Decision Forests for Computer Vision and Medical Image Analysis: Chapter 4 Classification Forests

1 Classification Forests

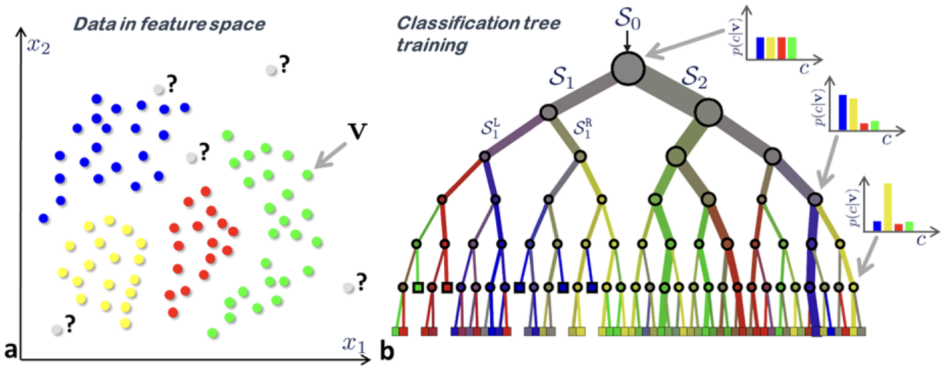


Fig. 4.1 Classification: training data and tree training. (**a**) Input data points are denoted with *circles* in their 2D feature space. Different colors denote different ground truth class labels. There are four classes here. *Gray circles* indicate unlabeled, previously unseen test data. (**b**) A binary classification tree. The edge thickness is proportional to the amount of training data going through it. Edge colors are a mix of the colors of the four classes, weighted in proportion to the associated class probabilities. During training a set of labeled training points $\mathcal{S}_0$ is used to optimize the parameters of the tree. In a classification tree the entropy of the class distributions associated with different nodes decreases (the confidence increases) when going from the root towards the leaves. Note the gray-ish color of the root node and the more distinct colors of the leaves

- **Goal of classification**: automatically associate an input data point $\mathbf{v}$ with a discrete class $c \in \{c_k\}$
- **Properties of classification forests:**
  - naturally handle problems with more than two classes
  - provide probabilistic output
  - can generalize well to unseen data
  - efficient due to a small set of tests applied to each data point
  - ease of parallel implementation
- **Problem to be solved:** Given a labeled set of training data, learn a general mapping which associates previously unseen test data with their corresponding classes
- each training point is a pair $(\mathbf{v}, c)$, where $\mathbf{v} = (x_1, \ldots, x_d) \in \mathbb{R}^d$ and we want to find the distribution $p(c|\mathbf{v})$ by optimizing an energy over a training set $\mathcal{S}_0$ of data and labels.
- **Forest training**: optimize the parameters of the weak learner at each split/test node j:

$$\boldsymbol{\theta}_j = \arg\max_{\boldsymbol{\theta} \in \mathcal{T}_j} I(\mathcal{S}_j, \boldsymbol{\theta})$$

The **objective function** I:

$$I(\mathcal{S}_j, \boldsymbol{\theta}) = H(\mathcal{S}_j) - \sum_{i \in [L,R]} \frac{|\mathcal{S}_j^i|}{|\mathcal{S}_j|} H(|\mathcal{S}_j^i)$$

where i indexes the child nodes. The two child sets are a function of the parent set and split parameters $\boldsymbol{\theta}$

https://link.springer.com/book/10.1007/978-1-4471-4929-3

- The **entropy** for a set $\mathcal{S}$ of training points is

$$H(\mathcal{S}) = -\sum_{c \in \mathcal{C}} p(c) \log p(c)$$

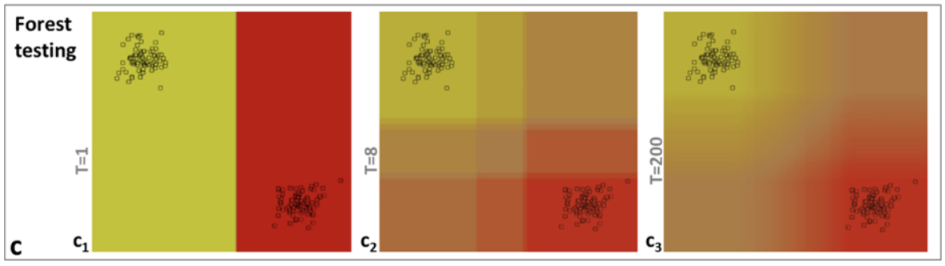
where $p(c)$ is calculated as the normalized empirical histogram of labels corresponding to the training points in $\mathcal{S}$

