semi-supervised kernel machine. Using DA, we treat unknown labels of the unlabeled samples as additional optimization variables. Additionally, the random nature of DA allows us to keep high diversity

among the trees, which is a necessity (as can be seen in Eq.(8)) for an improvement of the generalization error of the forest.

3.3.1 Deterministic Annealing Optimization

We apply deterministic annealing to iteratively solve Eq (12), by introducing a distribution over the predicted labels of unlabeled samples, $\hat{\mathbf{p}}$, and enforcing a controlled uncertainty into the whole optimization process. We write the new loss function as

$$\begin{aligned}\mathcal{L}_{DA}(\mathbf{g},\hat{\mathbf{p}}) =& \frac{1}{|\mathcal{X}_l|}\sum_{(\mathbf{x},y)\in\mathcal{X}_l}\ell(g_y(\mathbf{x}))+ \\ &+\frac{\alpha}{|\mathcal{X}_u|}\sum_{\mathbf{x}\in\mathcal{X}_u}\sum_{i=1}^{K}\hat{p}(i|\mathbf{x})\ell(g_i(\mathbf{x}))+ \\ &+\frac{T}{|\mathcal{X}_u|}\sum_{\mathbf{x}\in\mathcal{X}_u}\sum_{i=1}^{K}H(\hat{\mathbf{p}}),\end{aligned} \tag{13}$$

where T is the temperature parameter and $H(\hat{\mathbf{p}}) = -\sum_{i=1}^{K}\hat{p}(i|\mathbf{x})\log(\hat{p}(i|\mathbf{x}))$ is the entropy over the predicted distribution. Note that when the temperature is high, the dominating term is the entropy which needs to be minimized. Hence, at such stages, the model will maintain a large amount of uncertainty. As the system cools down, by decreasing the temperature $T \mapsto 0$, the optimization process will mainly operate over the original loss function Eq (12).

For a given temperature level, the learning problem can be written as

$$(\mathbf{g}^*,\hat{\mathbf{p}}^*) = \arg\min_{\mathbf{g},\hat{\mathbf{p}}}\mathcal{L}_{DA}(\mathbf{g},\hat{\mathbf{p}}). \tag{14}$$

We use a two step algorithm to solve this optimization problem. At the first step, we fix the distribution $\hat{\mathbf{p}}$ and optimize the learning model, and at the second step, we fix the learning model and find the optimal distribution. Note that both individual steps are convex optimization problems.

In detail, for a given distribution over the unlabeled samples, we randomly choose a label according to $\hat{\mathbf{p}}$. We repeat this process independently for every tree in the forest. At this stage, the optimization problem for the n^{th} tree becomes

$$\begin{aligned}\mathbf{g}_n^* = \arg\min_{\mathbf{g}}\ &\frac{1}{|\mathcal{X}_l|}\sum_{(\mathbf{x},y)\in\mathcal{X}_l}\ell(g_y(\mathbf{x}))+ \\ &+\frac{\alpha}{|\mathcal{X}_u|}\sum_{\mathbf{x}\in\mathcal{X}_u}\ell(g_{\hat{y}_u}(\mathbf{x})),\end{aligned} \tag{15}$$

where $\hat{y}_u$ is the randomly chosen label for this sample according to the distribution $\hat{\mathbf{p}}$. Since the margin maximizing loss function is convex, this loss function is also convex.

After we trained the random forest, we enter the second stage where we find the optimal distribution according to

$$\begin{aligned}\hat{\mathbf{p}}^* =& \arg\min_{\hat{\mathbf{p}}}\ \frac{\alpha}{|\mathcal{X}_u|}\sum_{\mathbf{x}\in\mathcal{X}_u}\sum_{i=1}^{K}\hat{p}(i|\mathbf{x})\ell(g_i(\mathbf{x}))+ \\ &+\frac{T}{|\mathcal{X}_u|}\sum_{\mathbf{x}\in\mathcal{X}_u}\sum_{i=1}^{K}\hat{p}(i|\mathbf{x})\log(\hat{p}(i|\mathbf{x})).\end{aligned} \tag{16}$$

For each sample, we can take the derivatives w.r.t. each class of the distribution and set them to zero to find the optimal solution. In detail, we define:

$$h_i(\hat{\mathbf{p}},\mathbf{x}) = \hat{p}(i|\mathbf{x})(\alpha\ell(g_i(\mathbf{x})) + T\log(\hat{p}(i|\mathbf{x}))). \tag{17}$$

The derivatives of this function can be written as

$$\frac{dh_i}{d\hat{p}_i} = \alpha\ell(g_i(\mathbf{x})) + T\log(\hat{p}(i|\mathbf{x})) + T. \tag{18}$$

By setting the derivatives to zero, we get

$$\hat{p}^*(i|\mathbf{x}) = \exp(-\frac{\alpha\ell(g_i(\mathbf{x})) + T}{T})/Z(\mathbf{x}), \tag{19}$$

where $Z(\mathbf{x}) = \sum_{i=1}^{K}\hat{p}^*(i|\mathbf{x})$ is the partition function. Note again that when the temperature is high, the distribution is close to a uniform distribution, while at very low temperatures it simulates a Dirac delta function, which is the hard decision rule of Eq (5) over the unlabeled data.

