Online Supplement to: Improving Web Catalog Design for Easy Product Search

I. Robert Chiang
Accenture, One Financial Plaza, Hartford, CT 06103, USA, robert.chiang@accenture.com
Manuel A. Nunez
School of Business, University of Connecticut, 2100 Hillside Road Unit 1041, Storrs, CT 06269, USA, manuel.nunez@business.uconn.edu

INFORMS Journal on Computing

---

1. Generic Versus Customized Catalogs

One of the key factors that affects catalog design is user experience. User experience affects the choice of distance matrix in the objective functions of both the assignment and design models. In some cases it is possible to compile statistical data to determine average click-counts between pages for a given catalog topology. Let $\hat{D}$ denote the matrix of observed average click-counts, that is, $\hat{D}_{ij}$ represents the average click-count from page i to page j when items shown on the two pages were purchased or evaluated together by multiple users. The matrix $\hat{D}$ might be biased because of an existing catalog topology. There are several ways to "de-bias" the matrix. The first possibility is to cross reference with the actual purchase record, which is less susceptible to the catalog structure. The second possibility is to trim the original click stream data using a "stop watch." Tracking software at the web servers routinely records for each session the pages visited as well as the elapsed time spent on each page. It is thus possible to use a time threshold to identify "transit" pages that were clicked but not evaluated with interest. Another approach to gathering product correlations that is outside the scope of current paper is to conduct an exit survey through a third-party service company (e.g. SurveySpot.com).

Alternatively, an experience factor θ^* can be found to minimize the difference between $\hat{D}$ and $D(\theta)$, and then solve (AP) for θ^*. The advantage of deriving θ^* is that the catalog can be optimized even if a minor topological change is made (e.g. changing from three links per page to four links per page). We call such an approach the "generic catalog approach." The generic approach makes sense when the web site visitors are more or less homogeneous, thus using one catalog (with one "optimized" θ) is sufficient.

Another approach when $\hat{D}$ is unavailable, or the designer wishes to have more flexibility in dealing with different levels of user experience, is to parametrically solve (AP) (or (DP)) in the paper for multiple experience factors θ and then, dynamically choose which optimal catalog topology to use according to a visitor's profile. We call such an approach the "customized catalog approach." In this section we discuss the implementation details of the generic and customized catalogs.

1.1. Generic-Catalog Approach

To determine a value of θ such that the weighted matrix $D(\theta) = \theta D^E + (1-\theta)D^N$ is as close to the observed distance matrix $\hat{D}$ as possible, we solve the following optimization problem

$$\text{(FT)} \quad \min \left\{ \|D(\theta) - \hat{D}\| : 0 \leq \theta \leq 1 \right\}.$$

In defining (FT) we use the Frobenius matrix norm. In other words, for a matrix M we have

$$\|M\| := trace(M^T M)^{1/2}.$$

Let θ^* denote the solution to (FT). Since (FT) is a simple one-dimensional constrained quadratic problem, it is easy to find its solution explicitly. If

$$\lambda := \frac{trace\left(\left(D^E - D^N\right)^T \left(D^N - \hat{D}\right)\right)}{\|D^E - D^N\|^2} \in]0,1[,$$

then set $\theta^* = \lambda$. Otherwise set

$$\theta^* = \begin{cases} 0 & \text{if } \|D^N - \hat{D}\| \leq \|D^E - \hat{D}\|, \\ 1 & \text{if } \|D^N - \hat{D}\| > \|D^E - \hat{D}\|. \end{cases}$$

1.2. Customized-Catalog Approach

In this approach we parametrically solve (AP) (or (DP)) in the paper and determine a set of optimal catalog topologies for various values of θ. Multiple versions of the catalog are kept and one of the versions will be used to interact with a visitor depending on the visitor's profile. When a visitor visits the web site for the first time, an experience factor θ close to zero is assigned to the visitor and the topology based on the distance matrix $D(\theta)$ (closer to D^N) is used. The web site would keep track of the frequency of visits of the visitor. Once the number of rapid clicks (before reaching the next page of interest) decreases, the visitor's

experience factor θ has probably improved, thus a different topology (closer to D^E) could be used.

