2013 IEEE Conference on Computer Vision and Pattern Recognition

Semi-supervised Node Splitting for Random Forest Construction

Xiao Liu[†], Mingli Song[†], Dacheng Tao[‡], Zicheng Liu[♮], Luming Zhang[†], Chun Chen[†] and Jiajun Bu[†]
College of Computer Science, Zhejiang University[†]
Centre for Quantum Computation and Intelligent Systems, University of Technology Sydney[‡]
Microsoft Research, Redmond[♮]
{ender_liux, brooksong, zglumg, chenc, bjj}@zju.edu.cn[†]
dacheng.tao@gmail.com[‡], zliu@microsoft.com[♮]

Abstract

Node splitting is an important issue in Random Forest but robust splitting requires a large number of training samples. Existing solutions fail to properly partition the feature space if there are insufficient training data. In this paper, we present semi-supervised splitting to overcome this limitation by splitting nodes with the guidance of both labeled and unlabeled data. In particular, we derive a nonparametric algorithm to obtain an accurate quality measure of splitting by incorporating abundant unlabeled data. To avoid the curse of dimensionality, we project the data points from the original high-dimensional feature space onto a low-dimensional subspace before estimation. A unified optimization framework is proposed to select a coupled pair of subspace and separating hyperplane such that the smoothness of the subspace and the quality of the splitting are guaranteed simultaneously. The proposed algorithm is compared with state-of-the-art supervised and semi-supervised algorithms for typical computer vision applications such as object categorization and image segmentation. Experimental results on publicly available datasets demonstrate the superiority of our method.

1. Introduction

Random Forest (RF) has been applied to various computer vision tasks including target tracking, object categorization and image segmentation. Though it is one of the state-of-the-art classifiers, its promising performance depends heavily on the size of the labeled data. Because labeling training samples is very time consuming, only a small size of labeled training set is given in some tasks, which usually leads to an obvious performance drop. Thus, sometimes the insufficiency of labeled data is a severe challenging issue in the construction of RF. A popular solution to overcome this problem is to introduce abundant unlabeled data to guide the learning, which is known as semi-supervised learning (SSL). However, though many approaches have been given on SSL, few of them are applicable to RF. The only existing representative attempt is the Deterministic Annealing based Semi-Supervised Random Forests (DAS-RF) [14], which treated the unlabeled data as additional variables for margin maximization between different classes. Similar to many other margin maximization methods, finding the exact solution of DAS-RF is NP-hard. Although an efficient deterministic annealing optimization is used to search an approximate solution, it cannot provide robust results efficiently. In spite of this, it has been pointed out [26] that the effectiveness of margin maximization based methods for SSL depends heavily on specific data distribution which is usually difficult to be satisfied in many applications. Hence, it is desirable to find a method that allows RF to utilize the unlabeled data without losing its flexibility.

In this paper, by analyzing the construction of an RF using a small size of labeled training dataset, we find that the performance bottleneck is located in the node splitting. From this insight, we tackle the aforementioned problem by introducing abundant unlabeled data to guide the splitting. Based on kernel density estimation and the law of total probability, we derive a nonparametric algorithm to utilize abundant unlabeled data to obtain an accurate quality measure for node splitting. In particular, to avoid the curse of dimensionality, the data points are projected from the original high-dimensional feature space onto a low-dimensional subspace before estimating the categorical distributions. Finally, a unified optimization framework is proposed to select a coupled pair of subspace and separating hyperplane for each node such that the smoothness of the subspace and the quality of the splitting are guaranteed simultaneously.

Our contribution is three-fold:

- We experimentally show that node splitting quality is the performance bottleneck for constructing RF with a small size labeled training set.
- We show that partitioning an arbitrary feature space

1063-6919/13 $26.00 © 2013 IEEE
DOI 10.1109/CVPR.2013.70

with a hyperplane can be treated as projecting the data points from the original high-dimensional space onto the one-dimensional subspace that is perpendicular to the separating hyperplane. Thus a unified optimization framework is presented to choose a coupled pair of subspace and hyperplane such that the subspace is smooth and the hyperplane can effectively partition the feature space.

- We present an efficient nonparametric estimation-based semi-supervised splitting method to construct RF.