3.4. Airbag

In semi-supervised learning, there is no guarantee that unlabeled data always helps [19, 6, 18], for instance if the problem structure is badly matched, if the unlabeled data is corrupted or from a different distribution, *etc.*. A nice aspect of our algorithm is that we directly monitor the strength of the ensemble by measuring the OOBE for the entire forest at each iteration. By considering Eq.(8) we can see that the strength of the forest has an inverse relationship with the generalization error (*i.e.*, the stronger the forest, the lower is its error). Furthermore, the OOBE has shown to be a good estimate of the generalization error [3].

We propose to monitor the dynamics of the OOBE of the ensemble. Let $e_{\mathcal{F}}^m$ be the OOBE of the forest

Algorithm 1 Semi-supervised Random Forests

Require: A set of labeled, $\mathcal{X}_l$, and unlabeled data $\mathcal{X}_u$.
Require: The size of the forest: N.
Require: A starting heat parameter T_0 and a cooling function $c(T, m)$
Train the RF: $\mathcal{F} \leftarrow$ trainRF($\mathcal{X}_l$).
Compute the OOBE: $e^0_{\mathcal{F}} \leftarrow$ oobe($\mathcal{F}, \mathcal{X}_l$).
Set the epoch: $m = 0$.
repeat
  Get the temperature: $T_{m+1} \leftarrow c(T_m, m)$.
  Set $m \leftarrow m + 1$.
  $\forall \mathbf{x}_u \in \mathcal{X}_u, k \in \mathcal{Y}$: Compute $p^*(k|\mathbf{x}_u)$.
  for n from 1 to N **do**
    $\forall \mathbf{x}_u \in \mathcal{X}_u$: Draw a random label, $\hat{y}_u$ from $p^*(\cdot|\mathbf{x}_u)$ distribution.
    Set $\mathcal{X}_n = \mathcal{X}_l \cup \{(\mathbf{x}_u, \hat{y}_u)|\mathbf{x}_u \in \mathcal{X}_u\}$.
    Re-train the tree: $f_n \leftarrow$ trainTree($\mathcal{X}_n$).
  end for
  Set $e^m_{\mathcal{F}} \leftarrow$ oobe($\mathcal{F}, \mathcal{X}_l$).
until Stopping condition
if $e^m_{\mathcal{F}} > e^0_{\mathcal{F}}$ **then**
  Reset the RF: $\mathcal{F} \leftarrow$ trainRF($\mathcal{X}_l$).
end if
Output the forest $\mathcal{F}$.

at iteration m. Then we monitor the improvements measured by $e^{m-1}_{\mathcal{F}} - e^m_{\mathcal{F}}$ and if this is not positive after a few trials, we stop the training and discard the latest forest.

To sum up, in contrast to most other SSL methods, semi-supervised random forests provide a principled way to detect if unlabeled data rather harm the system than help. As a reaction, we can use an *airbag mechanism* in order to stop the semi-supervised learning process and use only the labeled data. We call our method Deterministic Annealing based Semi-Supervised Random Forests (DAS-RF) and show the overall learning and airbag procedures in Algorithm 1.

4. Experiments

We conduct experiments on both machine learning and on challenging object category datasets. The main purpose is to proof the concept of our approach, analyze its empirical behavior and compare it to other SSL methods. For all experiments, we set the $\alpha = 0.1$ and used 100 trees.

Dataset	# Train	# Test	# Class	# Feat.
g50c	50	500	2	50
Letter	15000	5000	26	16
SensIt (com)	78823	19705	3	100

Table 1. Data sets for the machine learning experiments.

Method	SVM	TSVM	SER	RMSB	RF	DAS-RF
g50c	91.7	93.1	91.9	94.2	89.1	93.3
Letter	70.3	65.9	76.5	79.9	76.4	79.7
SensIt	80.2	79.9	81.9	83.7	76.5	84.3

Table 2. Classification accuracy (in %) for machine learning datasets. DAS-RF stands for our method.

