Goal Based Bidding

Krishnan Raman

2015-12-01

1 Forecasting

The SERP ads landscape for a single keyword over one quarter lookback window(1 qtr = 13 weeks = 91 days) is captured below:

DAY	BID	CPC	CLICKS	CONVERSIONS	REVENUE
day1	bid1	cpc1	clicks1	conv1	rev1
day2	bid2	cpc2	clicks2	conv2	rev2
...	...	...	...	...	...
day91	bid91	cpc91	clicks91	conv91	rev91

1. We posit 4 relationships

$$\begin{aligned} cpc &= f(bid) \\ clicks &= g(cpc) \\ conv &= h(clicks) \\ rev &= j(conv) \end{aligned}$$

2. All four functions f,g,h and j are only defined on the $[0, R+]$ space

3. All four functions are required to pass thru the origin
($zerobid => zerocpc => zeroclicks => zeroconv => zerorevenue$)

4. Based on empirical evidence, we assume h and j are linear with zero bias
$conv = h(clicks) => conv = w1 * clicks$
$rev = j(conv) => rev = w2 * conv$

5. Purely because of #4, rpc (revenue per click) is constant!
Proof:

$$\begin{aligned} rpc &= rev/clicks \\ &= w2 * conv/clicks \\ &= w2 * w1 * clicks/clicks \\ &= w2 * w1 \\ &= constant \end{aligned}$$

6. #4 also implies that cvr (a ratio of conversions to clicks) will also be a constant!!! In fact, $cvr = slope(h) = w1$

7. Based on empirical evidence, we require f and g to be monotonically increasing functions that gradually plateau beyond a certain point ie. S-shaped curves.Good candidates for f, g : logistic, log, cosine, inverse-power such as square-root function, cube-root function etc.

8. Given a scatterplot of {bid,cpc} , we have several candidates for f:

   (a) $cpc = a * log(1 + b * bid)$
   (b) $cpc = a * cos(b * bid) - a$
   (c) $cpc = a * bid^{\frac{1}{2+b}}$ (we prefer this form, instead of $cpc = a * \sqrt{bid}$)
   (d) $cpc = \frac{a}{1+e^{-b*(bid-c)}} - \frac{a}{1+e^{b*c}}$

   The Levenberg-Marquardt algorithm (LMA) is used to pick the best fit parameters for each of the candidate curves.The function f is the curve with the lowest rmse for the given data. (Note that a keyword may have a cosine fitting its 91-day data on one day, a logistic on the next day and so forth. The fit is determined purely by rmse.)

   Using LMA requires several rather subtle conditions must be met:

   (a) To begin LMA, a procedure for computing the gradients of each of the unknown parameters must be specified.
   For instance, to begin LMA on the cosine curve, the gradients are
   $\frac{d}{da} = cos(b * bid) - 1$
   $\frac{d}{db} = -a * bid * sin(b * bid)$
   All our candidate curves have specific gradient procedures.
   (b) The presence of origin (0,0) in the dataset introduces computational difficulties with the Jacobian. We use (0.001, 0.001) instead.
   (c) The starting parameters for LMA ie. initial guess, influences the final set of best-fit parameters and rate of convergence. We therefore normalize the dataset $x => \frac{x}{xmax}, y => \frac{y}{ymax}$ so as to ensure the dataset lies in $[0, 1]$ region. The starting guesses are then defaulted to unity. Once LMA has found the best fit params, these params must be "de-normalized" so as to fit the original data.
   (d) Some datasets may cause LMA a long time to converge on certain candidate curves - hence we allow a max of 5 seconds per curvefit & timeout after the time limit (using a sophisticated Scala timer library). In practice, the curvefits happen in a few milliseconds for 99% of the datasets. However, there are pathological cases.

9. The strategy for finding g is identical to that of f, except that we use the scatterplot of {cpc, clicks}.

10. Once f, g, h and j are fully specified, constrained optimization problems can be posed on this mathematical landscape and solved for optima.

2 Optimization

11. Goal Based Bidding can refer to any of the several goals, namely revenue, clicks, conversions etc. The most common goal is revenue maximization. The optimization problem in this case can be simply stated as - "Pick optimal bids to maximize revenue, while constrained by a fixed budget"

12. We are given a portfolio of n keywords with fixed budget B. We seek to find optimal bids for each of these n keywords so as to maximize the revenue from this portfolio, while keeping our costs as close to B as possible. We seek to neither overspend nor go under budget.

