Flat predictive models

John Alan McDonald

draft of 2018-02-13

Taiga's implementation of linear models is expressed in a way that is different from the treatment in most statistics texts. I happen to think that Taiga's approach is simpler, and better, particularly because it allows us to discuss these models as special cases of a general approach to predictive models. Whether you agree with that or not, you will need to understand Taiga's point of view if you want to use it successfullly.

Flat means linear or affine.

1 Linear spaces and functions

A *real linear space* (commonly refered to as a *real vector space*) $\mathbb{V}$ is a set of elements, called vectors, $\mathbf{v}_0$, $\mathbf{v}_1$, ...that is closed under *linear combinations*:

$$\mathbf{v} \;=\; a_0\mathbf{v}_0 \;+\; a_1\mathbf{v}_1 \in \mathbb{V} \tag{1.1}$$

for all $a_0, a_1 \in \mathbb{R}$. Note that $\mathbb{R}$ itself is a linear space under this definition.

(This can be generalized to other scalar fields, but $\mathbb{R}$ is sufficient for this discussion, and will be assumed in what follows.)

A *linear function* $\mathbf{f}$ from linear space $\mathbb{V}$ to linear space $\mathbb{W}$ preserves linear combinations:

$$\mathbf{f}(a_0\mathbf{v}_0 + a_1\mathbf{v}_1) \;=\; a_0\mathbf{f}(\mathbf{v}_0) \;+\; a_1\mathbf{f}(\mathbf{v}_1) \tag{1.2}$$

A *linear functional* is just a real-valued linear function: $\mathbf{f} : \mathbb{V} \to \mathbb{R}$.

The canonical linear space is $\mathbb{R}^n$, which we will take here to be the set of tuples of n real numbers. Implementations in Clojure/Java will most often approximate real tuples with instances of `double[n]`, often with instances of classes wrapping `double[n]`, and occasionally with `List<Number>`.

Note that the set of linear functions between two linear spaces, $\mathcal{L}(\mathbb{V}, \mathbb{W})$ is itself a linear space.

In fact, given any set of functions $\mathcal{F} = \mathsf{f} : \mathbb{D} \to \mathbb{V}$, from any domain $\mathbb{D}$ to a linear space $\mathbb{V}$, we get a linear space by closing those functions under linear combinations of their values: $\mathbb{F} = \{\mathsf{f}() = \sum_i a_i \mathsf{f}_\mathsf{i}() \ : \ \mathsf{f}_\mathsf{i} \in \mathcal{F}, a_i \in \mathbb{R}\}$

See Halmos [2] for thorough background.

2 Inner product space

An *inner product space* is a linear space together with an *inner product*, a symmetric, positive semidefinite, bilinear function of 2 arguments:

$$\begin{aligned} \mathrm{dot}(\mathsf{v}_0, \mathsf{v}_1) &= \mathrm{dot}(\mathsf{v}_1, \mathsf{v}_0) \\ \mathrm{dot}(\mathsf{v}, \mathsf{v}) &\geq 0 \\ \mathrm{dot}(\mathsf{v}, a_0\mathsf{v}_0 + a_1\mathsf{v}_1) &= a_0 \mathrm{dot}(\mathsf{v}, \mathsf{v}_0) + a_1 \mathrm{dot}(\mathsf{v}, \mathsf{v}_1) \end{aligned} \tag{2.1}$$

Skipping some details, we can identify the linear functionals on an inner product space with the vectors in that space:

$$\mathsf{v}^\dagger(\mathsf{w}) = \mathrm{dot}(\mathsf{v}, \mathsf{w}) \forall \mathsf{w} \in \mathbb{V} \tag{2.2}$$

L2 norm is $\|\mathsf{v}\|_2 = \sqrt{(}\mathrm{dot}(\mathsf{v}, \mathsf{v})$

L2 distance is $\|\mathsf{v}_0 - \mathsf{v}_0\|_2$.