2. Related Work

Node splitting is the key issue of tree-based classifiers. Payne *et al.* [16] tried to build optimal binary trees such that the least number of tests are required to approach the leaf nodes. However, their tree construction is based on recursive dynamic programming and the algorithm is feasible only for a small number of features. Hyafil *et al.* [12] showed that the optimal sense of minimizing the expected number of tests required to classify an unknown sample is an NP-complete problem. Wu *et al.* [22] suggested a histogram-based splitting criterion for decision design. The histogram of training data is plotted on each feature axis. A threshold is selected to partition the classes. A limitation of this method is that only few features (usually one) are considered at each stage such that the interaction between features cannot be observed. Rounds [19] proposed Komogorov-Smirnov (K-S) distance and test as the splitting criterion. He suggested that the K-S distance between parts of the partition should be as large as possible. Breiman *et al.* [5] proposed to use the Gini index as the impurity measure for internal nodes. The goodness of a split is defined by the decrease in impurity. Suen *et al.* [21] proposed an entropy-based splitting criterion for decision tree construction. The entropy measure was later used to construct the well-known ID3 decision tree [17]. An improved version [18], namely C4.5 tree, was proposed in which the normalized information gain is used as the criterion of splitting.

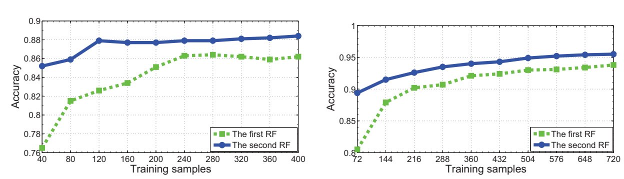
All the semi-supervised learning (SSL) methods rely on the smoothness assumption: if two points are close, the corresponding outputs should be similar [7, 24]. However, since the feature points are usually in high-dimensional space, one may have to face the curse of dimensionality to directly use the unlabeled data for estimation and thus there is insufficient number of observations to obtain a good estimation. Determined by how to address this problem, previous feasible discriminative SSL methods can be categorized into two families. The first family relies on the low density separation assumption: the decision boundary should lie in a low-density region. A classification margin for both unlabeled and labeled data is defined and maximized through global optimization. Typical methods in this family are Transductive Support Vector Machine (TSVM) [13] and DAS-RF [14]. The second family assumes that the high-dimensional data roughly lie on a low-dimensional manifold such that the unlabeled data can be efficiently used to infer the structure of the manifold without being troubled by the curse of dimensionality. A typical method in this family is LapSVM [2]. Both of the above two families predict the labels of unlabeled data as additional optimization variables, while the proposed method follows another line, *i.e.*, we project the data onto a low-dimensional subspace such that a small amount of data can give an accurate estimation.

3. Pre-analysis

The lack of training data influences RF construction in two ways: 1) the depth of the forest is limited and 2) the best splitting may not be chosen. Since the first influence is inevitable, we focus our effort on the second one in this paper.

Before doing this, it is necessary to understand the influence of splitting. Two RFs were constructed for comparison: the first RF was constructed conventionally only using a very small size labeled training set which may lead to bad splitting. The second RF was constructed using the same training set but an additional labeled set was used exclusively for better splitting. We constructed both RFs with 100 trees and used the popular entropy gain maximization criterion for splitting.

We used the Satimage and the Pendigits datasets from the LibSVM repository [6]. The comparison results of the two RFs are shown in Figure 1, where it can be seen that there are obvious performance improvements as a result of better splitting. From the comparisons above, it is obvious that the splitting quality is the performance bottleneck of RF construction when the size of the training set is small. For this reason, it is necessary to focus our effort on the node splitting strategies for RF construction.



(a) (b)

Figure 1. Recognition accuracy on (a) the Satimage dataset, and (b) the Pendigits dataset. The first RF was constructed conventionally only using a very small size of training data which may lead to bad splitting. The second RF was constructed using the same training set but an additional larger labeled set was used for better splitting.

4. Semi-supervised Node Splitting

RF consists of multiple decision trees: $F = \{t_1, t_2, \cdots, t_N\}$ and each is independently trained and tested.

In the RF construction stage, the algorithm learns a classification function $\mathcal{F} : \mathcal{X} \rightarrow \mathcal{Y}$ using the training samples $\{x_i \in \mathcal{X}\}_{i=1 \cdots l}$ and the corresponding labels $\{y_i \in \mathcal{Y}\}_{i=1 \cdots l}$, where $\mathcal{X} \subseteq \mathcal{R}^M$ is the feature space and $\mathcal{Y} = \{1 \cdots K\}$ is the label set. The trees are usually grown to the greatest possible extent without pruning. Each internal node of RF is binary split with a partition criterion. Each leaf node of RF is a voter which votes for the class into which the most samples fall.

