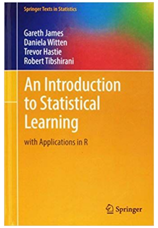
SVMs, Ensembles & Neural Networks

Contents

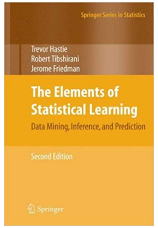
- Support Vector Machines
- Ensemble Learning
- Neural Networks

Readings



Section 8.2 (Ensemble learning)

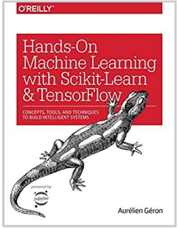
Section 9 (SVM)



Section 11 (Neural Networks)

Section 12 (SVM)

Sections 15, 16 (Ensemble learning)



SVM, p. 140

Ensemble learning, p. 175

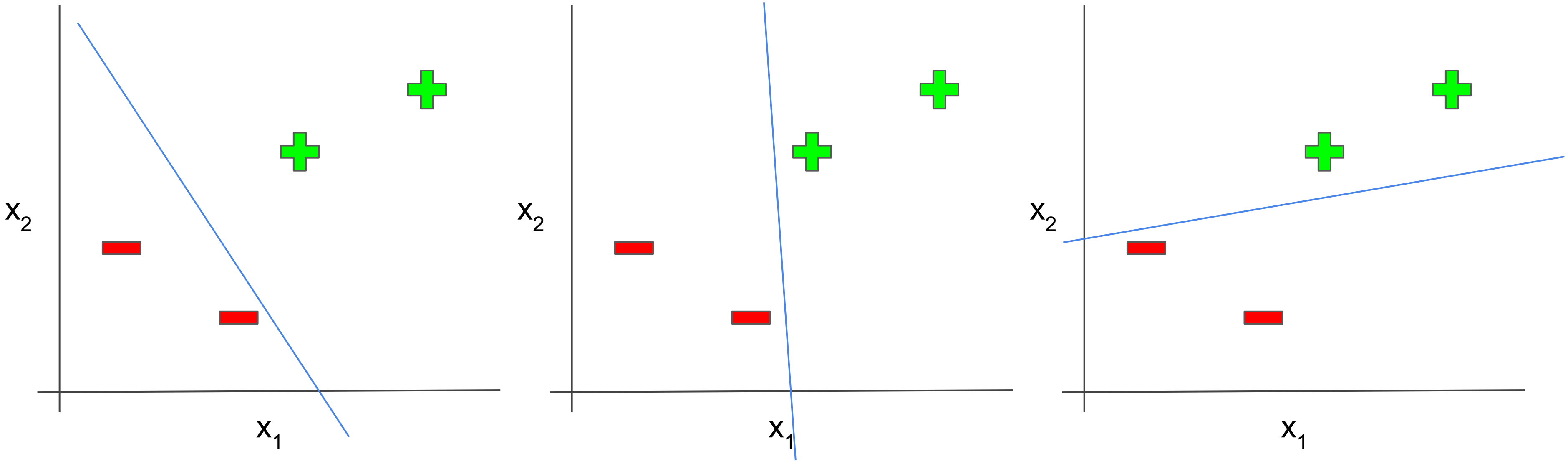
Neural Networks, p. 244

Support Vector Machines

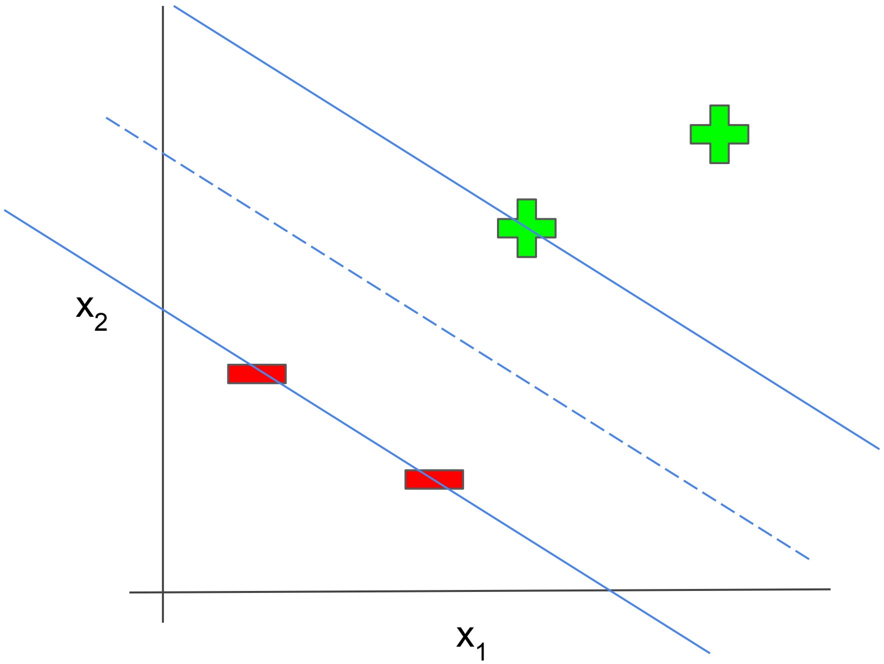
- Theoretically beautiful,
- Very powerful,
- Versatile (classification, regression, outlier detection) …

… machine learning model.

