Ensemble error:

$$\epsilon_{ens} = \sum_{k}^{n} \binom{n}{k} \epsilon^k (1-\epsilon)^{n-k}$$

"Soft" Voting

$$\hat{y} = \arg\max_j \sum_{i=1}^{n} w_i p_{i,j}$$

$p_{i,j}$: predicted class membership probability of the *i*th classifier for class label *j*

w_j : optional weighting parameter, default $w_i = 1/n, \forall w_i \in \{w_1, \ldots, w_n\}$

$$\frac{k(k-1)}{2} = classifiers(1v1)$$

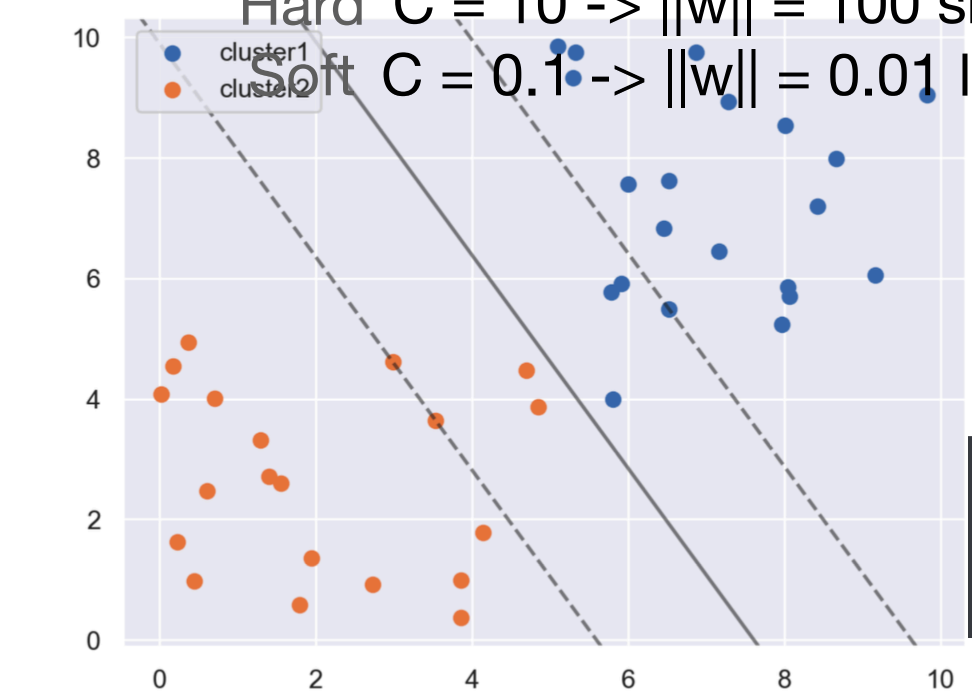
Bootstrap Sampling

internal unbiased estimate (OOB) random forests

$$P(\textbf{not chosen}) = \left(1 - \frac{1}{n}\right)^n,$$

$$\frac{1}{e} \approx 0.368, \quad n \to \infty.$$

Solution:



Hard C = 10 -> ||w|| = 100 small margin

Soft C = 0.1 -> ||w|| = 0.01 large margin

$$M = \frac{2}{\sqrt{\mathbf{w}^T\mathbf{w}}}$$

$$d = \frac{|w^T x|}{\|w\|}$$

Training accuracy: Less than 100%. (because soft-margin allows some slack/error)
Since the margins are soft (i.e not strictly enforced, allow some misclassification mistakes to occur), soft SVM's will generalize well to unseen data.