During testing, given a test case x, RF gives the probability estimation for each class as follows

$$p(k|x) = \frac{1}{N} \sum_{i=1}^{N} p_i(k|x), \tag{1}$$

where $p_i(k|x)$ is the probability estimation of class k given by the i^{th} tree. It is estimated by calculating the ratio that class k gets votes from the leaves in the i^{th} tree

$$p_i(k|x) = \frac{l_{i,k}}{\sum_{j=1}^{K} l_{i,j}}, \tag{2}$$

where $l_{i,k}$ is the number of leaves in the i^{th} tree that vote for class k. The overall decision function of RF is defined as

$$\mathcal{F} = \underset{k \in \mathcal{Y}}{\operatorname{argmax}}\, p(k|x). \tag{3}$$

4.1. Semi-supervised Splitting

Considering both accuracy and time cost, oblique linear split is the most popular split strategy. An oblique linear split is expressed as a function of the hyperplane

$$W \cdot x = \theta, \tag{4}$$

where $W \in \mathcal{R}^M$ and $\theta \in \mathcal{R}$ are the parameters. When there is only one non-zero element in W, the split strategy focuses on a single attribute in one node [4, 10]. Otherwise, the combination of multiple attributes is considered [14, 11].

Given the data falling into a node and a candidate hyperplane, a quality measure needs to be defined such that one can search for the best hyperplane to maximize the splitting quality. There are four common criteria to evaluate the splitting quality, *i.e.*, information gain [17], normalized information gain [18], Gini index [5], and Bayesian classification error [9]. As the discussion given in Appendix A, the key procedure of all the four criteria is the estimation of p_k, which reflects the categorical distribution of the k^{th} class.

Traditionally, we can use the law of total probability to calculate p_k

$$p_k = \int_R p(k|x) dp(x). \tag{5}$$

Since both $p(x)$ and $p(k|x)$ are unknown, finite labeled samples are used to make the estimation

$$p_k = \frac{1}{\sum_{i=1}^{|R|} w_{R_i}} \sum_{i=1}^{|R|} p(k|x_{R_i}) w_{R_i}, \tag{6}$$

where w_{R_i} is the i^{th} sample falling into a node R and w_{R_i} is its weight. Since all the samples are labeled, $p(k|x_{R_i})$ is either 1 or 0

$$p(k|x_{R_i}) = [y_{R_i} = k]. \tag{7}$$

The problem with the fully supervised splitting is that, although the distribution $p(k|x_{R_i})$ is given by the labeled sample, the sparse labeled data cannot give a good approximation of the marginal distribution which may lead to a worse choice of the separating hyperplane. If only given few labeled samples, for example, one would choose to partition the two-dimensional space with the hyperplane shown in Figure 2(a). However, its estimation of the probability distribution is not good. In contrast, given more data, a better splitting can be found, as shown in Figure 2(b).

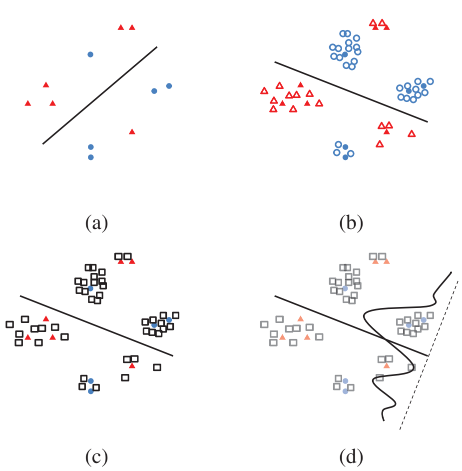


Figure 2. The red triangles and the blue circles are labeled samples of two classes while the black squares are unlabeled data. If only given a small number of labeled data, one may partition the feature space with the hyperplane shown in (a). When given more labeled data, one could find a better splitting strategy as in (b). Even the abundant data are unlabeled one can still choose the appropriate separating hyperplane as in (c) by combining the law of total probability and the kernel-based density estimation. As shown in (d), our method goes a bit further. We carry out the kernel-based density estimation in a one-dimensional subspace of the original feature space. Through this means, the curse of dimensionality can be avoided.

