Classification forests

- goal : automatically associate an input data point $\underline{v}$ with a discrete class $c \in \{c_k\}$

- Interesting properties of classification forests.
  - naturally handle multi-class problems
  - provide a probabilistic o/p.
  - can generalize
  - efficient (parallel implementation)
  - margin-maximization behavior
  - quality of posterior / confidence can be controlled via tree params.

- Classification Task
  - Given a labeled set of training data, learn a general mapping which associates previously unseen test data with their corresponding classes.

* training point ⇒ (v, c)
* we wish to compute the posterior distribution $p(c|v)$

Objective Function

$$\theta_j^* = \underset{\theta \in \mathcal{T}_j}{\arg\max}\ I(S_j, \theta)$$

where θ_j are the parameters of the weak learner at split node j.

$$I(S_j, \theta) = H(S_j) - \sum_{i \in \{L, R\}} \frac{|S_j^i|}{|S_j|} H(S_j^i)$$

classical information gain.

$$H(S) = -\sum_{c \in C} p(c) \log[p(c)]$$ ← entropy

computed as the normalized empirical histogram of labels in S.

maximizing $I(S_j, \theta)$ produces trees where the entropy of nodal class distribution ↓ when going from root → leaves. certainty ↑

Class Re-balancing

* Class imbalance can have a detrimental effect on forest training.
  1. resampling training data
  2. class weighting by its inverse frequency computed from the prior distribution

} can reduce imbalance issues.

Randomness

* Random node optimization

$\rho = |T_j| \rightarrow$ controls amount of randomness.

$\uparrow$ a subset of features ($T_j \subseteq T$)

* optimization is done using this reduced set of features.

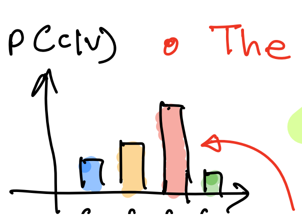
The Leaf & Ensemble Prediction Models

* Classification forests produce an entire class distribution instead of a single class point prediction.

* During testing $\Rightarrow$
  - input test data point is given to root.
  - at each node a test is applied and data point is sent to the appropriate node. (repeated until a leaf node is reached)
  - At the leaf node, the stored posterior $p_t(c|v)$ is read off.

of the t^{th} tree

these are computed from the leaf statistics when the training data are partitioned across the leaves.



- The forest posterior is computed as,

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

c_1 c_2 c_3 c_4 predicted class

Effect of Model Parameters

Effect of Forest Size

* Forest size (# trees) ↑ ⇒ smoothness of the posterior ↑

higher confidence near training points.
lower confidence away from training region

generalization behavior

* Quality of uncertainty is key for determining the inductive generalization away from training data (confidence in regions far from training data)

Multiple Classes & Training Noise

* unlike SVMs, same forest model can handle both binary & multi-class classification problems.

* with larger training noise ⇒ classification uncertainty ↑

Effect of Tree Depth

* Tree depth ↑ ⇒ prediction confidence ↑

But this could result in overfitting

* Too shallow trees produce low confidence posteriors.

* Having multiple trees and an optimal tree depth can alleviate the overfitting problem.

function of tree depth

Effect of the weak learner

* The choice of the weak learner depends on the application at hand

Effect of Randomness (ρ-value)

* increasing randomness by setting $\rho = |\mathcal{T}_j| < |\mathcal{T}|$ reduces correlation between the trees in the forest.
* Larger randomness $\rightarrow$ well rounded decision boundaries with much lower overall confidence

Maximum Margin Classification with Forests