Lecture 9

IDS575: Statistical Models and Methods
Theja Tulabandhula

Notes derived from the book titled "Elements of Statistical Learning [2nd edition] (Section 9.4, Chapters 12 and 15)

We continue our foray into more supervised learning methods, viz., Random Forests, Multivariate Adaptive Regression Splines and Support Vector Machines.

1 Random Forests

The Random Forests method improves on *bagging* by reducing the correlation between the sampled trees. Below is a brief description of bagging.

1.1 Bagging

Bagging (bootstrap aggregation) improves the performance of a classifier through averaging predictions across bootstrapped models. For each bootstrap sample $Z^{*b}, b = 1, ..., B$, we fit our model giving prediction $\widehat{\mathrm{f}}^{*b}(x)$ at some new point x. The bagging prediction is:

$$\widehat{\mathrm{f}}_{\mathrm{bag}}(x) = \frac{1}{B}\sum_{b=1}^{B}\widehat{\mathrm{f}}^{*b}(x).$$

Note 1. Each bootstrap tree could involve different features than the original and may have different number of terminal nodes.

Note 2. Bagged estimates for classification cannot be used as estimates of the true conditional distribution of the class given input.

In regression with squared loss, bagging helps smooth out the high variance in predictions, especially when inputs are highly correlated.

1.2 Details of the Random Forests (RF) Method

The key idea in RF is to build a large collection of *de-correlated* trees and then taking averages.

The shortcoming of bagging is that the expectation of the average of B trees is the same as any single tree. This means that error can only be decreased via variance reduction. But since each of the trees are built using the same data, they are correlated with each other.

Example 1. If B i.i.d. random variables are averaged the variance of the average is $\frac{1}{B}\sigma^2$. On the other hand, if they are pairwise correlated, then the variance of the average is $\rho\sigma^2 + \frac{1-\rho}{B}\sigma^2$. So as B increases there is no further reduction in the first term.

RF tries to improve on bagging by reducing the correlation between trees without increasing the variance too much.

The RF algorithm is shown in Figure 1. The key feature is that before each split, we select $m \leq p$ of the input variables randomly as candidates for splitting, while building a tree with each bootstrapped sample. Smaller m reduces correlation across trees.

Algorithm 15.1 *Random Forest for Regression or Classification.*

1. For $b = 1$ to B:
   (a) Draw a bootstrap sample $\mathbf{Z}^*$ of size N from the training data.
   (b) Grow a random-forest tree T_b to the bootstrapped data, by recursively repeating the following steps for each terminal node of the tree, until the minimum node size n_{min} is reached.
      i. Select m variables at random from the p variables.
      ii. Pick the best variable/split-point among the m.
      iii. Split the node into two daughter nodes.
2. Output the ensemble of trees $\{T_b\}_1^B$.

To make a prediction at a new point x:

Regression: $\hat{f}_{\text{rf}}^B(x) = \frac{1}{B}\sum_{b=1}^B T_b(x)$.

Classification: Let $\hat{C}_b(x)$ be the class prediction of the bth random-forest tree. Then $\hat{C}_{\text{rf}}^B(x) = \textit{majority vote}\ \{\hat{C}_b(x)\}_1^B$.

Figure 1: The RF algorithm.

Example 2. In Figure 2, RF is compared to GBM with shrinkage[1] for the California housing dataset. From the plot, we can see that RF stabilizes with about 200 trees whereas boosting continues to improve as more trees are added.

