Regression costs for decision trees

John Alan McDonald

2018-01-23

The purpose of this document is work thru an alternative to L_2 cost that is a bit more efficient to compute, and gives the same results when choosing split predicates in decision tree growing.

1 Greedy decision trees

A general binary decision tree consists of

- internal *split* nodes, each containing a predicate that determines whether a record goes to the left or right child of that node.
- terminal *leaf* nodes, each containing a leaf model function whose value is the tree's prediction for any record that ends up in that node.

Greedy split optimization — choose the best out of all feasible splits, and repeat on the resulting child nodes until there are no feasible splits — is the most common way of growing decision trees. It depends on several things:

1. A cost function c used to define 'best'.
2. An enumeration of splits to consider. Pure greedy splitting considers all 'feasible' splits on all attributes.
   (a) For categorical attributes, that, in general, means considering every partition of the categories into 2 subsets. However, for some important cost functions (eg 2-class Gini, L_2), it can be shown that the optimal split can be found by

sorting the categories by the corresponding score function (eg the response mean for L_2 cost), and then considering only splits by score.

(b) For numerical attributes, the most general split would come from treating the distinct values like the categories of a categorical variable. However, no one does that, mostly because there are usually too many distinct values. Instead, only splits by $\leq$ vs $>$ one of the distinct values are considered.

3. A feasibility test that determines whether a given split on a attribute is allowed. The most common case here is to require both children of the split contain some minimum number of training records.

2 Cost functions for L_2 numerical regression

Let $\mathcal{T} = \{(y, \mathrm{x})\}$ be the training data in the node to be split. It is a set of pairs of predictor record x and ground truth response y, where $y \in \mathbb{R}$ for numerical regression. We are considering splits on some particular predictor field x_k, which might be numerical or categorical.

The cost function for L_2 regression is the sum of squared deviations from the mean:

$L_2(\mathcal{T}) = \sum_{y \in \mathcal{T}} (y - \bar{y}_{\mathcal{T}})^2$, where $\bar{y}_{\mathcal{T}} = \frac{1}{\#\mathcal{T}} \sum_{y \in \mathcal{T}} y$.

Note that computing this *accurately*, in an online fashion, for moderate $\#\mathcal{T}$, the number of records in $\mathcal{T}$, allowing for the updating/downdating needed for fast split optimization, requires some care.

However, a little bit of algebra will let us use a simpler alternative to get the same splits.

Any split partitions the training y-values $\mathcal{T} = \{y\}$ into left and right subsets: $\mathcal{T} = \mathcal{L} \uplus \mathcal{R}$. The split cost is:

$$
\begin{aligned}
c(\mathcal{L},\mathcal{R}) =& L_2(\mathcal{L}) + L_2(\mathcal{R}) \\
=& \sum_{y\in\mathcal{L}} (y-\bar{y}_{\mathcal{L}})^2 + \sum_{y\in\mathcal{R}} (y-\bar{y}_{\mathcal{R}})^2 \\
=& \sum_{y\in\mathcal{L}} \left[y^2 - 2\bar{y}_{\mathcal{L}}y + \bar{y}_{\mathcal{L}}^2\right] + \sum_{y\in\mathcal{R}} \left[y^2 - 2\bar{y}_{\mathcal{R}}y + \bar{y}_{\mathcal{R}}^2\right] \\
=& \sum_{\mathcal{L}\uplus\mathcal{R}} y^2 - \frac{\left(\sum_{\mathcal{L}} y\right)^2}{\#\mathcal{L}} - \frac{\left(\sum_{\mathcal{R}} y\right)^2}{\#\mathcal{R}}
\end{aligned}
$$

Since $\sum_{\mathcal{L}\uplus\mathcal{R}} y^2$ doesn't depend on the split, minimizing $c(\mathcal{L},\mathcal{R})$ is equivalent to minimizing $-\left[\frac{\left(\sum_{\mathcal{L}} y\right)^2}{\#\mathcal{L}} + \frac{\left(\sum_{\mathcal{R}} y\right)^2}{\#\mathcal{R}}\right]$, so we can use $\frac{-\left(\sum_{\mathcal{T}} y\right)^2}{\#\mathcal{T}}$ as our cost function in split optimization.