4.1. Machine Learning Datasets

For fair comparison, we implemented the original random forest (RF) algorithm as proposed by Breiman [4] and evaluated it within the same framework as our DAS-RF. Also SERBoost [14] and RMSBoost [15] were evaluated in the same framework. For SVM and TSVM we used standard packages. We use the *g50c*, *Letter*, and *SensIt* datasets from the Semi-Supervised Benchmarks [6] and LibSVM repository [5]. A summary of these data sets is presented in Table 1. For *g50c*, we use the original splits. For the last two datasets, we randomly partition the original training set into two disjoint sets of labeled and unlabeled samples. We randomly select 5% of the training set to be labeled and assign the rest (95%) to the unlabeled set. We repeat this procedure 10 times and report the average classification accuracy in Table 2. As can be seen from this table, our method is always among the best two over these datasets with respect to other semi-supervised methods. Table 3 also shows the average computation time for these methods. It can also be seen, that DAS-RF is of course slower than the supervised methods, which comes from the fact that has to process 20 times more unlabeled data on the larger datasets. However, compared to the other semi-supervised methods, our method is faster in the presence of large amounts of data. Since our method is inherently parallel, we also implemented it on a GPU resulting in an *additional 3-times speed-up* compared to the CPU implementation.

4.2. Caltech-101

For performing the categorization experiments, we chose the popular Caltech-101 dataset consisting

Method	SVM	TSVM	SER	RMSB	RF	DAS-RF	GPU-DAS-RF
Letter	25	74	3124	125	35	72	29
SensIt	195	687	1158	514	125	410	137

Table 3. Computation (train+test) time (in seconds) for machine learning datasets. Compared to supervised RFs, our method is slower due to the iterative optimization over the unlabeled data but has the same speed during testing. Note that for the *g50c* data the computation times where similar for all algorithms.

Algorithm	$l = 15$	$l = 30$
RF	0.72	0.64
DAS-RF	0.70	0.60
LinSVM	0.74	0.65
improvement	2%	4%

Table 4. Comparison of RF and DAS-RF in terms of classification error over different numbers of labeled samples.

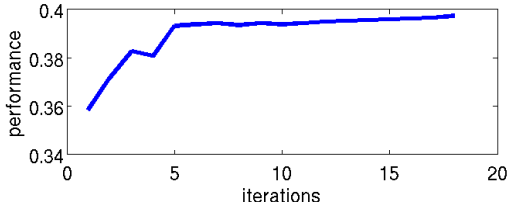

Figure 1. Improved average performance over the iterations for the multi-class problem.

Class	RF_{err}	DAS-RF_{err}	Relative Improvement
C4	0.0081	0.0033	58%
C5	0.0078	0.002	65%
C20	0.011	0.0013	87.5%
C33	0.007	0.003	52%
C81	0.0027	0.001	62.5%

Table 5. Comparison of RF with DAS-RF based on the binary classification error.

of 101 object categories with between 31 and 800 labeled samples per category. Bosch *et al.* [2] demonstrated state-of-the-art performance on that dataset using Random Forests. This dataset still provides a challenging benchmark for an inherently multi-class classifier.

In particular, for representation we use the L1-normalized PHOG [2] shape descriptors as introduced by Bosch *et al.* with 180 and 360 degrees, respectively. We trained RFs with 100 trees, using the information gain as node splitting criterion, ten random tests and a maximum tree depth of twenty. Additionally, in contrast to [2], we train multi-class RFs. For training, we use a randomly chosen subset of labeled data and all other samples as unlabeled data. Also, note that in this work, we use much weaker representations and less engineering for the sake of speed and clarity than, for instance, compared to [2].

For our SSL process, we allow a maximum of $m = 20$ iterations. For the cooling starting parameter we chose a simple exponential cooling function. We conduct our experiments with the typical amount of 15 and 30 labeled data, respectively. The final results are depicted in Table 4 while the improvement over the iterations is given in Figure 1. In these experiments, we also compare the results with the linear SVM for sanity check.

In the next experiment, we trained 100 1-vs.-all binary classifiers trained with 30 labeled samples and measured their improvements. For sake of space in Table 5 we depicted the five best improving binary classifiers. Note that on all classes we got an average improvement of 33% and never observed an increasing error rate during training with DAS-RF. The reason why the improvements here are much better than compared to the multi-class experiment is that the latter problem is much more difficult.

Airbag The purpose of this experiment is to show two things: First, it shows that if unlabeled data does not help, the dynamics of the OOBE can be used as a safety mechanism and, second, that trivial self-training RFs do not succeed on a diffucult multi-class categorization task. Hence, we trained two semi-supervised multi-class classifiers. However, while one was trained using the same settings as above the second one was trained on artificial corrupted unlabeled data. Additionally, we trained a RF performing self-learning on the (not corrupted) unlabeled data, *i.e.*, without using DA. The results are depicted in Figure 2. As can be seen, after 6 iterations the OOBE increases over a tolerance level from one iteration to the other one and we can automatically stop the training. As a result, we get the same performance as if only training on labeled data. The self-learning experiment fails even on not corrupted unlabeled data.