Separating hyperplanes



Maximal Margin Classifier



Ensemble Learning

- Bagging
- Random Forests
- Boosting

Bagging

- Reduces variance, increases accuracy
- Trained on bootstrapped samples
- Average the prediction of hundreds of trees into a single prediction; trees grow deep, remain unprunned
- Lacks interpretability of the model, but does not overfit

Random Forests

- Bagging produces correlated trees
- Random forests only allow a small subset of predictors in each split, preventing strong predictors to emerge in roots, making trees correlated
- Lacks interpretability of the model, but does not overfit

Boosting

- Uses the entire training set, not bootstrapped samples
- Fitting residuals instead of outcomes
- Lacks interpretability, may **overfit** for large number of trees

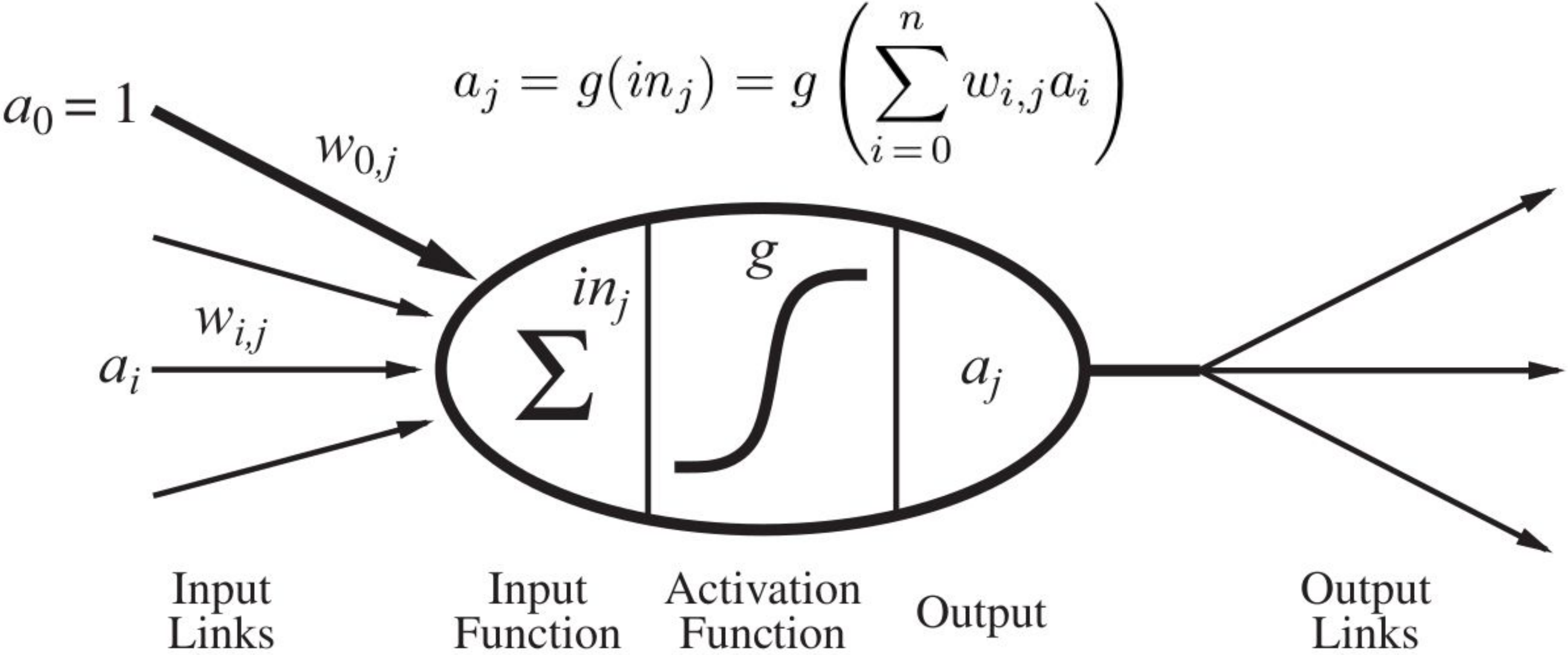
1. Set $\hat{f}(x) = 0$ and $r_i = y_i$ for all i in the training set.
2. For $b = 1, 2, \ldots, B$, repeat:
   (a) Fit a tree $\hat{f}^b$ with d splits ($d+1$ terminal nodes) to the training data (X, r).
   (b) Update $\hat{f}$ by adding in a shrunken version of the new tree:
   $$\hat{f}(x) \leftarrow \hat{f}(x) + \lambda \hat{f}^b(x).$$
   (c) Update the residuals,
   $$r_i \leftarrow r_i - \lambda \hat{f}^b(x_i).$$
3. Output the boosted model,
$$\hat{f}(x) = \sum_{b=1}^{B} \lambda \hat{f}^b(x).$$

Neural Networks

- Biologically inspired (McCulloch & Pitts, 1943)
- Extremely powerful
- Extremely versatile …

… machine learning model.

A Neuron


$a_j = g(in_j) = g\left(\sum_{i=0}^{n} w_{i,j} a_i\right)$
$a_0 = 1$
$w_{0,j}$
$w_{i,j}$
a_i
in_j
Σ
g
a_j
Input Links
Input Function
Activation Function
Output
Output Links

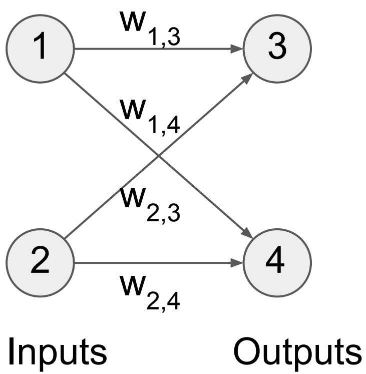
Neural Networks

- Feed-forward:
  Connections in one direction, from inputs to outputs.
  No loops; directed acyclic graph.

- Recurrent:
  Connects its own outputs to its own inputs.
  Have short-term memory.

Perceptron

- Single-layer neural network
- All inputs connected directly to outputs



Multilayer feed-forward neural network

- Learning by back-propagation