3 Affine spaces and functions

An *affine space* is similar to a linear space, only closed under *affine combinations* rather than linear. That is, $\mathbb{A}$ is a set of elements, called points, p_0, p_1, ...such that:

$$\mathsf{p} \ = \ a_0\mathsf{p}_0 \ + \ a_1\mathsf{p}_1 \in \mathbb{A} \forall \mathsf{p}_0, \mathsf{p}_1 \in \mathbb{A}; a_0, a_1 \in \mathbb{R}; a_0 + a_1 = 1 \tag{3.1}$$

for all $a_0, a_1 \in \mathbb{R}$. The key distinction is the constraint that $a_0 + a_1 = 1$.

Abstract differences$\rightarrow$ linear translation space.

Note that every linear space is automatically an affine space, but not the reverse.

A canonical example of an affine space that is not a linear space is a hyperplane in $\mathbb{R}^n$ that doesn't go thru the origin, such as the set of all vectors whose first coordinate is 1. (TODO: picture of such a hyperplance in $\mathbb{R}^2$).

Linearization via equivalence classes of homogeneous coordinates.

Implementations:

1. (Coordinate frame) Pick an origin. Identify elements with elements of $\mathbb{R}^n$. An affine function on $\mathbb{R}^m \mapsto \mathbb{R}^n$ is a linear function plus an 'intercept' term in $\mathbb{R}^n$.
2. (Linear Lifting) Identity elements with equivalence classes in $\mathbb{R}^{n+1}$. $[p_0, p_1, \ldots, p_n$ $[p_0/p_n, p_1/p_n, \ldots, 1]$. Every linear function $\mathbb{R}^{m+1} \mapsto \mathbb{R}^{n+1}$ is an affine function on the equivalence classes.

4 Euclidean space

Affine space where translation space is an inner product space.

5 Flat predictive models

Idea is to restrict model space to linear or affine functions. Although still probably taught as the default model type, very tricky to use well. Rigidity leads to lack of expressiveness and instability, change any training data can have unbounded effects on predictions far from the change.

6 Embedding data in flat spaces

Practical issue is that linear functions only make sense on linear domains; affine on affine domains (which includes linear.)

Data comes as lists of records, with attributes some of whose values are numerical, but many taking value in more complicated domains (eg an address).

Clojure implementation: record = object; dataset = list of records; attribute = function.

After numerical attributes, the next simplest are probably categorical, by which I mean an attribute whose values fall in some finite set. Tricky issue is whether the set of possible value is know ahead of time (usually assumed and usually wrong) or whether every new

data set may have new values (eg USA set of states might change, more common is need to cover territories missing from a training set, etc.)

Whatever set of attributes we start with, we will need to construct an embedding of the record objects into a flat space.

7 Regression

Codomain is $\mathbb{R}$, so we want flat functionals. Complete model is $\mathsf{g} : \mathbb{D} \to \mathbb{R}$ constructed from $\mathsf{e} : \mathbb{D} \to \mathbb{F}^n$ and $\mathsf{f} : \mathbb{F}^n \to \mathbb{R}$ via $\mathsf{g} = \mathsf{f} \circ \mathsf{e}$.

8 Training by minimizing L2 cost

9 Training by minimizing L1 and quantile cost

10 Regularization

'Ridge regression' = minimize L2 norm of parameter vector; 'Lasso' = L1 norm. Neither makes sense

11 Typesetting

This document was typeset using MikTeX 2.9 [4] and TeXworks 0.6.1 [3] on Windows 10. I used `arara` [1] 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.

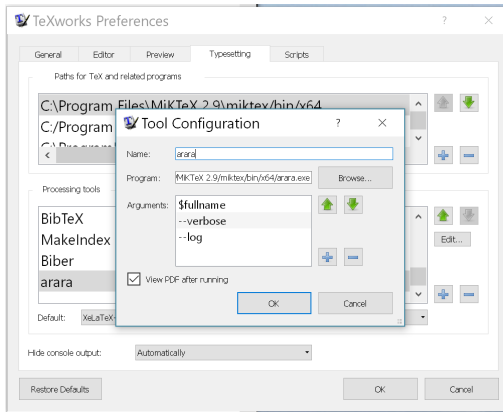

Figure 11.1: Configuring TeXworks for `arara`.

References

[1] 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).

[2] Paul R. Halmos.
Finite-dimensional Vector Spaces.
Princeton, NJ: Van Nostrand, 1958.

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

[4] Christian Schenk.
MikTEX: typsetting beautiful documents.
URL: https://miktex.org/ (visited on 2017-03-13).