Unfortunately, the insufficiency of labeled training data usually leads to a sparse distribution and a bad approximation like Figure 2(a). Our solution to overcome the this limitation is to introduce abundant unlabeled samples to estimate p_k. The law of total probability is still used to calculate the probability distribution p_k over different classes

$$\hat{p}_k = \frac{1}{\sum_{i=1}^{|R|} w_{R_i}} \sum_{i=1}^{|R|} \hat{p}(k|x_{R_i}) w_{R_i}. \tag{8}$$

Since we now have many more data points, a much better approximation for the marginal distribution of $p(x)$ in R can be obtained. A new problem to arise is that the posteriori distribution $\hat{p}(k|x_{R_i})$ of unlabeled data is unknown. Since there are no priors of the categories, we can estimate $\hat{p}(k|x_{R_i})$ as the probability density ratio

$$\hat{p}(k|x_{R_i}) = \frac{p(x_{R_i}|k)}{\sum_{j=1}^{K} p(x_{R_i}|j)}. \tag{9}$$

For $p(x_{R_i}|k)$, we apply a kernel-based density estimation with Gaussian kernel [20]

$$K_h(u) = h^{-d}(2\pi)^{-d/2} \exp\left\{-\frac{1}{2h^2} u^T u\right\}, \tag{10}$$

where h is the bandwidth to be determined and d is the data dimension. We then have the following estimation for an unlabeled sample

$$p(x_{R_i}|k) = \frac{1}{n_k} \sum_{y_j=k} K_h(x_{R_i} - x_j), \tag{11}$$

where n_k is the number of samples that are labeled k.

4.2. Kernel Based Density Estimation on Low-dimensional Subspace

In this part, we derive a convenient formula to select the bandwidth h. The asymptotic mean squared error (AMISE) criterion [20] is applied to select the optimal bandwidth

$$AMISE(\mathbf{H}) = \frac{1}{4}\mu_2^2(K) \int \left[tr\left\{\mathbf{H}^T H_p(u) \mathbf{H}\right\}\right]^2 du + \frac{||K||_2^2}{n \det(\mathbf{H})}, \tag{12}$$

where $\mathbf{H}$ is the bandwidth matrix, n is the sample size, $\mathbf{H}_p(u)$ is the Hessian of the density $p(u)$ to be estimated and $\mu_2^2(K)$ is the squared variance of the kernel. In our case, K is the Gaussian kernel and the bandwidth matrix is $\mathbf{H} = h\mathbf{I}_d$. Since the true density $p(x|k)$ is unknown, we adopt the commonly used rule-of-thumb [20] to replace the unknown true density by a reference density $q(x)$, *e.g.*, a Gaussian distribution with its covariance matrix equal to the sample covariance. To simplify the calculation, we conduct whitening preprocessing before applying the estimation so that the sample covariance is the identity matrix, and thus $q(x) = \mathcal{N}(0, \mathbf{I})$. Based on the above discussions, AMISE for the conditional density $p(x|y = k)$ can be simplified as

$$AMISE(h) = \frac{2d + d^2}{2^d + 4\pi^{d/2}} + \frac{1}{2^d \pi^{d/2} n_k h^d}. \tag{13}$$

Taking the derivative with respect to h and setting it as 0, we obtain the optimal bandwidth

$$h_{opt} = \left(\frac{4}{(d+2)n_k}\right)^{1/(4+d)}. \tag{14}$$

Figure 2(c) shows a case where, by combining the law of total probability and the kernel-based density estimation, the appropriate separating hyperplane with abundant unlabeled data can be chosen.

Since the feature points are usually in very high-dimensional space, directly estimating the densities is still not a good idea, not only because of the intractable time complexity but also because there are insufficient labeled samples to make a good estimation in the high-dimensional space.

Our solution to this problem is to estimate the densities in a low-dimensional subspace rather than in the original high-dimensional feature space. Note that another way of looking at the hyperplane partition is that the original data are projected onto the one-dimensional subspace that is perpendicular to the separating hyperplane. In particular, by using a hyperplane $W \cdot x = \theta$ to partition the feature space, the algorithm makes a projection with the projection function

$$z = W \cdot x. \tag{15}$$

Furthermore, if the data can be well separated by the hyperplane, it is reasonable to believe that the metric in the subspace will reflect the intrinsic distance between data points. Thus, the labels of both labeled and unlabeled data distribute smoothly along the subspace. We can therefore estimate the density $p(k|z)$ instead of $p(k|x)$. In this case, the Gaussian kernel becomes

$$K_h(u) = \frac{1}{h\sqrt{2\pi}} \exp\left\{-\frac{1}{2h^2} u^2\right\}, \tag{16}$$

and the optimal choice of the bandwidth is

$$h_{opt} = \left(\frac{4}{3n_k}\right)^{1/5}. \tag{17}$$

We want to search for a coupled pair of subspace and separating hyperplane such that the smoothness of the subspace and the quality of the splitting are guaranteed simultaneously. An alternative optimization strategy is adopted to couple the two procedures by iterating the following two updating steps:

- Project the data onto the given subspace and estimate the categorical distribution.