- training by maximizing information gain tends to produce trees where the entropy of the class distributions associated with the nodes decreases (ie. prediction confidence increases) from root to leaves (FIG 4.1B)
- **Randomness** can be injected by randomized node optimization (before training node j, randomly sampled $\rho = 1000$ parameter values from billions of possibilities, then information gain maximization through exhaustive search on reduced set of possibilities)
- **Leaf and Ensemble Predicition Models** during testing each tree leaf yields the posterior $p_t(c|\mathbf{v})$ and the forest output is defined:

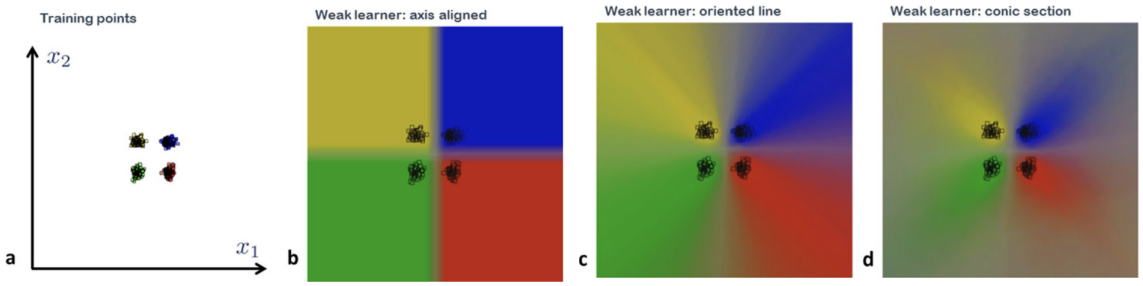
$$p(c|\mathbf{v}) = \frac{1}{T} \sum_{t=1}^{T} p_t(c|\mathbf{v})$$

1.1 Effect of Model Parameters

- **Forest size**: in testing, increasing the forest size T produces smoother class posteriors. Each individual tree produces overconfident predictions (notice sharp boundary in c_1) and increasing T results in higher confidence near the training points/lower confidence away.



- **Multiple classes and training noise**: An increase in training noise results in a larger overall uncertainty in the testing posterior
- **Tree depth**: As tree depth D increases, the overall prediction confidence also increases. Changing D can control overfitting: too large-overfit; too short- low confidence posteriors
- **Weak learner**: choice of weak learner is context dependent. For example, axis aligned are easy to compute but have blocky artifacts bad for generalization



- **Randomness**: Reducing ρ results in increasing randomness of each tree and reduces their correlation. Larger randomness reduces the blocky artifacts in axis aligned weak learners. Larger randomness yields much lower overall confidence.

https://link.springer.com/book/10.1007/978-1-4471-4929-3

1.2 Maximum Margin Classification with Forests

- SVM's are popular due to their ability to separate data via a margin-maximizing surface which yields good generalization.
- For a decision tree given a linearly separable 2-class training data set, any separating line that fully separates the classes results in maximum information gain, but a line placed right in the middle of the gap is the maximum margin solution.
- Consider this situation where we constrain weak learners to be vertical lines only:

$$h(\mathbf{v}, \boldsymbol{\theta}_j) = [\phi(\mathbf{v}) > \tau] \text{ with } \phi(\mathbf{v}) = x_1$$

The gap Δ is then

$$\Delta = x_1'' - x_1'$$

where x_1' and x_1'' correspond to the first feature of the two support vectors (see below).

