A Decision Theoretic Framework for Implicit Relevance Feedback

Onno Zoeter Nick Craswell Michael Taylor John Guiver Ed Snelson
Microsoft Research Cambridge
Cambridge, United Kingdom
{onnoz,nickcr,mitaylor,joguiver,esnelson}@microsoft.com

Abstract

In the process of developing a search system, there is the difficult question of how to evaluate search results quality. A system can be evaluated on multiple metrics, and we must rely on domain experts to define these metrics and their relative importance. This paper presents a decision theoretic approach, which separates the modeling of user behavior from the evaluation (utility) function. The sole task of the model is to predict user behavior, for example predicting explicit relevance labels and implicit 'click' behavior. The utility function describes what kind of ranking would be preferred by users. Given a model and the utility function, we can see the ranking problem as an optimal decision problem. We describe this decision theoretic approach in more detail and present a simple example of a decision theoretic model.

1 Introduction

When building a search engine there is the fundamental issue of evaluation, with an underlying question 'what search results do we believe users will prefer?' An improved evaluation metric, that captures user preferences more accurately, can help us build a system that better satisfies users. Evaluation is not a machine learning question, but a machine learning system should work in harmony with a powerful evaluation framework.

In classical Information Retrieval, the most common form of evaluation is via human relevance labels. For each query some documents are labeled, and we reward a system if it ranks positively-labeled documents above negatively-labeled ones. However, there are problems with labels. Real users have a multitude of subjective opinions on relevance, and these depend on context. By contrast, labelers are usually hired hands rather than real users, who are not in a real context and they must make guesses about real user opinions. Labels may not always look like an unbiased sample of real user opinions.

Modern search engines observe user behavior, by logging which queries were typed, which results were presented and which results were clicked. Click information has an advantage over labels, because it pertains to real users. It allows us to observe the subjective choices of users as they view a results list, and build information about the true population of users, perhaps also recording information about context (for example user location and native language). This is also much less expensive than employing relevance labelers. However, clicks are noisy because users sometimes click and later decide a document was irrelevant. Also, we can only gather clicks on results that users see, and users look at very few documents per query. Lastly, clicks are prone to cheating. In this paper we will not touch upon in this potential fraud and assume an ideal stream implicit feedback.

Labels and clicks each have their own advantages, and there may be other criteria we wish to capture. For example, reducing the amount of repetition in search results can increase user satisfaction, but does not necessarily increase the number of positively-labeled documents. Assuming the system is capable of producing results with many positively-labeled and much redundancy, but is also capable of producing one with fewer positively-labeled and less redundancy, there is the problem of choosing a tradeoff between the two that best satisfies the user.

The question is how to build a model that works well, given the above concerns about evaluation. One approach (Figure 1a) is to choose an evaluation measure as the gold standard for relevance, such as the label-based metric DCG [4], and build a model to optimize it such as LambdaRank [2]. The inputs may be features to do with query and documents, and possibly even past user click behavior [1]. The ranking of documents is according to a score (R), which is optimized for a particular cost function. The cost function may be based on single documents, pairs of documents or rankings of documents, but for simplicity Figure 1a shows R for a single document.

Another approach (Figure 1b) has the observed query/document features as input, the observed user behavior as output and a hidden node R that gives the relevance score for ranking. For example, predicting clicks based on the identity of the query-document pair and the document's position in the ranking [3]. In general the outputs may include both clicks and labels. In this approach, if there is careful design to model noise in the outputs, it is possible to argue that the relevance score R is correct. However, because no explicit decision has been made on an overall metric/utility, it may be difficult to evaluate.

A third approach (Figure 1c) has the same sort of inputs and outputs, but now the model has the sole task of predicting outputs. Unlike the other two approaches, the ranking score R is not in the model at training time. Instead, ranking is according to utility U, which is a function of the observable variables. The utility function might give a ranking score R for each individual document (as in the figure), but the more powerful case is a multi-document list-based utility. We later consider list-based metric DCG, but the utility could incorporate DCG, click metrics and list diversity metrics. We note that the design and specification of the ranking utility is a completely separate task from that of modelling and predicting user interactions.