- Search for a separating hyperplane according to the quality measure. Set the projection subspace as the perpendicular direction to the selected hyperplane.

Notice that we do not have to search for the optimal separating hyperplane, but only need to choose the best one from a candidate sets. We list the details of the proposed method in Algorithm 1.

Algorithm 1 Semi-supervised Splitting

Input: Labeled training data X_l that fall into R and the corresponding labels Y_l.
Input: Unlabeled training data X_u that fall into R.
Output: The parameters of the chosen hyperplane W_R and θ_R.

Randomly generate the set of parameters Ω for candidate hyperplanes.
Search for a hyperplane with parameters W^0 and θ^0 that maximize the quality measure considering only labeled data.
Set the iteration number t = 0.
repeat
  Set t = t + 1.
  Project all the samples onto the subspace that is perpendicular to the separating hyperplane: $z = W^t \cdot x$.
  for each labeled samples $x_i \in X_l$ **do**
    Use the given label as posterior distribution $p(k|z_i) = [y_i = k]$.
  end for
  for each unlabeled samples $x_i \in X_u$ **do**
    Calculate $p(k|z_i)$ with the kernel-based density estimation.
  end for
  Measure the split quality of each hyperplane.
  Choose the hyperplane parameters W^t and θ^t that maximize the quality measure.
until the chosen separating hyperplane is stable or the algorithm reaches enough iterations.
Set $W_R = W^t$ and $\theta_R = \theta^t$.
Return W_R and θ_R.

5. Random Forest Construction Based on Semi-supervised Splitting

In the RF construction stage, an individual training set for each tree is generated from the original training set using bootstrap aggregation. The samples which are not chosen for training are called Out-Of-Bag (OOB) samples of the tree and can be used for calculating the Out-Of-Bag-Error (OOBE), which is an unbiased estimation of the generalization error. It has been shown that the OOBE is an unbiased estimation of the generalization error [3]. Leistner *et al.* [14] first proposed the 'airbag' algorithm to use the OOBE to measure the performance of RF to detect whether the variation to the original algorithm would harm the system rather than assist. They decided whether to discard or retain a new forest by comparing its OOBE with the previous forest. A shortcoming of the 'airbag' approach is that it regards the RF as a whole, so some trees with high performance may be dragged down by bad trees.

To overcome the limitation of 'airbag' algorithm, we propose to independently compare the OOBE of single decision trees in the supervised and semi-supervised models. In each model, we use a set of labeled bootstrap data to construct the supervised decision tree and use the same set of labeled data and a set of unlabeled bootstrap data to construct the semi-supervised tree. The tree with smaller OOBE will be retained. The overall learning and control procedures of the proposed algorithm are shown in Algorithm 2.

Algorithm 2 Semi-supervised Splitting

Input: Labeled training data X_l and the corresponding labels Y_l.
Input: Unlabeled training data X_u.
Input: The size of the forest N.
Output: The learned RF F.

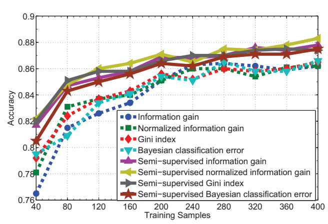
Initialize an empty forest F.
for the i^{th} decision tree in F **do**
  Generate a new labeled set X_l^i and a new unlabeled set X_u^i using the bootstrap aggregation.
  Train the tree with only labeled samples: $t_l^i = trainTree(X_l^i)$.
  Compute the OOBE: $e_l^i = oobe(t_l^i, X_l - X_l^i)$.
  Train the tree with both labeled and unlabeled samples: $t_u^i = semiTree(X_l^i, X_u^i)$.
  Compute the OOBE: $e_u^i = oobe(t_u^i, X_l - X_l^i)$.
  if $e_l^i > e_u^i$ **then**
    $F = F \cup t_u^i$.
  else
    $F = F \cup t_l^i$.
  end if
end for
Return F.

6. Experiment and Analysis

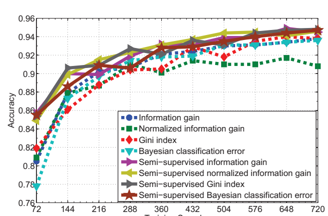
We compared the proposed semi-supervised splitting with different splitting criteria on typical machine learning tasks. We show that by introducing abundant unlabeled data, obvious accuracy improvement can be achieved. We also applied the proposed semi-supervised splitting RF for object categorization and image segmentation. Our method achieves state-of-the-art categorization performance on the Caltech-101 dataset [15] and segmentation performance on the MSRC dataset [8].