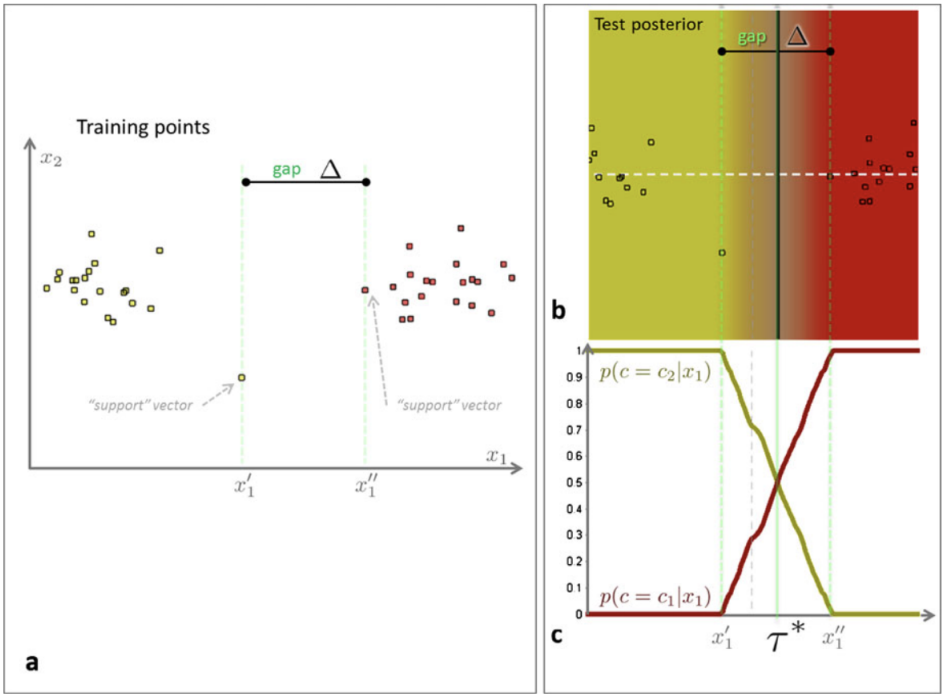


Fig. 4.9 Forest's maximum margin properties. (**a**) Input 2-class training points. They are separated by a gap of dimension Δ. (**b**) Forest posterior. Note that all of the uncertainty band resides within the gap. (**c**) Cross-sections of class posteriors along the horizontal, white dashed line in (**b**). Within the gap the class posteriors are linear functions of x_1. Since they have to sum to 1 they meet right in the middle of the gap. In these experiments we use $\rho = 500$, $D = 2$, $T = 500$ and axis aligned weak learners

- The optimal separating line is at position τ^*

$$\tau^* = \arg\min |p(c = c_1|x_1 = \tau) - p(c = c_2|x_1 = \tau)|$$

assume that when training a node its available test parameters (τ) are sampled from a uniform distribution, then the forest posteriors behave linearly within the gap region:

$$\lim_{\rho \to |\mathcal{T}|, T \to \infty} p(c = c_1|x_1) = \frac{x_1 - x_1'}{\Delta}, \forall x_1 \in [x_1', x_1'']$$

https://link.springer.com/book/10.1007/978-1-4471-4929-3

Consequently, since $\sum_{c\in\{c_1,c_2\}} p(c|x_1) = 1$, we have

$$\lim_{\rho\to|\mathcal{T}|,T\to\infty} \tau^* = x_1' + \frac{\Delta}{2}$$

ie. the optimal separation is placed in the middle of the gap

- Each individual tree is not guaranteed to produce maximum margin separation; however, the combination of multiple trees at the limit $T \to \infty$ produces max margin behavior
- **Effect of randomness on optimal separation**: When using more randomness (smaller ρ) individual trees are not guaranteed to split data perfectly, and may yield a sub-optimal information gain and lower confidence in the posterior.