In particular it is interesting to study the shape of the objective function of (AP) as a function of θ. Let $z(\theta)$ denote the optimal objective value of (AP) as a function of the experience factor θ. Consider the set Q representing the feasible region for (AP), that is,

$$Q := \left\{ X : Xe = e, X^T e = e, X_{il} \in \{0,1\}, \forall i, l \right\}.$$

Notice that $|Q| = n!$. Let $X \in Q$ be fixed. Since $D^N \geq D^E$, we have

$$trace\left(FXD^E X^T\right) - trace\left(FXD^N X^T\right) \leq 0,$$

and so, $\theta\left(trace\left(FXD^E X^T\right) - trace\left(FXD^N X^T\right)\right) + trace\left(FXD^E X^T\right)$ is a linear function on θ with negative slope, that is, it is linearly decreasing in θ for X fixed.

Therefore, the function $z(\theta)$ is the minimum of $n!$ decreasing linear functions evaluated at θ, so that $z(\theta)$ must be a piecewise linear concave decreasing function of θ. As such, the lowest value of $z(\theta)$ is attained at $\theta = 1$, i.e., when the user has perfect visibility (perhaps by memorizing) and uses the shortest path to search for the next item. On the other hand, when $\theta = 0$, $z(\theta)$ achieves its maximum value, corresponding to the totally random search for the next item. Moreover, the optimal assignment $X(\theta)^*$ remains constant inside the interval segments where $z(\theta)$ is linear with respect to θ. In other words, the optimal solution is unstable only at the points where $z(\theta)$ changes its slope.

When the number of linear pieces in $z(\theta)$ is small, the interval $[0,1]$ can be partitioned into a small number of disjoint subintervals such that there is a unique optimal catalog topology associated with each subinterval. Hence, the web catalog designer needs to keep only a small number of catalog topologies to deal with a wide range of different types of user experience. Computational experience shows that the number of linear pieces in $z(\theta)$ is generally small. Similar remarks apply to the design model.

For example, consider a 4-node problem with frequency matrix obtained by normalizing the weights given to pairs of items according to Table 1. Figure 1 shows the optimal objective value of (DP) as a function of θ. As mentioned before, the objective values determine a piecewise linear concave decreasing function with roughly three linear segments. For each segment there is a unique optimal topology and assignment as illustrated in Figure 2. Even though there are 912 possible solutions (4! permutations for each of the 38 possible connected topologies) only three are optimal and need to be maintained.

Table 1: Weights Between Pairs of Items

Item	A	B	C	D
A	0	10	16	1
B	100	0	5	15
C	100	200	0	15
D	500	500	200	0

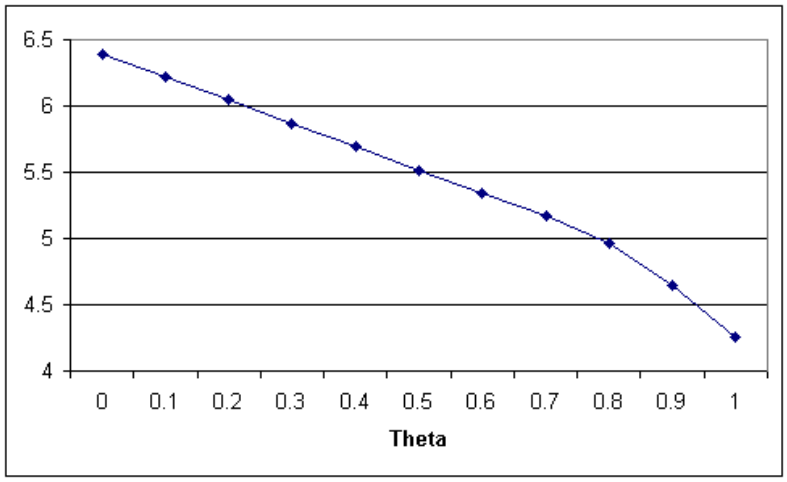

Figure 1: (DP) Optimal Objectives as θ Varies

Finally, we generated two samples of 10 problem instances each. Each sample instance consists of 12 nodes/pages. In the first sample the entries of each frequency matrix were randomly generated from a uniform distribution over the interval $[30, 70]$ and then normalized to satisfy (3) from the paper. On the other hand, in the second sample the entries of each frequency matrix were generated from two different distributions: one is uniform over the interval $[30, 70]$ for entries below the diagonal and the other is uniform over the interval $[480, 520]$ for entries above the diagonal. The idea in the second sample is to have asymmetric skewed frequencies for each problem instance. We used complete enumeration to determine the number of linear pieces of the optimal objective function $z(\theta)$ as θ ranged from 0 to 1 for each instance. Figure 3 shows the distribution of the number of pieces for both samples. Notice that for problems in both samples the number of linear pieces is generally small and the instances with fully uniform frequencies tend to have slightly more linear pieces than the instances with skewed frequencies.