All three approaches incorporate end user input in the ranker and thereby effectively turn the ranking problem into a collaborative filtering problem. We concentrate on the third approach, which is decision theoretic, and present a simple example of a model.

2 A decision theoretic framework for implicit relevance feedback

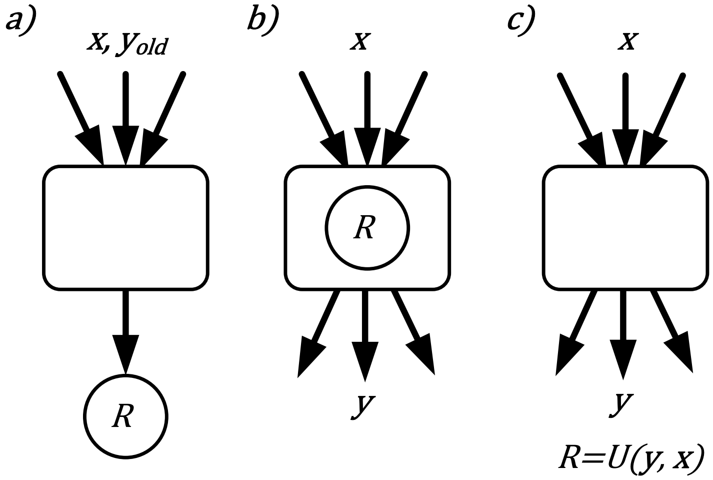
Decision theory is a very well established field which dates back at least to the works of Daniel Bernoulli in the 18th century. The decision problem we will be interested in here has the following very general form.

Given a set of inputs $x \in \mathcal{X}$ we are asked to select an action $a \in \mathcal{A}$. After performing the action we observe outputs $y \in \mathcal{Y}$. A utility function $U : \mathcal{X} \times \mathcal{Y} \mapsto \mathbb{R}$ assigns a scalar utility (reward) to the observed x, y-pair. The outputs in general do not follow deterministically from the inputs and action. A model $p(y|x, a)$ gives the probability of observing y after selecting a when x is observed. The optimal action a^* is the action that leads to the maximal expected utility

$$a^* = \operatorname*{argmax}_a \mathbb{E}_{p(y|x,a)} \left[U\left(x, y\right) \right] .$$

We propose to view the document selection and ranking problem as an optimal decision problem in this very general framework. The action a then is to select n links to documents from a set $\mathcal{D}$ after a query is issued. In other words, if for now we do not exclude multiple links to the same document, the action a is one from $\mathcal{D}^n$, an astronomically large space. (Section 3.1 discusses some of the implications of working with such a large space of actions). The action is taken at a moment when several inputs x are known. What exactly is considered to be part of x depends on the sophistication of the system, but it will typically contain

Figure 1: Three different approaches to the incorporation of implicit feedback into a ranker; a) uses historic user behavior as input to predict a single relevance score R, b) postulates a single R variable that explains how user behavior y follows from inputs x and at the same time is a good scale for ranking. Approach c), proposed in this paper, constructs the best possible model to explain y from x and separately defines a utility function U that puts a preference ordering on possible explicit and implicit behaviors.



the query itself and properties of the documents in $\mathcal{D}$. In richer models we could expect to have information about the user (such as her geographical location), the historical scoring behavior of a human judge that will score a document, etc. We will give several examples in the following section. It is important however to stress that the elements in x, what is known at ranking time, can be interpreted very broadly. Many possible extensions and refinements of the basic ranking model can be treated as a suitably general interpretation of the above model. Section 6 will give an example.

The added benefit of the decision theoretic interpretation is in the modeling of multiple outputs: apart from the rating of each document, y can include many implicit relevance feedbacks such as clicks, dwell times, click backs, abandonments, etc. The utility $U(x, y)$ provides a yardstick that tells us the degree of success of the action (the selection and ranking). The utility function can reward, via the dependence on x, basic properties of the selected documents such as their freshness and the reputability of their sources. In addition, through y, certain behaviors of users will be rewarded. For instance, if a judge gives a good rating to a selected document, the reward will be higher (note that in this framework the ratings are not considered to be ground truths, but are stochastic; if multiple judges would rate the same document, ratings can be different). The reward will be higher if the best rated documents are shown at the top. Similarly $U(x, y)$ can be very low for an abandoned query, higher for a query with many clicks with long dwell times at the landing page, etc. How exactly certain behaviors are rewarded, and which are preferred depends on the choice of yardstick, on the choice of U.