3 Cost functions for L_2 vector-valued regression

Let $\mathcal{T} = \{(\mathrm{y},\mathrm{x})\}$ be the training data in the node to be split. Here the ground truth response y is a vector, $\mathrm{y} \in \mathbb{R}^m$, rather than a single number.

The cost function for L_2 vector-valued regression is the sum of squared L_2 distances from the mean vector:

$$
\begin{aligned}
L_2(\mathcal{T}) =& \sum_{y\in\mathcal{T}} \|\mathrm{y} - \bar{\mathrm{y}}_{\mathcal{T}}\|_2^2 \\
=& \sum_{y\in\mathcal{T}} \sum_{i=0}^{m-1} (y_i - \bar{y}_{i\mathcal{T}})^2
\end{aligned}
$$

Following the same reasoning as in section 2, we get for a simpler cost:

$$
c(\mathcal{L},\mathcal{R}) = -\left[\frac{\sum_{i=0}^{m-1}\left(\sum_{\mathcal{L}} y_i\right)^2}{\#\mathcal{L}} + \frac{\sum_{i=0}^{m-1}\left(\sum_{\mathcal{R}} y_i\right)^2}{\#\mathcal{R}}\right]
$$

4 Optimizing categorical splits

Sorting by score: [1] [3] [6]

Approximations [5].

5 Typesetting

This document was typeset using MikTEX 2.9 [7] and TEXworks 0.6.1 [4] on Windows 10. I used `arara` [2] to run `xelatex`, `biber`, `xelatex`, and `xelatex`. An alternative is to call these 4 commands by hand.

I believe only MikTEX and TEXworks are Windows specific; the actual typesetting tools should be usable on Linux and MacOS as well.

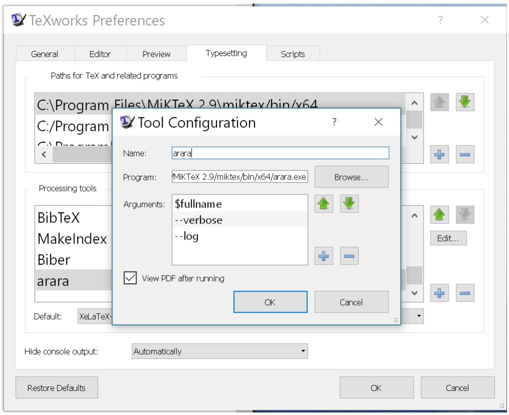

Figure 5.1: Configuring TEXworks for `arara`.

References

[1] Leo Breiman, Jerome H. Friedman, Richard Olshen, and Charles J. Stone. *Classification and Regression Trees*. Wadsworth, 1984.

[2] Paulo Roberto Massa Cereda, Marco Daniel, Brent Longborough, and Nicola Louise Cecilia Talbot. *arara: The cool TEX automation tool*. URL: https://github.com/cereda/arara (visited on 2017-03-13).

[3] Walter D. Fisher. "On grouping for maximum homogeneity". In: *Journal of the American Statistical Association* 53.294 (1958), pages 789–798.

[4] Jonathan Kew and Stefan Löffler.
TeXworks: A simple interface for working with TeX documents.
url: https://github.com/texworks/ (visited on 2017-03-13).

[5] Wei-Yin Loh and Nunta Vanichsetakul.
"Tree structured classification via generalized discriminant analysis".
In: *Journal of the American Statistical Association* 83 (1988), pages 715–728.

[6] Brian D. Ripley.
Pattern Recognition and Neural Networks.
Cambridge University Press, 1996.

[7] Christian Schenk.
MikTeX: typsetting beautiful documents.
url: https://miktex.org/ (visited on 2017-03-13).