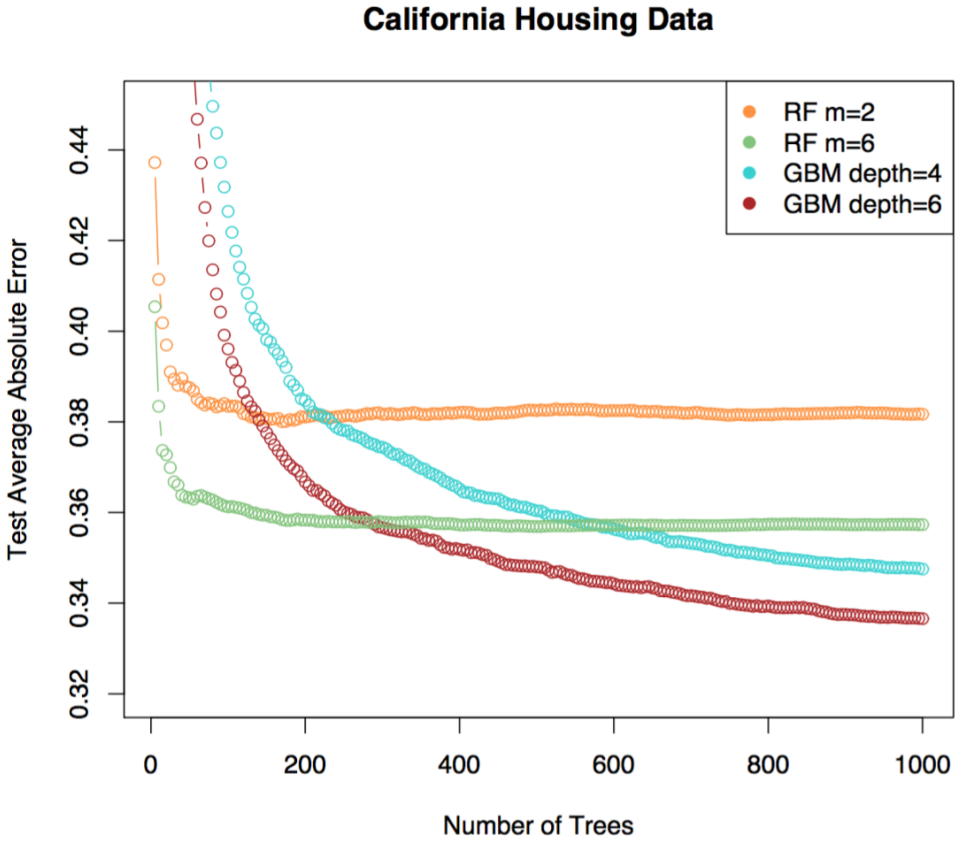

Figure 2: The RF algorithm performance on a housing dataset.

Note 3. RF takes an average for regression, and it takes the majority vote for classification.

Note 4. Out-of-Bag (OOB) samples are used to compute validation performance: for each (x_i, y_i), only those trees that did not use this data are averaged. This can be used in lieu of K-fold cross validation. OOB samples can be used to estimate variable important estimates as well, although we will not discuss this further here.

1.3 Interpretation

We will look at a couple of ways to get interpretability with this class of models. First is via relative importance of variables, and the second is via partial dependence plots.

[1]Shrinkage is an additional parameter that scales down the contribution of each tree while being added to the sum.

The idea here is to isolate those input variables that are the most relevant to the prediction task.

For a single tree, let the measure of (squared) relevance for variable X_l be denoted by $I_l^2(T) = \sum_{t=1}^{J-1} \hat{\imath}_t^2 1[v(t) = l]$. Here, the sum is over the $J-1$ internal nodes. We are checking if the variable $v(t)$ used to split the node was l or not, and if it is, we are adding the improvements denoted as $\hat{\imath}_t^2$. While constructing the tree, the variable that is chose gives the maximal estimated improvement $\hat{\imath}_t^2$ in squared error over that for a constant over the entire region.

For boosted tree models, the formula for importance is simply $I_l^2 = \frac{1}{M} \sum_{m=1}^{M} I_l^2(T_m)$.

Note 5. Since these measures are relative, one can scale the largest variable's score to 100 and rescale others respectively.

Note 6. For classification, this is very similar, and we add up the K trees for each m.

Example 3. Figure 3 shows the relative importance of certain variables in a prediction task involving California housing dataset.