6.1. Data Classification

To quantitatively evaluate the improvement over the traditional splitting criteria, we test our method on the Satimage and Pendigits datasets. The Satimage dataset has 4435 training samples and 2000 testing samples while the Pendigits dataset has 7494 training samples and 3498 testing samples. We implement Breiman's Random Forests [4] with the four different splitting criteria, *i.e.*, information gain, normalized information gain, Gini index and Bayesian classification error. The traditional versions of these criteria are used as baselines. The proposed semi-supervised splitting was applied to the four criteria. For each dataset, we randomly chose a part of the training data as the labeled data and left the remainder as unlabeled data. The ratios of the chosen data range from 0.01 to 0.1. The weights of the labeled samples were set at 1.0 while the weights of the unlabeled samples were set through cross-validation. We built the RF with 100 trees and 10 hyperplanes were randomly generated as candidates in the internal node of RF. To avoid computation cost in testing, only two attributes were considered in each hyperplane, and the RF was constructed without pruning. The training-testing procedures were repeated 10 times. We show the comparison results in Figure 3 and it is clear that substantial performance improvement is achieved as a result of using our method.



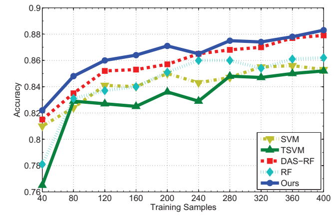
(a)



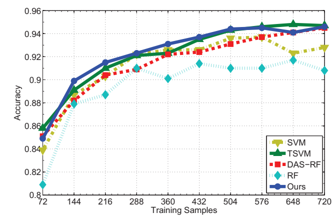
(b)

Figure 3. The classification accuracy of Random Forests with traditional splitting criteria (dashed lines) and the proposed semi-supervised splitting (solid lines). The results on the Satimage dataset and the Pendigits dataset are shown in (a) and (b) respectively.

We also compared the proposed semi-supervised splitting RF with the state-of-the-art semi-supervised and supervised classifiers: RF [4], TSVM [13], SVM [6] and DAS-RF [14]. The comparisons are shown in Figure 4. It can be seen that the proposed method performs the best among them.



(a)



(b)

Figure 4. The classification accuracy of the proposed method with state-of-the-art classifiers. The results on the Satimage dataset and the Pendigits dataset are shown in (a) and (b) respectively.

6.2. Object Categorization

We used the Caltech-101 dataset for the object categorization experiment. The Caltech-101 dataset is composed of 101 categories. Each class contains 31 to 800 images; most of the images are medium resolution, *i.e.* about 300×300 pixels. Following the common experiment setup, we randomly chose 15 images per category as the labeled training data and 15 images as the unlabeled training data. The remaining images were used for testing. We followed the ScSPM approach [23] in which each image was represented as a 21504-dimension feature vector. PCA was then applied to the resulting vector. Maintaining 99.5% energy, we obtained a 4000-dimensional feature vector for each image. We constructed a RF with 100 trees and used 100 candidate hyperplanes in the internal node. The weights of the unlabeled data were set at 0.5 while the weights of the labeled data were set at 1.0. Normalized information gain is used as the splitting criterion to construct the RF. The proposed method and the traditional RF achieved 69.3% and 64.4% categorization accuracy respectively while the accuracy of SVM was 65.8%. Using the same features, our method brought an improvement of 4.9 percent over the traditional RF and 3.5 percent over the SVM, by introducing abundant unlabeled data.