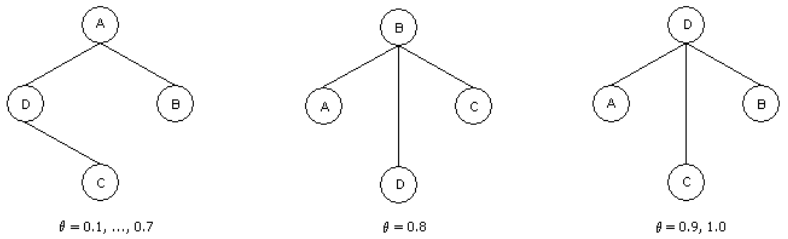

Figure 2: Optimal Topologies and Assignments as θ Varies

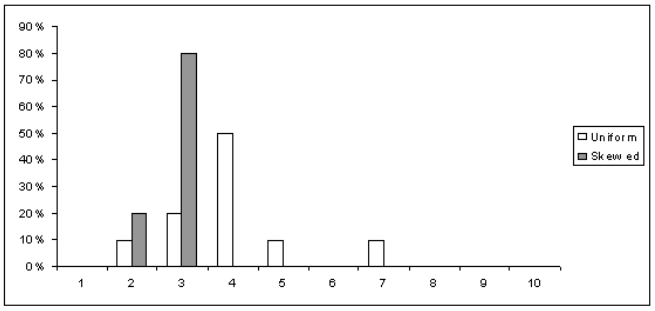

Figure 3: Frequency of Linear Pieces in the $z(\theta)$ Function

2. Incorporating Item Sequences

In our analysis we have assumed that the user's decision to go to the next page to search for an item depends solely on the item on the currently visited page, that is, the search process is memoryless in that the next visited item does not depend on the sequence of visited items except for the last one. Our model can be extended to deal with a more complex setup in which sequences of 2 or more items are taken into account. To simplify, suppose that we only want to consider traversals consisting of exactly $s \geq 2$ items. Let $F_{i_1 \cdots i_s}$ be the proportion of occurrences in which the sequence $i_i, \cdots, i_s$ of items were purchased (in that order) in a collection of shopping carts or was evaluated (in that order) in a collection of browsing sessions. Similarly to (3) in the paper, we have

$$\sum_{i_1 \cdots i_s} F_{i_1 \cdots i_s} = 1. \tag{1}$$

Given a sequence $i_i, \cdots, i_s$ of items located on pages $p_i, \cdots, p_s$, respectively, a distance matrix D, and assuming that distances are additive, we obtain a total traveled distance of

$$D_{p_1 \cdots p_s} := \sum_{j=1}^{s-1} D_{p_j p_{j+1}}. \tag{2}$$

Using (1) and (2), we obtain the following polynomial optimization assignment problem in terms of the assignment matrix X,

$$\begin{aligned}
(\text{PAP}) \quad \min \quad & z(X) := \sum_{i_1 \cdots i_s} \sum_{p_1 \cdots p_s} F_{i_1 \cdots i_s} D_{p_1 \cdots p_s} X_{i_1 p_1} \cdots X_{i_s p_s}, \\
\text{s.t.} \quad & \\
& Xe = e, \\
& X^T e = e, \\
& X_{il} \in \{0,1\}, \forall i, l;
\end{aligned}$$

where, in this case, $z(X)$ represents the average number of clicks it takes to find any sequence of s consecutive items. Analogously, we obtain the polynomial design problem,

$$\begin{aligned}
(\text{PDP}) \quad \min \quad & z(X,Y) = \left(\sum_{i_1 \cdots i_s} \sum_{p_1 \cdots p_s} F_{i_1 \cdots i_s} D_{p_1 \cdots p_s}(Y) X_{i_1 p_1} \cdots X_{i_s p_s} \right) + C(Y), \\
\text{s.t.} \quad & \\
& Xe = e, \\
& X^T e = e, \\
& \mathcal{N} V^k = b^k, \forall k, \\
& \sum_{k=1}^{n} V^k \leq n(n-1)Y, \\
& V^k \geq 0, X_{il}, Y_{kl} \in \{0,1\}, \forall i, k, l,
\end{aligned}$$

where $D_{p_1 \cdots p_s}(Y)$ is defined by using (2) and the corresponding distance induced by the topology Y.