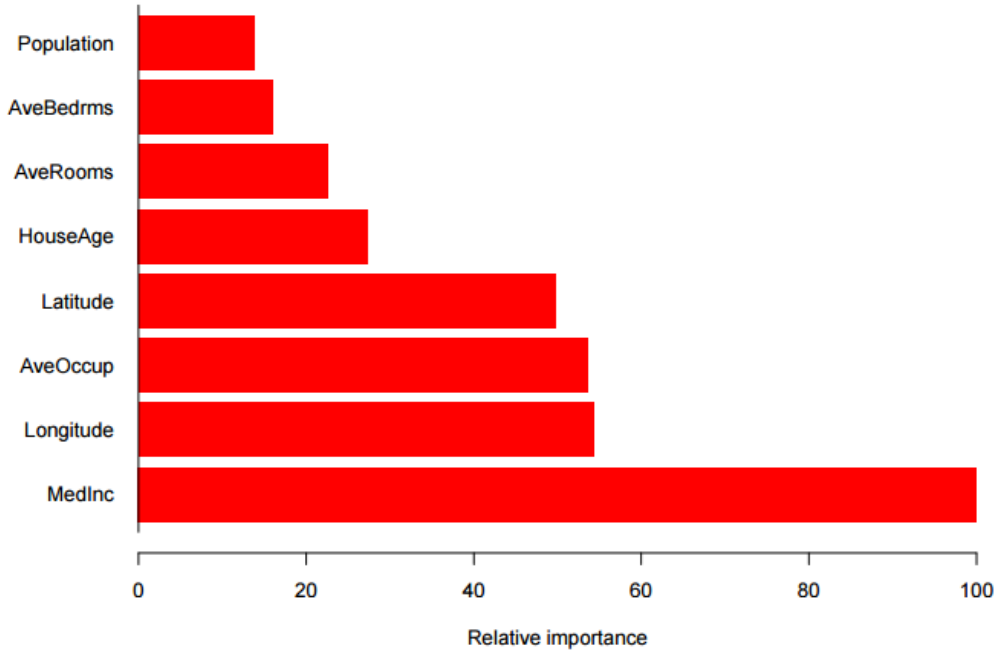

Figure 3: Example relative importance plot.

Partial dependence plots help understand how $f(X)$ depends on some of the input variables (up to 2 variables). Let X_S denote a subset of $l < p$ variables (and X_C its complement). Let $f(X) = f(X_S, X_C)$. Then, the *partial dependence* of $f(X)$ on X_S is given as:

$$f_S(X_S) = E_{X_C} f(X).$$

They can be estimated using data as: $f_S(X_S) = \frac{1}{N}\sum_{i=1}^{N} f(X_S, x_{iC})$. This is true for both regression and classification.

2 Multivariate Adaptive Regression Splines (MARS)

MARS gives us a regression model that is composed of 1-dimensional basis functions defined below:

$$(x-t)_+ = \begin{cases} x-t & \text{if } x > t \\ 0 & \text{if } x \leq t \end{cases}, \text{ and}$$

$$(t-x)_+ = \begin{cases} t-x & \text{if } x < t0 \\ 0 & \text{if } x \geq t \end{cases}.$$

Here, t is called a *knot*.

Example 4. An example plot of these 1-dimensional functions is shown in Figure 4.

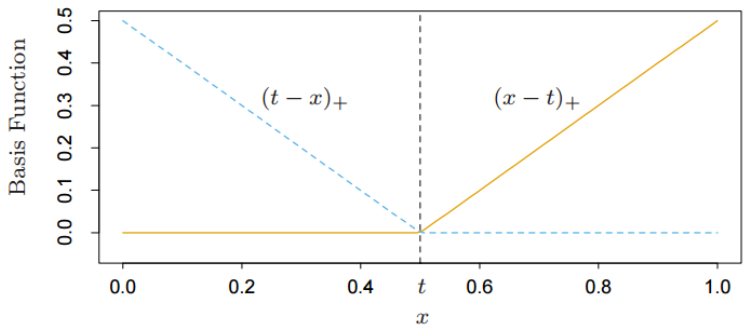

Figure 4: The basis functions (for $t = 0.5$).

