Appendix for Uplift Model

April 15, 2016

This is a more thorough description and derivation of the equations for the Uplift modeling framework outlined in UpliftModeling.ipynb by Dorian Goldman. Prepared by Ming Zhao.

First, some definitions:

- $x_i \in X$: independent variable for individual customer i;
- $a_i(x_i) \in A$: action (e.g., treated or untreated, price up or down) for x_i;
- $y_i(x_i) \in R$: reward or outcome for x_i from action a_i;
- $w_i(x_i) \in W$: probability distribution of the population X;
- $B(a|x)$: distribution of action $a(x_i)$ for the experiment, a.k.a., bias;
- $\Pi(a|x)$: similar to $B(a|x)$ but for the whole population, a.k.a. policy;
- Q : probability distribution of reward for the experiment, specifically:

$$Q(y,a,x) = R(y|a,x)B(a|x)w(x); \tag{1}$$

- Ω : distribution of reward, similar to Q but for the whole population:

$$\Omega(y,a,x) = R(y|a,x)\Pi(a|x)w(x); \tag{2}$$

- $h(x_i)$: policy function for x_i, i.e., what action to use for x_i;
- $h^*(x_i)$: optimal police function we want to solve for;

The general goal is to solve for the best policy function h^* that maximizes the expectation of total reward for the whole population:

$$\mathbb{E}_\Omega(y) := \int y(x)\Omega(x)\mathrm{d}x \tag{3}$$

However, since we do not know Ω, we use Q, which we know, to estimate Ω using *importance sampling*:

$$\mathbb{E}_\Omega(y) := \int y(x)\Omega(x)\mathrm{d}x = \int y(x)\frac{\Omega(x)}{Q(x)}Q(x)\mathrm{d}x =: \mathbb{E}_Q\left(\frac{\Omega}{Q}y\right). \tag{4}$$

Recall from equations 1 & 2 that

$$\frac{\Omega}{Q} = \frac{\Pi(a|x)}{B(a|x)}, \tag{5}$$

so Eq 4 becomes

$$\mathbb{E}_\Omega(y) = \mathbb{E}_Q\left(\frac{\Pi}{B}y\right) \sim \sum_{i=1}^{N}\frac{\Pi_i}{B_i}y_iQ_i =: V_h(y) \tag{6}$$

Because of the numerical approximation above, the normalization factor for coefficient $\frac{\Pi_i}{B_i}Q_i$ is unknown. For numerical stability, a common approach is to normalize it with the sum of all coefficients to get the *self-normalized importance sampling estimate*[1]:

$$V_h(y) \sim \hat{V}_h(y) = \frac{\sum_{i=1}^{N}\frac{\Pi_i}{B_i}y_iQ_i}{\sum_{i=1}^{N}\frac{\Pi_i}{B_i}Q_i} \tag{7}$$

Note that $\frac{\Pi_i}{B_i}Q_i$ needs to be normalized together because B_i is part of Q.

Q is the distribution of the experiment, i.e., data we collected, it can be approximated by a sum of delta functions, each represents a customer i. Based on Eq 1 it can be written as:

$$Q(y,a,x) \sim \frac{1}{N}\sum_{i=1}^{N}\delta(y-y_i)\delta(a-a_i)\delta(x-x_i). \tag{8}$$

Similarly, B for the experiment can be written as:

$$B(a|x) = \frac{1}{N}\sum_{i=1}^{N}\delta(a-a_i). \tag{9}$$

Note that a_i is a function of x_i, and can only take a few assigned actions or values. For instance, if $a_i \in \{0 = \text{untreated}, 1 = \text{treated}\}$, then $B(a_i) \in \{B(0), B(1)\}$. And in the special case that a is randomly and uniformly distributed, $B(0) = B(1) = 1/2$. Likewise, in the special case where $y_i \in \{0,1\}$ and $a_i \in \{0,1\}$, and both y and a are uniformly distributed and independent to each other, then $Q_i(y,a,x) \in \{(1,1,x_i),(1,0,x_i),(0,1,x_i),(0,0,x_i)\}$

[1] see §9.2 in: http://statweb.stanford.edu/∼owen/mc/Ch-var-is.pdf

and $Q_i = 1/4$. In the extreme case that everyone gets $y = 1$ regardless of the action a_i, $Q_i(y, a, x) \in \{(1, 1, x_i), (1, 0, x_i)\}$, and $Q_i = 1/2$ for uniform distribution.

For each experiment, the sum of delta functions in Q is generally a constant C, i.e., $Q_i = C_i/N$. C_i is unknown in some cases, or hard to compute for complex distributions, but can be normalized by the self-normalization term in the denominator of Eq 7. Thus, Eq 7 becomes:

$$\hat{V}_h(y) = \frac{\sum_{i=1}^{N} \frac{\Pi_i}{B_i} y_i Q_i}{\sum_{i=1}^{N} \frac{\Pi_i}{B_i} Q_i} = \frac{\frac{C}{N} \sum_{i=1}^{N} \frac{\Pi_i}{B_i} y_i}{\frac{C}{N} \sum_{i=1}^{N} \frac{\Pi_i}{B_i}} = \frac{\sum_{i=1}^{N} y_i \frac{\mathbf{1}(a_i = h(x_i))}{B(a_i|x_i)}}{\sum_{i=1}^{N} \frac{\mathbf{1}(a_i = h(x_i))}{B(a_i|x_i)}} \tag{10}$$

For **special case:** $y \in \{0, 1\}$, we can split the positive and negative parts:

$$\hat{V}_h(y) = \frac{\sum_{i_{\text{pos}}} \frac{\mathbf{1}(a_i = h(x_i))}{B(a_i|x_i)}}{\sum_{i_{\text{pos}}} \frac{\mathbf{1}(a_i = h(x_i))}{B(a_i|x_i)} + \sum_{i_{\text{neg}}} \frac{\mathbf{1}(a_i = h(x_i))}{B(a_i|x_i)}}, \tag{11}$$

where i_{neg} and i_{pos} are the i such that $y_i = 0$ and $y = 1$ respectively. Now define

$$f(h) := \frac{f^-(h)}{f^+(h)} := \frac{\sum_{i_{\text{neg}}} \frac{\mathbf{1}(a_i = h(x_i))}{B(a_i|x_i)}}{\sum_{i_{\text{pos}}} \frac{\mathbf{1}(a_i = h(x_i))}{B(a_i|x_i)}},$$

then

$$\hat{V}_h = \frac{1}{1 + \frac{f^-}{f^+}}.$$

Now we want to evaluate different policies $\hat{V}_h(y)$ by reducing its optimization to classification error, and comparing the standard classification algorithims in sci-kit learn to the R uplift model. To maximize this expression (which is bounded by 1), we want f^-/f^+ to be as close to 0 as possible (0 being the unique maximum of $x \mapsto \frac{1}{1+x}$ on $x \geq 0$).

Notice that this is equivalent to minimizing the strictly convex, bounded from below loss:

$$\mathcal{L}_h = \frac{1}{N} \sum_{i=1}^{N} (y_i - \lambda)(h(x_i) - a_i)^2,$$

where $0 < \lambda < 1$. For now, in order to do this, we simply train well known classifiers on the sets $\{y_i = 0\}$ and $\{y_i = 1\}$. More precisely we want to find h which is close h^*, defined by

$$h^*(x_i) = a_i \text{ if } y_i = 1$$

$$h^*(x_i) = 1 - a_i \text{ if } y_i = 0$$

Now observe since $\mathcal{L}_h$ is strictly convex and bounded from below, it has a unqiue minimizer. Since h^* above achieves the minimum, which is λ, this is the unique solution.