3 Examples of utility functions

3.1 Discounted Cumulative Gain

The discounted cumulative gain (DCG) [4] is an example of a utility function that only takes into account the human relevance scores at every position. It is based on a discount function $d(p)$ over positions $p \in \{1, \ldots, n\}$, and a gain function $g(s)$ over human relevance scores $s \in \{1, \ldots, 5\}$. The position discount function is monotonically decreasing from the top position $p = 1$, to the bottom position $p = n$: $d(1) > d(2) > \cdots d(n)$, and a gain function $g(s)$ that is increasing for better relevance scores: $g(1) < g(2) < \cdots < g(5)$. If $s[1], \ldots, s[n]$

are the scores received for the documents selected by a, the discounted cumulative gain is given by

$$DCG\left(s[1], \ldots, s[n]\right) = \sum_{p=1}^{n} d(p) g(s[p])\,.$$

To maximize the DCG we would select and rank such that the expected DCG is highest. The expectation is then with respect to $p(s[1], \ldots, s[n]|x, a)$ which represents the best estimate of the human relevance scores for the documents selected by a given x

$$a^* = \arg\max_a \mathbb{E}_{p(s[1],\ldots,s[n]|x,a)} \left[\sum_{p=1}^{n} d(p) g(s[p]) \right]\,.$$

Different choices of $g(s)$ lead to different ranking principles (decision rules). If $g(s)$ is convex in s the resulting principle is *risk seeking*: for two documents with the same expected rank but different variance the document with the larger variance is preferred. This is because a larger than expected score leads to a bigger rise in utility than the decrease in utility that results if a lower than expected score is encountered. We could say that such a convex gain functions leads to a "going for the jackpot" effect. The often used exponential function $g(s) = 2^s - 1$ (for example [7]) has this effect. It is important to realize that this is not a conservative ranking principle.

If we have a linear gain $g(s) = s$, the expected utility only involves the expectations of scores:

$$\begin{aligned} a^* &= \arg\max_a \mathbb{E}_{p(s[1],\ldots,s[n]|x,a)} \left[\sum_{p=1}^{n} d(p) g(s[p]) \right] \\ &= \arg\max_a \sum_{p=1}^{n} d(p) \mathbb{E}_{p(s[1],\ldots,s[n]|x,a)} \left[s[p] \right]\,. \end{aligned}$$

hence we get a ranking principle that simply orders documents according to their expected human relevance score:

$$a^* = \arg\max_a \sum_{p=1}^{n} d(p) \mathbb{E}_{p(s[p]|x,a)} \left[s[p] \right]\,.$$

This utility function is an example where the optimal action can be found in $\mathcal{O}\left(|\mathcal{D}|\right)$ time despite the fact that the space of all possible selections and rankings is $\mathcal{D}^n$. This is due to the fact that the score probability $p(s[p]|x, a)$ is not explicitly a function of position (the judge is presented with each document independently). This means that the expected score can be computed for each document and the documents simply sorted to obtain the optimal ranking. There are many interesting utility functions that lead to $\mathcal{O}\left(|\mathcal{D}|\right)$ ranking principles, but in general approximations might be necessary.

Note that, since there is no element in the utility function that encourages diversity in the results, we need to explicitly add the constraint that links to documents cannot be replicated. Otherwise a^* would be n duplications of the link with the highest expected HRS.

3.2 Clicks

An analogous utility function that only takes into account whether a user clicked on a document or not could be

$$U_{\text{clicks}}(c[1], \ldots, c[n]) = \sum_{p=1}^{n} d(p) c[p]\,.$$

If $p(c[p] = 1|x, a)$, the probability of a click on the document that was put in position p by a, is modelled based on a link specific and position specific contribution it will in general not simplify to an $\mathcal{O}(|\mathcal{D}|)$ ordering rule. This is because now $p(c[p]|x, a)$ is explicitly a function of p — any given document will be clicked with a different probability depending on where it is placed. It can be that position and link effects combine in complex non-linear ways, in particular if unknown parameters are treated in a Bayesian way. However there are suitable heuristics for ordering in $\mathcal{O}(|\mathcal{D}|)$, e.g. compute the probability a document will be clicked if it were placed in position 1, and order by that.