13. Each keyword in the portfolio has a valid range of cpcs. ie. $[0, cpc_{max}]$

14. The total cost for the portfolio must equal the budget.
    $cost = \sum_{i=1}^{n} cpc_i * clicks_i = \sum_{i=1}^{n} cpc_i * g_i(cpc_i) = B$

15. We seek to maximize the total revenue from the portfolio, given by
    $revenue = \sum_{i=1}^{n} rev_i = \sum_{i=1}^{n} j_i(h_i(g_i(cpc_i))) = \sum_{i=1}^{n} w_i * g_i(cpc_i)$

16. The optimization problem is given by the lagrangian, obtained by combining the goal #15 with the constraint #14, using the lagrange multiplier λ
    $L = \sum_{i=1}^{n} w_i * g_i(cpc_i) + \lambda * (B - \sum_{i=1}^{n} cpc_i * g_i(cpc_i))$
    This may be simplified by rewriting the constant B as a summation.
    $L = \sum_{i=1}^{n} w_i * g_i(cpc_i) + \lambda * (\sum_{i=1}^{n} \frac{B}{n} - \sum_{i=1}^{n} cpc_i * g_i(cpc_i))$
    The summation operator may then be factorized out.
    $L = \sum_{i=1}^{n} (w_i * g_i(cpc_i) + \lambda * (\frac{B}{n} - cpc_i * g_i(cpc_i)))$
    We now have n independent equations, one per keyword, that may be simultaneously maximized in parallel!

    In other words, for a given value of λ, the set of n equations
    $L_1 = w_1 * g_1(cpc_1) + \lambda * (\frac{B}{n} - cpc_1 * g_1(cpc_1))$

    $L_2 = w_2 * g_2(cpc_2) + \lambda * (\frac{B}{n} - cpc_2 * g_2(cpc_2))$
    ...
    $L_n = w_n * g_n(cpc_n) + \lambda * (\frac{B}{n} - cpc_n * g_n(cpc_n))$
    can all be simultaneously maximized.

17. For a given λ, looking at just one equation at a time ie.
    $L_i = w_i * g_i(cpc_i) + \lambda * (\frac{B}{n} - cpc_i * g_i(cpc_i))$
    we seek the optimal cpc cpc_{opt} in the range $[0, cpc_{max}]$ that maximizes L_i.

18. Finding the optimal x that maximizes the expression
$w * g(x) + \lambda * (\frac{B}{n} - x * g(x))$
is a straightforward exercise, since w, B, n, and λ are constants and g is an S-curve.
Since g(x) is a monotonistically increasing function, the Brent Method(or Bisection, Secant etc.) can be used to find the optimal x.

19. So the more important question becomes : what value of λ is optimal ?
It turns out that high values of cause the budget to be underspent, and low values blow the budget. Since λ is a nonzero number, we could simply investigate the space $[0, \lambda_{max}]$, by starting with a very high λ_{max}. If we blow the budget, increase λ, else decrease λ. Using Bisection, we find the optimal λ that causes the cost to be arbitrarily close to the budget B. In practice, since budget tends to be in dollars, getting the difference between the cost and budget below 1 cent suffices.

20. To reiterate: The GBB revenue maximization problem boils down to n optimizers for *cpc* nested inside 1 giant λ optimizer. Each iteration results in $n+1$ optimizations being performed. After $log(B)$ iterations, we arrive at an optimal solution.

21. Is it possible to obtain a closed-form solution ie. can the optimal cpcs (and hence optimal bids) be computed using an analytic formula amenable to solving by hand, instead of using a computer?
In general, no.
But for certain interesting special cases, it is actually possible to solve the problem by hand !!!
Consider n identical keywords ie. we have the same f, g, h, j functions for each keyword. Furthermore, let $g(x) = \sqrt{(}x)$ and let $h = j = 1$. This problem is easily solvably by hand. The solution is left as an exercise to the interested reader.
Also consider the above scenario, except that each keyword k converts x times as fast as the base keyword ie. the conversion rates are an integral multiple of some common base cvr. Again, this problem is easily solved.
Other hand-solvable exercises include cases where the Taylor expansion of $g(cpc)$ may be used, conveniently eliminating higher powers of cpc, since the prices in dollars tend to be tiny fractions.