6.3. Image Segmentation

We applied the proposed semi-supervised splitting RF for the task of image segmentation in the 9-class MSRC dataset. The 9 classes are: building, grass, tree, cow, sky, aeroplane, face, car and bicycle. We randomly chose 150 images as the labeled training data and 150 images as the unlabeled training data from a total of 480 images, leaving the remainder as the testing data. We used the SLIC [1] to over-segment each image into 200 superpixels. The label of a superpixel was assigned as the majority of the pixel-level ground truth. We used the 9-dimensional color moment, 48-bins histogram of RGB, 9-bins histogram of gradient [25] and 59-bins histogram of LBP as the features. We constructed a RF with 100 trees, each having a maximum depth of 15. During splitting, 10 hyperplanes were randomly generated as candidates and the hyperplane that maximized the information gain was chosen. We set the weight of a labeled superpixel at 1.0 and set the weight of an unlabeled superpixel at 0.3 through cross-validation. Similar to most popular segmentation approaches, we applied a CRF stage after having learnt the class posteriors of the superpixels. The class posteriors from RF were used as unary potentials and pairwise potentials were defined over adjacent superpixels. The parameters of the CRF model were trained using the same 150 labeled training data used to construct the RF. When testing, the alpha-expansion graph-cut algorithm was used to infer the CRF model. We show the segmentation results of the proposed semi-supervised splitting RF in Figure 5. The original images and the ground truths are also shown. The segmentation accuracy of the proposed method and traditional RF is 87.5% and 83.6% respectively. As can be seen, our method is more accurate than the alternative method.

7. Conclusion and Future Work

In this paper, we introduced a semi-supervised splitting method that uses abundant unlabeled data to guide the node splitting of random forests. We derived a nonparametric algorithm to estimate the categorical distributions of the internal nodes such that an accurate quality measure of splitting can be obtained. Our method can be combined with many popular splitting criteria, and the experimental results show that it brings obvious performance improvements to all of them.

In the future, we would like to investigate the problem of constructing RFs without labeled data. A unified splitting framework that can handle both labeled and unlabeled data would be the extension.

ACKNOWLEDGEMENTS

This work was supported in part by the National Natural Science Foundation of China (61170142), by the National Key Technology R&D Program under Grant (2011BAG05B04), by the Program of International S&T Cooperation (2013DFG12841), and by the Fundamental Research Funds for the Central Universities (2013FZA5012). M. Song is the corresponding author.

Appendix A

We consider four usual choices: information gain, normalized information gain, Gini index and Bayesian classification error.

Information Gain:

The information gain is defined as the subtraction of entropies before and after splitting

$$\triangle H(R,W,\theta) = H(R) - \frac{|R_l|}{|R|}H(R_l) - \frac{|R_r|}{|R|}H(R_r), \quad (18)$$

where R is an internal node, R_l and R_r are its left and right child respectively. $H(R)$ is the Shannon entropy of R

$$H(R) = -\sum_{k=1}^{K} p_k \log p_k, \quad (19)$$

where p_k is the probability distribution of the k^{th} classes in R.

Normalized Information Gain:

The normalized entropy gain is defined as the quotient of the information gain and a normalized factor

$$\triangle N(R,W,\theta) = \frac{\triangle H(R,W,\theta)}{-\left(\frac{|R_l|}{|R|}\log\frac{|R_l|}{|R|} + \frac{|R_r|}{|R|}\log\frac{|R_r|}{|R|}\right)}. \quad (20)$$

Gini Index:

The Gini index is a measure of the impurity of a node and its definition is

$$G(R) = \sum_{k=1}^{K} p_k(1-p_k). \quad (21)$$

When used for qualifying splitting, the following function should be maximized

$$\triangle G(R,W,\theta) = G(R) - \frac{|R_l|}{|R|}G(R_l) - \frac{|R_r|}{|R|}G(R_r). \quad (22)$$

Bayesian Classification Error:

The Bayesian classification error of a node is defined as

$$C(R) = 1.0 - \operatorname*{argmax}_{k\in 1\cdots K} p_k. \quad (23)$$

When used for qualifying splitting, the following function needs to be maximized

$$\triangle C(R,W,\theta) = C(R) - \frac{|R_l|}{|R|}C(R_l) - \frac{|R_r|}{|R|}C(R_r). \quad (24)$$

From the above discussion, it is noticeable that all the four criteria heavily depends on the estimation of the probability distribution p_k.