To encourage diversity, one simple approach would be to introduce a concave function f of the simple DCG-like sum of clicks:

$$U_{\text{clicks page}}(c[1], \ldots, c[n]) = f\left(\sum_{p=1}^{n} d(p)c[p]\right) .$$

This captures the notion that the step from 0 clicks to 1 is bigger than that from 1 to 2. The transformed utility would penalize systems with DCG near zero. For an ambiguous query with several types of result, a ranking optimised to avoid zero DCG could potentially present results of each type, hedging its bets by giving a more diverse results list. It then becomes important to model correlations between click events. An independence assumption is probably rather coarse in the case of clicks. For instance for ambiguous queries, clicks on links to two different interpretations will be anti-correlated. We will briefly come back to this in Section 6.

3.3 Combinations of basic utilities

The decision theoretic framework allows for a principled trade-off between desired behavior of the searcher and relevance cues from a selected set of human judges. In general the utility function should depend on both. A straightforward way is by taking weighted combinations of basic utility functions such as presented in Section 3.1 and 3.2.

A benefit of the decision theoretic framework is that the probabilistic model that predicts user behaviors and the preference ordering over them (utility function) can be developed largely independent. Once the type of outputs are decided upon, different models can be constructed, tested and compared using example x, y pairs (e.g. using log likelihood). The parameters in a utility function, for instance the weights given to each component in a linear combination of simple utilities, can be tweaked and adjusted over time without the need of changing the model. This is something that would not be easy to do in the approaches (a) and (b) in Figure 1.

4 An example model: binomial generalized linear models

As an example we consider a generative model $p(y|x, a)$ that is deliberately simple. It is for more than one reason a sub-optimal model for the ranking problem, but its simplicity allows us to focus on the key implications of the decision theoretic framework. We will use a Bayesian treatment of a generalized linear model [5].

In the model we take as outputs per position the click event and 6 human relevance labels

$$y[p] = [\text{click}, \text{bad}, \ldots, \text{perfect}] .$$

Each element in the vector $y[p]$ is a binary indicator.

The input x contains two elementary relevance scores for every document p that is selected by the action a: a BM25f score [8], and a link-based popularity score (static rank). Both are discretized to allow the fitting of arbitrary non-linearities within the GLM framework. We also have a query ID indicator and a query-document ID indicator, which are only on for specific instances of that query and query/document pair respectively. This allows us to effectively store query and query/document specific statistics in the weights. In addition there is a bias term that is always 1. The input for the label submodels is given by

$$x_{\text{label}}[p] = \left[\text{bias} = 1, \text{bm}_1, \ldots, \text{bm}_k, \text{sr}_1, \ldots, \text{sr}_l, \text{q}_1, \ldots, \text{q}_q, \text{qd}_1, \ldots, \text{qd}_m\right] .$$

The input for the click sub model has additionally a position indicator

$$x_{\text{click}}[p] = \left[\text{bias} = 1, \text{pos}[p] = 1, \text{bm}_1, \ldots, \text{bm}_k, \text{sr}_1, \ldots, \text{sr}_l, \text{q}_1, \ldots, \text{q}_q, \text{qd}_1, \ldots, \text{qd}_m\right] .$$

Click Utility
1.: `http://www.adobe.com/products/acrobat/readstep2.html`
2.: `http://www.adobe.com/products/acrobat/readermain.html`
3.: `http://www.adobe.com/products/acrobat/alternate.html`
Human Relevance Utility (DCG)
1.: `http://www.adobe.com/`
2.: `http://www.adobe.com/downloads`
3.: `http://www.adobe.com/products/acrobat/readermain.html`
Mixed Utility
1.: `http://www.adobe.com/products/acrobat/readermain.html`
2.: `http://www.adobe.com/downloads`
3.: `http://www.adobe.com/`

Table 1: Reorderings of the top-ranked positions for the “Abobe” query

Again all elements in these vector are binary indicators.