The idea in MARS is to use such pairs of basis functions, defined for each x_{ij} in the input data matrix $\mathbf{X}$ (lets call this set of functions $\mathcal{C}$).

Example 5. We will deonte the functions of the j^{th} coordinate and x_ij using $h_1(X) = (X_j - x_{ij})_+$ and $h_2(X) = (x_{ij} - X_j)_+$. We use generic notations $h_i(X)$ to think of these basis functions as functions of all coordinates notationally.

The MARS model will be of the form

$$f(X) = \beta_0 + \sum_{m=1}^{M} \beta_m h_m(X),$$

where each h_m would either be one of the basis functions above or a product of such basis functions. This model is built as follows:

- Say we have some functions $\mathcal{M}$ already chosen (say this number is M). At the beginning, assume $\mathcal{M}$ has the function $h_0(X) = 1$ (we find β_0 by minimizing RSS).
- To add a new function we do the following. For every function $h_l \in \mathcal{M}$ and every function[2] in $\mathcal{C}$, we consider adding that derived function:

$$\widehat{\beta}_{M+1}\, h_l(X) \cdot (X_j - t)_+ + \beta_{M+2} h_l(X) \cdot (t - X_j)_+,$$

  which gives the largest decrease in the RSS (all coefficients are re-estimated).
- Do this till we reach a maximum number of functions (a design choice). Then, if needed, delete some of these functions incrementally that show the lowest change in a cross-validation score[3].

Example 6. When $\mathcal{M}$ just has one function $h_0(X) = 1$, we consider adding functions of the form $\widehat{\beta}_1 \cdot 1 \cdot (X_j - t)_+ + \beta_2 \cdot 1 \cdot (t - X_j)_+$. There are Np such functions. We get the best one among these and add it to $\mathcal{M}$. Say it was the one corresponding to x_{72}. Then $\mathcal{M}$ has three functions now: $\{h_0(X), h_1(X) = (X_2 - x_{72})_+, h_2(X) = (x_{72} - X_2)_+\}$. Next stage, we may add functions that may for example be $h_3(X) = (X_1 - x_{51})_+ \cdot (x_{72} - X_2)_+$ etc.

Why these functions? When these functions get multiplied, they they tend to be non-zero in a small region of the input space. Thus they can model local aspects of the regression problem well.

Another interesting restriction for MARS is that higher order products can only come in as functions if the lower order functions are already in $\mathcal{M}$.

3 Support Vector Machines

The big idea with this class of methods is to produce non-linear decision boundaries by constructing a linear boundary in a large transformed version of the input space.

Lets look at the support vector classifier first (we'll bring in the term *machine* a bit later).

As usual, let $\{x_i, g_i\}_{i=1}^n$ be our training data. Let $g_i \in \{-1, 1\}$. Let a hyperplane be defined as $\{x : f(x) = x^T\beta + \beta_0 = 0\}$.

(Linear) SV classifier is defined as: $G(x) = \text{sign}(f(x))$ (basically the hyperplane splits the two classes by a linear decision boundary).

[2] We avoid using the function involving x_{ij} if h_l already involves x_{ij}.

[3] Actually, there is an estimate called the Generalized Cross Validation (GCV) estimate that is used here, very similar to the least squared loss estimate.

Note 7. If the classes are separable, then it is possible to find a $f(x)$ such that $y_i f(x_i) > 0$ for all $i = 1, .., N$. In fact there may be many such classifiers.

Which classifier should we choose: we will choose the classifier that maximizes the geometric *margin* M between the training data for class -1 and 1. The optimization problem written below captures this:

$$\max_{\beta,\beta_0,M} 2M \text{ such that}$$
$$y_i \left(x_i^T \frac{\beta}{\|\beta\|_2} + \frac{\beta_0}{\|\beta\|_2} \right) \geq M, i = 1, ..., N.$$

See the left panel of Figure 5. The band/margin is M units from points on either side and is 2M wide.