Discussion The experiments demonstrate that our method leverages a reasonable usage of unlabeled data for random forests. The ML experiments show

[2](10.3.2009) Available for download at http://www.robots.ox.ac.uk/ vgg/research/caltech/phog.html

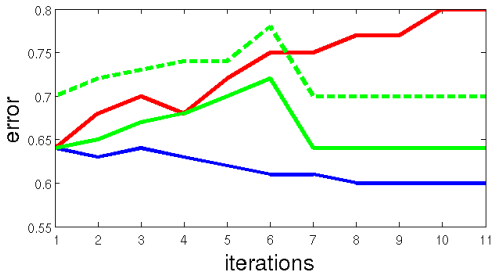

Figure 2. Training a DAS-RF on corrupted unlabeled data (green) and its OOBE (green dashed) and on not corrupted data (blue). After 6 iterations the SSL stops and it is trained only on labeled data. Self-learning is depicted in red.

that we perform comparable or even outperform other state-of-the-art ML methods in both speed and accuracy. Although our RF implementation cannot achieve the same results as reported in [2], we believe that a compareable improvement can be achieved using DAS-RF with better descriptors and data set-tuning techniques.

5. Conclusion

In this paper, we introduced a novel algorithm for semi-supervised learning with random forests. In order to incorporate unlabeled data we proposed a maximum margin approach using an iterative deterministic annealing-style optimization technique. The method is easy to implement, converges fast and is, in contrast to most other SSL methods, inherently multi-class. Additionally, the dynamic measurement of the out-of-bag-error allows for easy detection if the usage of unlabeled data might not help. The method showed state-of-the-art performance on machine learning datasets and constant improvement using unlabeled data over the Caltech-101 categorization task.

Additional incorporation of diversity among the trees should further improve the results. The usage of the algorithm with similar classifiers such as randomized ferns is straight-forward.

References

[1] M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *JMLR*, 7:2399–2434, 2006.

[2] A. Bosch, A. Zisserman, and X. Munoz. Image classification using random forests and ferns. In *ICCV*, 2007.

[3] L. Breiman. Out-of-bag estimates. Technical report, 1996.

[4] L. Breiman. Random forests. *Machine Learning*, 2001.

[5] C.-C. Chang and C.-J. Lin. Libsvm: a library for support vector machines. Software available at http://www.csie.ntu.edu.tw/˜cjlin/libsvm, 2001.

[6] O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-Supervised Learning*. Cambridge, MA, 2006.

[7] O. Chapelle, J. Weston, and B. Schölkopf. Cluster kernels for semi-supervised learning. In *NIPS*, volume 15 of *NIPS*, pages 585–592, 2003.

[8] T. Joachims. Transductive inference for text classification using support vector machines. In *ICML*, pages 200–209, 1999.

[9] C. C. Kemp, T. L. Griffiths, S. Stromsten, and J. B. Tenenbaum. Semi-supervised learning with trees. In *NIPS*, 2004.

[10] Y. Lin. A note on margin-based loss functions in classification. *Statistics & Probability Letters*, 68(1):73–82, June 2004.

[11] G. S. Mann and A. Mccallum. Simple, robust, scalable semi-supervised learning via expectation regularization. In *ICML*, pages 593–600, 2007.

[12] F. Moosmann, B. Triggs, and F. Jurie. Fast discriminative visual codebooks using randomized clustering forests. In *NIPS*, pages 985–992, 2006.

[13] K. Rose. Deterministic annealing, constrained clustering, and optimization. In *IJCNN*, 1998.

[14] A. Saffari, H. Grabner, and H. Bischof. Serboost: Semi-supervised boosting with expectation regularization. In *ECCV*, pages 588–601, October 2008.

[15] A. Saffari, C. Leistner, and H. Bischof. Regularized multi-class semi-supervised boosting. In *CVPR*, 2009.

[16] T. Sharp. Implementing decision trees and forests on a gpu. In *ECCV*, 2008.

[17] V. Sindhwani, S. S. Keerthi, and O. Chapelle. Deterministic annealing for semi-supervised kernel machines. In *ICML*, 2006.

[18] A. Singh, R. D. Nowak, and X. Zhu. Unlabeled data: Now it helps, now it doesn't. In *NIPS*, 2008.

[19] X. Zhu. Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison, 2005.

[20] X. Zhu and Z. Ghahramani. Learning from labeled and unlabeled data with label propagation. Technical report, 2002.

[21] H. Zou, J. Zhu, and T. Hastie. New multi-category boosting algorithms based on multi-category fisher-consistent losses. *Annals of Applied Statistics*, 2008.