Even though the more general polynomial assignment problem (PAP) is discussed in some papers (see Barvinok and Stephen, 2003, for instance), to the best of our knowledge no approximation heuristic has been proposed for the case $s \geq 3$. On the other hand, since the feasible regions of (PAP) and (PDP) are the same as the feasible regions of (AP) and (DP) in the paper, respectively, we readily obtain an approximation method by using our GA approach to solve both polynomial problems. Notice that the chromosome representation and the genetic operators (mutation and crossover) remain the same for both problems. The only difficulty arises in computation of the objective value for each chromosome within a given population, which is more complex than in the quadratic case.

In particular, when $s = 3$ we obtain an interesting result in the following theorem.

Proposition 1 *Given a distance matrix D, suppose that we estimate the multiple-item frequency array F for $s = 3$ by using a pair-wise frequency matrix $\hat{F}$ as follows*

$$F_{ijk} = \hat{F}_{ij}\hat{F}_{jk}, \tag{3}$$

where $\hat{F}$ has been normalized so that

$$\sum_{ijk} \hat{F}_{ij}\hat{F}_{jk} = 1.$$

Then the objective $z(X)$ of the corresponding (PAP) problem reduces to

$$z(X) = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{p=1}^{n}\sum_{q=1}^{n} \hat{F}_{ij}\left(\hat{F}_{+i} + \hat{F}_{j+}\right) D_{pq}X_{ip}X_{jq}, \tag{4}$$

where $\hat{F}_{+k}$ and $\hat{F}_{k+}$ are the totals of the k-th column and the k-th row of $\hat{F}$, respectively.

Proof. Under the conditions of Proposition 1, we have

$$\begin{aligned}
z(X) &= \sum_{ijk}\sum_{pqr} \hat{F}_{ij}\hat{F}_{jk}D_{pq}X_{ip}X_{jq}X_{kr} + \sum_{ijk}\sum_{pqr} \hat{F}_{ij}\hat{F}_{jk}D_{qr}X_{ip}X_{jq}X_{kr} \\
&= \sum_{ij}\sum_{pq} \hat{F}_{ij}D_{pq}X_{ip}X_{jq}\sum_{k}\hat{F}_{jk}\sum_{r}X_{kr} + \sum_{jk}\sum_{qr}\hat{F}_{jk}D_{qr}X_{jq}X_{kr}\sum_{i}\hat{F}_{ij}\sum_{p}X_{ip} \\
&= \sum_{ij}\sum_{pq} \hat{F}_{ij}D_{pq}X_{ip}X_{jq}\hat{F}_{j+} + \sum_{jk}\sum_{qr}\hat{F}_{jk}D_{qr}X_{jq}X_{kr}\hat{F}_{+j} \\
&= \sum_{ij}\sum_{pq} \hat{F}_{ij}\left(\hat{F}_{j+} + \hat{F}_{+i}\right) D_{pq}X_{ip}X_{jq},
\end{aligned}$$

and the result follows. ■

The significance of Proposition 1 is that if we define a matrix G such that

$$G_{ij} := \hat{F}_{ij}\left(\hat{F}_{+i} + \hat{F}_{j+}\right),$$

for all i, j, then

$$z(X) = trace\left(GXDX^T\right),$$

and (PAP) for $s = 3$ reduces to a regular quadratic assignment (AP) in the paper. Furthermore, the simplification technique described in the proof of the proposition can be used to reduce any other (PAP) instance to a quadratic assignment problem with an appropriately chosen frequency matrix as long as we use an analogous estimation (3), that is,

$$F_{i_1\cdots i_s} = \prod_{j=1}^{s-1} \hat{F}_{i_j i_{j+1}},$$

with

$$\sum_{i_1 \cdots i_s} \prod_{j=1}^{s-1} \hat{F}_{i_j i_{j+1}} = 1.$$

Similar remarks apply to the polynomial version of (DP) in the paper.