For each position and output type we have a submodel. That is, the model factorizes as

$$p(y|x) = \prod_{p=1}^{n} p(\text{click}[p]|x_{\text{click}}[p]) \prod_{i=\text{bad}}^{\text{perfect}} p(y[p,i]|x_{\text{label}}[p])\,.$$

A generalized linear model (GLM) consists of a likelihood model $p(y|\theta)$, a linear combination of inputs x and model weights w: $x^\top w$, and a link function $g(\theta)$ that maps the parameter θ to the real line. In this section we will use building blocks that have a binomial likelihood model and a logit (log-odds) link function. In a generative model interpretation the *inverse* logit link function

$$g^{-1}(s) = \frac{1}{1+\exp(-s)}$$

plays a central role. This inverse link function is the well known sigmoid function that maps the outcome of the inner product $x^\top w \in \mathbb{R}$ to the $[0,1]$ space of the success parameter θ in the binomial. If we have N examples in our training set for which the inputs have value x, and we observe c positive outcomes, the likelihood becomes:

$$p(c|x,w) = Bin\left(c; g^{-1}\left(x^\top w\right), N\right)\ . \tag{1}$$

When used for prediction, x would represent the properties of the selected documents and $N = 1$.

To learn the settings of w we use the approximate Bayesian inference procedure from Appendix A with a factorized Gaussian prior.

5 Experiments

We have collected three months worth of data from the logs of a major web search engine. For each query in that time period, we have recorded the number of times each url in the result lists have been clicked on, and also the number of times they have been presented (and not clicked on). We also have a break down of these aggregates by position. We use a small subset of 200,000 query-document pairs, where we also have human relevance labels and document content feature vectors. We focused on the first three ranked documents for reliability of data and simplicity. We trained the GLM model as described in the previous section and appendix, and we used the model predictions and different utilities to make different optimal orderings for specific queries.

Table 1 shows the results of using three different utilities to re-order the results of the query “Abobe”. We observe that the click utility favours popularity — each of the results point in some way to Acrobat Reader. The Human Relevance utility favours definitiveness, and the 'definitive' `www.adobe.com` is presented first. The combined utility prefers some combination between the two. We can weight the combined utility in different ways between the two extremes to make a trade-off.

This is just one example query out of many, and more extensive experimentation needs to be done to definitively capture the effects of using different utilities for ranking.

6 Extensions: Whole page relevance

The decision theoretic framework is a principle for whole page relevance. Only for certain models and utility functions (e.g. the ones discussed in Sections 3.1 does it reduce to a link based ranking principle. Many interesting models that make use of whole page relevance can be considered. For instance the concave function of the sum of clicks from Section 3.2 is an example of a whole page utility. It penalizes 0 clicks and would therefore encourage diversity on the result page.

A model that is flexible enough to track diversity on a page could for instance make use of a multi-dimensional latent space of interests, where each axis represents a genre or topic. A document would correspond to a point in this space indicating how well it covers each of the topics (jaguar the car and jaguar the animal say). An intention underlying a query can then be a vector in this space. A projection of the document point to the intention vector indicates the degree of agreement. A probability of click is then a function of this agreement. We could in a training set for instance observe that for the query "jaguar" there are two dominant vectors (intentions) in the population of searches: one roughly in the car, and one in the animal direction. A utility that penalizes 0-clicks on a page would then lead to a diverse offering, even if the top n car links are all on average more popular than the most popular animal link.

7 Summary

We have presented a decision-theoretic approach to document ranking, that separates the task of modeling user behavior from the task of defining a utility function. The model, given document/query information, predicts user behavior such as explicit labels or implicit click records. A utility function then describes what kind of ranking would be most useful for users, in terms of the observable document/query information and user behavior. For a particular model and a particular utility function, the decision theoretic approach defines an optimal ranking.

In this approach, the model has the sole task of predicting observable variables, and does not necessarily model or make assumptions about end-user 'relevance'. The utility function can be defined by a domain expert, who defines a small number of metrics and can potentially control their balance by hand-tuning one or two parameters. This adjustment can affect the ranking immediately, without the need to retrain the model.

References

[1] E. Agichtein, E. Brill, and S. Dumais. Improving web search ranking by incorporating user behavior information. In *SIGIR*, 2006.