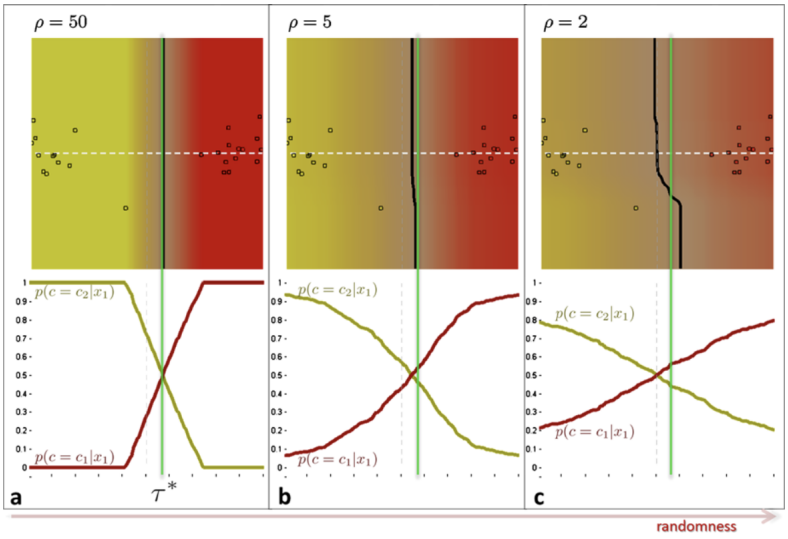


Fig. 4.10 The effect of randomness on the forest margin. (**a**) Forest posterior for $\rho = 50$ (small randomness). (**b**) Forest posterior for $\rho = 5$. (**c**) Forest posterior for $\rho = 2$ (highest randomness). These experiments have used $D = 2, T = 400$ and axis-aligned weak learners. The *bottom row* shows 1D posteriors computed along the white dashed line. Increasing randomness produces less well defined separating surfaces. The optimal separating surface, *i.e.* the loci of points where the class posteriors are equal (shown in *black*) moves towards the left of the margin-maximizing line (shown in *green* in all three experiments). As randomness increases individual training points have less influence on the separating surface

- **Influence of the weak learner model**: using linear weak learners still generally produces globally nonlinear classification due to the fact that multiple linear split nodes are organized hiearchically.
- **Effect of randomness model**: In bagging, randomness is injected by randomly sampling different subsets of training data so each tree sees a different training subset. specific support vectors may not be avialble in some of the trees and the posterior associated with those trees will tend to move the optimal separating surface away from the max margin one. In randomized node optimization, all available trainning data is used and enables us to control the maximum margin behavior by simply changing ρ

https://link.springer.com/book/10.1007/978-1-4471-4929-3

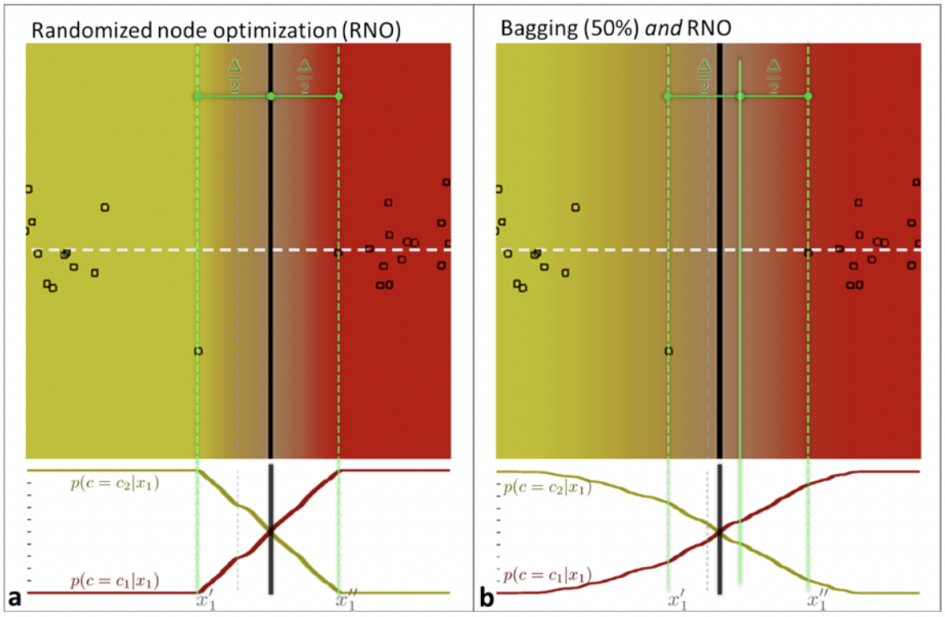


Fig. 4.13 Max-margin: bagging v randomized node optimization. (**a**) Posterior for forest trained with randomized node optimization. (**b**) Posterior for forest trained with bagging. In bagging, for each tree we use 50 % random selection of training data with replacement. Loci of optimal separation are shown as *black lines*. In these experiments we use $\rho = 500$, $D = 2$, $T = 400$ and axis-aligned weak learners. Areas of high entropy are shown in *gray* to highlight the separating surfaces