3. Greedy Heuristic for (AP)

Input parameters: Graph $G = (N, E)$ and frequency matrix F.

Output parameter: Assignment matrix X.

Algorithm body:

$X \leftarrow 0$;
for each $i = 1, \ldots, |N|$ **do**
    ItemState$_i \leftarrow$ unassigned;
    PageState$_i \leftarrow$ free;
end for;
stop $\leftarrow$ **false**;
while not stop **do**
  if exists an unexplored item **then**
    $i \leftarrow$ next unexplored item;
    if ItemState$_i$ = unassigned **then**
      $k \leftarrow$ next free page;
      $X_{ik} \leftarrow 1$;
      ItemState$_i \leftarrow$ assigned;
      PageState$_k \leftarrow$ not free;
    end if;
    for each $(k, l) \in E$ **do**
      if PageState$_l$ = free **then**
        $j \leftarrow \arg\max\{F_{ij} : \text{ItemState}_j = \text{unassigned}\}$; (*)
        $X_{jl} \leftarrow 1$;
        ItemState$_j \leftarrow$ assigned;
        PageState$_l \leftarrow$ not free;
      end if;
    end for;

ItemState$_i \leftarrow$ explored;
else
stop $\leftarrow$ **true**;
end if;
end while;
return X.

Each item has three states: unassigned, assigned, and explored. Each page has two states: free and not free. Initially all items are unassigned and all pages are free. When an unassigned item is placed onto a page its status changes to assigned (but still unexplored). Once all pages connected to a page containing an assigned item become not free, then the assigned item changes state to explored. Statement (*) implements the greedy strategy, i.e., given an assigned item i, the algorithm will find the item j with the highest frequency F_{ij} from among the unassigned items to link from the page containing item i.

4. Greedy Heuristic for (DP)

Input parameters: Frequency matrix F and integer vector $(L_1, \ldots, L_n)$.
Output parameter: Graph $G = (N, E)$ and assignment matrix X.
Algorithm body:

$X \leftarrow 0$;
$N \leftarrow \{1, \ldots, n\}$;
$E \leftarrow \emptyset$;
k $\leftarrow 0$;
for each $i = 1, \ldots, |N|$ **do**
ItemState$_i \leftarrow$ unassigned;
$\pi_i \leftarrow 0$;
end for;
stop $\leftarrow$ **false**;
while not stop **do**
if exists an unexplored item **then**
$i \leftarrow$ next unexplored item;
if ItemState$_i$ = unassigned **then**

$k \leftarrow k + 1$;

$\pi_i \leftarrow k$;

ItemState$_i \leftarrow$ assigned;

end if;

for each $h = 1, \ldots, L_{\pi_i}$ **do**

$j \leftarrow \arg\ h\text{-}\max\{F_{ij} : 1 \leq j \leq |N|, j \neq i\}$; (**)

if $\pi_j = 0$ **then**

$k \leftarrow k + 1$;

$\pi_j \leftarrow k$;

ItemState$_j \leftarrow$ assigned;

end if;

$E \leftarrow E \cup \{(\pi_i, \pi_j)\}$;

end for;

ItemState$_i \leftarrow$ explored;

else

stop $\leftarrow$ **true**;

end if;

end while;

for each $i = 1, \ldots, |N|$ **do**

$X_{i\pi_i} \leftarrow 1$;

end for;

return G and X.

The element L_k of the vector $(L_1, \ldots, L_n)$ represents the maximum number of links allowed on page k. Each item has three states: unassigned, assigned, and explored. The integer π_i represents the page to which item i was assigned when $\pi_i > 0$ and indicates that item i has not been assigned when $\pi_i = 0$. Initially all items are unassigned. When an unassigned item is placed onto a page its status changes to assigned (but still unexplored). Once all links on a page containing an assigned item connect to other items, then the assigned item changes state to explored. Statement (**) implements the greedy strategy, i.e., given an assigned item i and an integer h, the algorithm will find the item j with the h-th highest frequency F_{ij} from among the other items.

References

Barvinok, A., T. Stephen. 2003. The distribution of values in the quadratic assignment problem. *Mathematics of Operations Research* **28** 64–91.