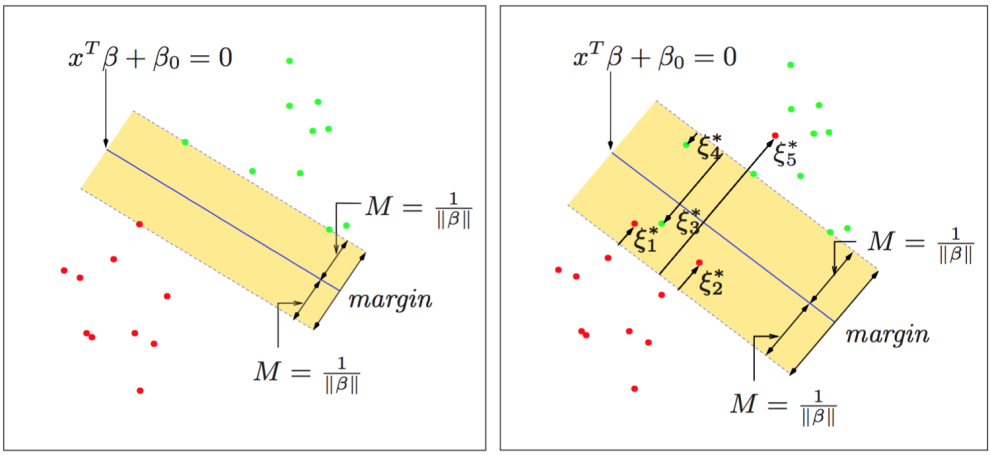

Figure 5: Linear support vector classifier. Left: separable instance. Decision boundary is shown as a solid line. We can choose margin $M = \frac{1}{\|\beta\|_2}$ without loss of generality. Right: the non-separable case, with ξ_i^* labeled points are on the wrong side of the margin by a multiplicative factor of the margin.

We can set $M = \frac{1}{\|\beta\|_2}$ without any loss of generality because margin depends on the scale of the hyperplane normal vector β, which itself does not affect classification accuracy. Thus we get:

$$\max_{\beta,\beta_0} \frac{2}{\|\beta\|_2} \text{ such that}$$
$$y_i(x_i^T\beta + \beta_0) \geq 1, i = 1, ..., N,$$

which can be equivalently written as[4]:

$$\min_{\beta,\beta_0} \frac{1}{2}\|\beta\|_2^2 \text{ such that}$$
$$y_i(x_i^T\beta + \beta_0) \geq 1, i = 1, ..., N.$$

Note 8. The above formulation is a convex optimization problem.

When classes overlap, we allow for some training observations to be on the wrong size of the margin. Lets define *slack* variables ξ_i and modify the above problem to:

$$\min_{\beta,\beta_0,\xi_i} \frac{1}{2}\|\beta\|_2^2 + C\sum_{i=1}^{N}\xi_i \text{ such that}$$
$$y_i(x_i^T\beta + \beta_0) \geq 1 - \xi_i, i = 1, ..., N,$$
$$\xi_i \geq 0, i = 1, ..., N.$$

One key observation from the above formulation is that training data observations that are far away from the boundary and in their correct class do not affect the choice of the boundary.

4 Summary

We learned the following things:

- The Random Forests method.
- A model for regression called MARS.
- A model for classification based on the idea of margins called Support Vector Machine.

A Sample Exam Questions

1. Compare and contrast Random Forests with Bagging.
2. What are the key characteristics of a MARS model?
3. What is the idea of margins?

[4]Notice the square!

B Proportional Decrease in Model Error (R^2)

R^2 is defined as:

$$R^2 = \frac{\text{MSE}_0 - \text{MSE}}{\text{MSE}_0},$$

where $\text{MSE}_0 = \text{ave}_{x \in \text{Test}}(\bar{y} - y^{true})^2$ and $.rmMSE = \text{ave}_{x \in \text{Test}}(\widehat{\text{f}}(x) - y^{true})^2$