[2] C. Burges, R. Ragno, and Quoc Viet Le Le. Learning to rank with nonsmooth cost functions. In *NIPS*, 2006.

[3] G. Dupret, V. Murdock, and B. Piwowarski. Web search engine evaluation using clickthrough data and a user model. In *WWW Workshop on Query Log Analysis*, 2007.

[4] Järvelin and J. Kekäläinen. IR evaluation methods for retrieving highly relevant documents. In *SIGIR*, 2000.

[5] P. McCullagh and J. A. Nelder. *Generalized Linear Models.* CRC Press, 2nd edition, 1990.

[6] Thomas Minka. *A Family of Algorithms for Approximate Bayesian Inference.* PhD thesis, Massachusetts Institute of Technology, January 2001.

[7] Marc Najork, Hugo Zaragoza, and Michael Taylor. HITS on the web: How does it compare? In *SIGIR*, pages 471–478, 2007.

[8] S. Robertson, H. Zaragoza, and M.J. Taylor. A simple BM 25 extension to multiple weighted fields. In *CIKM*, pages 42–29, 2004.

[9] Onno Zoeter. Bayesian generalized linear models in a terabyte world. In *IEEE Conference on Image and Signal Processing and Analysis*, 2007.

[10] Onno Zoeter and Tom Heskes. Gaussian quadrature based expectation propagation. In *AISTATS*, pages 445–452, 2005.

A Quadrature EP for binomial GLMs

The non-linearity of the inverse link function $g^{-1}(s)$ in (1) has the effect that with a Gaussian prior on w the posterior after observing an input-output pair x_i, y_i

$$p(w|x_i, y_i) = \frac{p(y_i|x_i, w)p(w)}{p(y_i|x_i)}$$

is not Gaussian. For the logit-binomial there is no compact closed form solution to this posterior.

We use quadrature EP [10] to approximate the posterior over the weights $p(w|\{x_i, y_i\})$ by a factorized Gaussian. Quadrature EP is a generalization of expectation propagation [6] where general non-linearities are handled using low dimensional Gaussian quadrature approximations. This approximation is particularly suited for GLMs [9]. For GLMs with Gaussian priors both the likelihood terms and the prior are log-concave. Since log-concavity is preserved under multiplication, the posterior therefore is guaranteed to be log-concave as well. This guarantees uni-modality of the posterior and makes a Gaussian approximation very suitable.

Within quadrature EP a proposal distribution (the kernel in the Gaussian quadrature) is required. For many GLMs the prior is a suitable choice. But since for large N the binomial likelihood can shift the prior significantly, the prior might be ill matched with the posterior.

For binomial GLMs our initial approach was to use a proposal distribution based on a Gaussian approximation of the likelihood using Laplace's method. If we define $s = x^\top w$ and have a Gaussian prior

$$p(s) = N(s; m, v)$$

and the Laplace approximation to the likelihood as

$$\begin{aligned} Bin\left(c; g^{-1}(s), N\right) &\approx zN\left(s; m_{\text{Lap}}, v_{\text{Lap}}\right) \\ m_{\text{Lap}} &= \arg\max_s Bin\left(c; g^{-1}(s), N\right) \\ \left(v_{\text{Lap}}\right)^{-1} &= -\frac{\partial^2}{\partial s^2} \log Bin\left(c; g^{-1}(s), N\right)\Big|_{s=m_{\text{Lap}}} , \end{aligned}$$

the proposal distribution is given by the product of the prior and the Gaussian approximation to the likelihood

$$r(s) \propto N(s; m, v) N\left(s; m_{\text{Lap}}, v_{\text{Lap}}\right) .$$

Both the mode and the variance can be found in closed form based on derivatives. However, we have found that the mean of this proposal distribution can still be significantly off and lead to failure of the quadrature. The practical solution that we have adopted in our initial experiments has been to use the precision derived from the above approximation, with the mean set to the mode of the posterior, found numerically. We have found this to be a very robust approach defining the proposal distribution.

For $N = c$ and $c = 0$ we use a Gaussian proposal with a mean given by the mode of the posterior (again found numerically), and a variance equal to the prior variance.