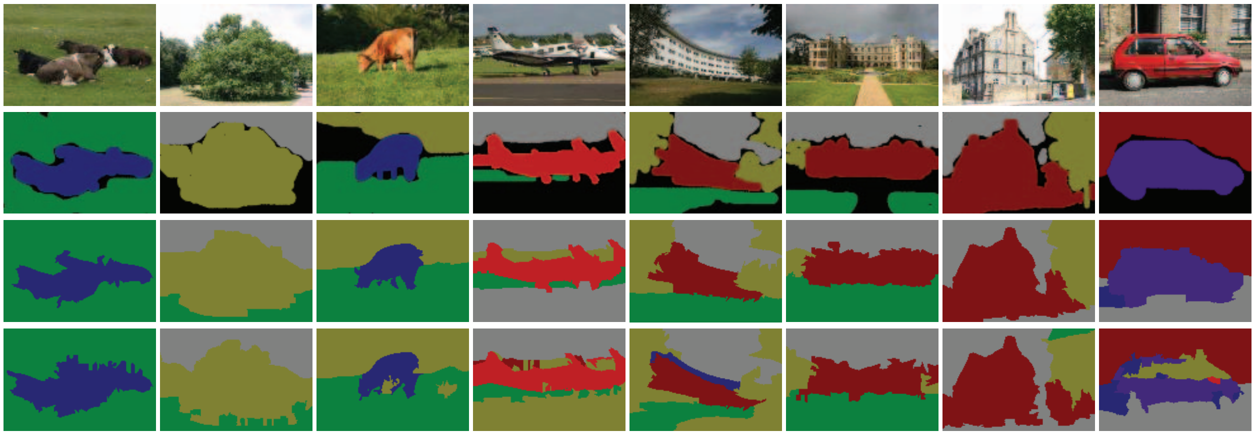


Figure 5. The original images, ground truths, the proposed segmentation results and traditional RF-based segmentation results on the MSRF dataset are shown from the top to the bottom row. The black area in the ground truth images is not labeled.

References

[1] R. Achanta, A. Shaji, K. Smith, A. Luchi, P. Fua, and S. Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34:2274–2282, 2012.

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

[3] L. Breiman. Out-of-bag estimates. *Technical report*, 1966.

[4] L. Breiman. Random forests. *Machine Learning*, 45:5–32, 2001.

[5] L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and regression trees*. CA: Wadsworth Int., Belmont, 1984.

[6] C.-C. Chang and C.-J. Lin. Libsvm: a library for support vector machines. *ACM Trans. Intell. Syst. Tech.*, 2, 2011.

[7] O. Chapelle, B. Schölkopf, and A. Zien. *Semi-supervised Learning*. The MIT Press, Cambridge, Massachusetts, Lodon, England, 2006.

[8] A. Criminisi. Microsoft research cambridge object recognition image dataset. version 1.0, 2004.

[9] R. Duda and P. Hart. *Pattern Classification and scene analysis*. Wiley-Interscience, 1973.

[10] P. Geurts, D. Ernst, and L. Wehenkel. Extremely randomized trees. *Machine Learning*, 63:3–42, 2006.

[11] T. Ho. The random subspace method for constructing decision forests. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20:832–844, Aug. 1998.

[12] L. Hyafil and R. Rivest. Constructing optimal binary decision trees is NP-complete. *Information Processing Letters*, 5:11–17, 1976.

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

[14] C. Leistner, A. Saffari, J. Santner, and H. Bischof. Semi-supervised random forests. *Proc. ICCV*, pages 506–513, Sept. 2009.

[15] F.-F. Li, R. Fergus, and P. Perona. Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 objct categories. *Proc. CVPR Workshop on Generative-Model Based Vision*, 2004.

[16] H. Payne and W. Meisel. An algorithm for constructing optimal binary decision trees. *IEEE Trans. Comput.*, C-26:905–516, Sept. 1977.

[17] J. Quinlan. Induction of decision trees. *Mach. Learn.*, 1:81–106, Mar. 1986.

[18] J. Quinlan. *C4.5: Programs for machine learning*. Morgan Kaufmann Publishers, 1993.

[19] E. Rounds. A combined non-parametric approach to feature selection and binary decision tree design. *Pattern Recognition*, 12:313–317, 1980.

[20] B. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman and Hall/CRC, Apr. 1986.

[21] C. Suen and Q. Wang. ISOETRP - an interactive clustering algorithm with new objectives. *Pattern Recognition*, 17:211–219, 1984.

[22] C. Wu, D. Landgrebe, and P. Swain. The decision tree approach to classification. *School Elec. Eng., Purdue Univ., Lafayette*, IN, Rep. RE-EE 75-17, 1975.

[23] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. *Proc. CVPR*, pages 1794–1801, 2009.

[24] Y. Yang, F. Nie, D. Xu, J. Luo, Y. Zhuang, and Y. Pan. A multimedia retrieval framework based on semi-supervised ranking and relevance feedback. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34:723–742, 2012.

[25] L. Zhang, M. Song, Z. Liu, X. Liu, J. Bu, and C. Chen. Probabilistic graphlet cut: Exploiting spatial structure cue for weakly supervised image segmentation. *Proc. CVPR*, 2013.

[26] T. Zhang and F. Oles. A probability analysis on the value of unlabeled data for classification problems. *Proc. ICML*, pages 1191–1198, 2000.