SVM non separable, convex function, max
M = regularize

$$\text{Minimize } \mathcal{L}(\mathbf{w}, b) = \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{n}(1 - y_i(\mathbf{w}^T\mathbf{x}_i + b))_+$$

$$W_{n+1} = W_n - \alpha\,(df(w)\,/\,dw)$$

$$\frac{\mathcal{L}(\mathbf{w},b)}{\partial \mathbf{w}} = \mathbf{w} + C\sum_{i=1}^{n} \begin{cases} 0 & if\ \ y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 \\ -y_i\mathbf{x}_i & otherwise \end{cases}$$

$$\frac{\mathcal{L}(\mathbf{w},b)}{\partial b} = C\sum_{i=1}^{n} \begin{cases} 0 & if\ \ y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 \\ -y_i & otherwise \end{cases}$$

svm and logit, differ in training but same in testing

Fit of the model B/V
K = N; Low V High B
5x20 Repeat K = High V Low B

Fit of the hold-out-test-set-perf
K = N; High V Low B
5x20 Repeat K = Low V High B

Hard SVMs tend to overfit

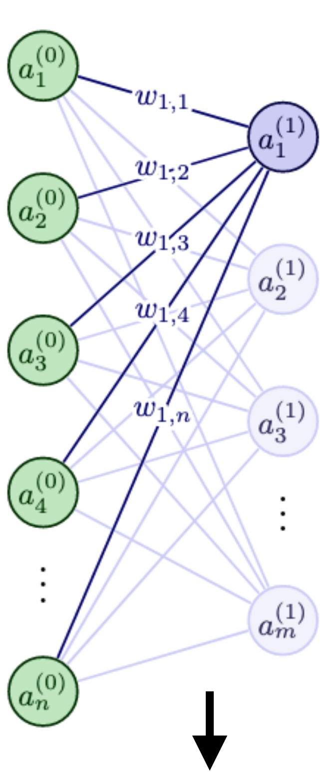
Primal version of classifier:

$$f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b$$

Dual version of classifier:

$$f(\mathbf{x}) = \sum_{i}^{N} \alpha_i y_i (\mathbf{x}_i{}^\top \mathbf{x}) + b$$

many alpha are 0, allows for kernel trick aka distance/similarity



$$a_1^{(1)} = \sigma\left(w_{1,0}0 + w_{1,1}1 + \ldots + w_{1,n}n + b_1^{(0)}\right) = \sigma\left(\sum_{i=1}^{n} w_{1,i}i + b_1^{(0)}\right)$$

$$\begin{aligned} \text{Err}(x_0) &= E[(Y - \hat{f}(x_0))^2 | X = x_0] \\ &= \sigma_\varepsilon^2 + [\text{E}\hat{f}(x_0) - f(x_0)]^2 + E[\hat{f}(x_0) - \text{E}\hat{f}(x_0)]^2 \\ &= \sigma_\varepsilon^2 + \text{Bias}^2(\hat{f}(x_0)) + \text{Var}(\hat{f}(x_0)) \\ &= \text{Irreducible Error} + \text{Bias}^2 + \text{Variance}. \end{aligned}$$

Decision trees

Advantages

- Interpretable
- Non-parametric method
- Able to fit arbitrary decision boundaries (not just linear!)
- Don't need to scale features to match each other
- Can be combined with techniques to make it better (like bagging and boosting)

Disadvantages

- Easy to overfit
- Needs some kind of pruning and tree-growth limits to avoid overfitting
- For regression trees, the output is bounded by the limits of the training samples

Boosting better than other ensemble
Higher accuracy: focusing more on the obv difficult to class, reduce noise/outliers, increase acc
More efficient: Each base learner is trained on a subset o the training data, and the weights of the training data are adjusted based on the errors of the previous base learners.
Robustness: Boosting can be more robust than other ensemble methods to changes in the data (like dist.), focuses more on difficult-to-classify obv (tend to be more stable across different data dist.)
Flexibility: Can be applied to range of ML prob (classification, regression, & ranking). Can combined with other ML techn, like feature selection & model interpretation

The radius basis function (RBF) takes inspiration from the normal distribution formula as following:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

Ensure we have a normal bell curve shape to convert distances to similarities

$$f(x, l, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-l}{\sigma}\right)^2}$$

Vanish Grad = small gradient, slow updates, slow to converge
Exploding Grad = large gradient, large steps, divergence