The Impact of Images on User Clicks in Product Search

Sung H. Chung1, Anjan Goswami1, Honglak Lee2, Junling Hu1
1eBay Inc., 2065 Hamilton Ave., San Jose, CA, 95125
2Computer Science & Engineering, University of Michigan, Ann Arbor, MI, 48109
{suchung,angoswami,juhu}@ebay.com, {honglak@eecs.umich.edu

ABSTRACT

Product search engine faces unique challenges that differ from web page search. The goal of a product search engine is to rank relevant items that the user may be interested in purchasing. Clicks provide a strong signal of a user’s interest in an item. Traditional click prediction models include many features such as document text, price, and user information. In this paper, we propose adding information extracted from the thumbnail image of the item as additional features for click prediction. Specifically, we use two types of image features – photographic features and object features. Our experiments reveal that both types of features can be highly useful in click prediction. We measure our performance in both prediction accuracy and NDCG. Overall, our experiments show that augmenting with image features to a standard model in click prediction provides significant improvement in precision and recall and boosts NDCG.

Categories and Subject Descriptors
H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval; I.2.6 [Artificial Intelligence]: Learning

General Terms
Experimentation, search engine

Keywords
Click Prediction, Online Shopping, Product Search Ranking, Vertical Search, Image Features

1. INTRODUCTION

With the rapid growth of E-commerce, product search becomes an essential tool for people to find relevant items in online shopping. The challenge of a product search engine is to rank relevant items that users are interested in purchasing. Clicks are typically a strong signal indicating a user’s interest in an item. Therefore, correctly predicting the click probability of each item is an important step to maximize sales revenue.

Standard click prediction models use query, item (document), and user information as inputs for modeling. Item features typically include the text of an item and price (in product search). However, it has been well studied in marketing and consumer psychology [29] that the product images play a critical role for online shopping. In addition, we believe that images are important due to the following reasons: First, our search log data show that search result pages with images get more clicks than those without images for the same query. Figure 1 shows a search result page for women’s dresses, where images are featured prominently. Second, previous experiments [12] show that ranking with image quality features boosts NDCG based on human judgment.

In this paper, we built a click prediction model with item image features as inputs. We adopt a logistic regression model for predicting the click probability. Our model uses typical factors in click prediction model for product search [38], which includes query-level features, item text, price, and seller reputation features. On top of these standard features, we add features extracted from the thumbnail image of an item. Our image features include both photographic image quality features, such as brightness and contrast, and
Click prediction has been well studied in online advertisement and sponsored search domains [8, 7, 41]. The probability of clicking on a link can be modeled with a statistical classifier. Such a classifier assigns probabilities to two possible outcomes: click and no click. Thus the problem can be transformed into training a good binary classifier that predicts the likelihood of clicks, and the training can be done with existing query log data. There are two ways of improving a classifier: One is selecting the most appropriate machine learning algorithm, the other is improving features for the classifier.

People have experimented with different machine learning algorithms for click prediction, including ranking SVM [39], binary SVM [21], probit regression [14], decision trees [11], and gradient boosted decision trees [41, 12]. In this paper, we use logistic regression, which has proven to be fast and achieves similar performance to SVM [9]. Although we experimented with gradient boosted decision trees as well (not reported here), we found that logistic regression is less prone to over-fitting and achieves better performance on our dataset.

The second way to improve a classifier’s performance is to use good features. For this purpose, researchers have used various features for click prediction model. The most important feature is historical Click Through Rate (CTR) when such data exist [32, 8, 17]. However, in eBay, millions of new items daily get listed and these new items will not have such historical data. When an item does not have click history, many other features can be used. Two major types of features are item-related features and user-related features. Item-related features use information of document text or category to calculate its relevance to the query. User-related features include both user demographic information and user behavior information. König et al. [24] considered various text features from the query and documents for estimating CTR for news queries. Cheng et al. [8] used user specific and demographic information for click prediction in sponsored search. They also found historical CTR and text features very useful. Hassan et al. [16] used user behavior information, such as dwell time and query reformulation, to build click prediction model.

For product search, we can introduce additional features, such as price, as part of item-related features. In addition, we can add other types of features, such as seller-related features. This typically includes seller ratings and user reviews on the seller. Wu and Bolivar [40] uses seller-related features such as the percentage of positive feedback and top seller score in the conversion prediction model, which is closely related to the click prediction model. Wang et al. [38] used a comprehensive feature set that they categorize into four types: item, search, buyer (user) features, and seller features.

In a short paper from last year [12], we studied the correlation between image features and user clicks in product search. In this paper, we build click predictor models over multiple domains to demonstrate the effect of using image features on user clicks; furthermore, we investigate the problem of (1) learning features from images and (2) selecting effective features from a large number of features to address the over-fitting issue.

Images have been extensively studied for image search engines [2, 22]. The purpose of such research is to directly rank images related to a query. In our experiments, we want to use images along with item and user related features to rank product listings. In product listings, images are presented along with with other information such as item titles, item descriptions and item prices.

Features extracted from images vary in different computer vision applications [30]. Specifically, scale invariant feature transform (SIFT) [28], Histogram of Oriented Gradients (HoG) [10], wavelets, and color histogram [30] are among the most popular features used in various image retrieval systems. In addition, recently, features automatically learned in an unsupervised fashion through Restricted Boltzmann Machines (RBMs) have become popular in the computer vision community (e.g., [26, 36]). RBM features are convenient in that they can be used to automatically generate a large number of useful features. RBMs have a
In this paper, we used several photographic features. Some of these features are commonly used in assessing picture quality in digital camera industry. Recently, such features have also been used to assess product image quality [13] for online shopping. Additionally, we constructed more complex object image features to represent image color and shape. Furthermore, we apply a variant of RBM features to capture shape information in our experiments.

For evaluating click prediction models, most researchers have used precision, recall and F-measure [41, 16, 38, 24]. However, DCG has also been used by a few researchers [6]. In this paper, we measure our model performance against precision/recall, AUC and NDCG.

3. CLICK PREDICTION FRAMEWORK

Click Prediction Model. The goal of a product search engine is to present a list of products relevant to the query. A user typically clicks on a product listing if he or she is interested in buying it. Hence, user clicks are a strong indicator of the buying intent.

In this paper, we model the problem of click prediction as a binary classification problem. We use logistic regression as our classifier to predict the probability of a click given the query, item and user information. Thus, our goal is estimating the following conditional probability: 

$$P(X = 1 | query \ and \ item).$$

The conditional probability of click or no-click given the features can be written using the logit function as follows:

$$P(y = 1 | x) = \frac{\exp(w^T x)}{1 + \exp(w^T x)}.$$  

(1)

Here, \(x \in \mathbb{R}^n\) denotes a vector of feature variables, and \(y \in \{0, 1\}\) denotes no-click and click classes, respectively. Each data point \((x_i, y_i)\), where \(i \in \{1, ..., m\}\), corresponds to the \(i\)-th query-item pair. The logistic regression model has parameters \(w \in \mathbb{R}^n\) that need to be learned, and the maximum likelihood learning of logistic regression (with the entire set of training examples) is expressed as follows:

$$\min \frac{1}{m} \sum_{i=1}^{m} \log(1 + \exp(-y_i (x_i^T w)))$$  

(2)

Feature Selection. As we will describe later, our system uses a large number of features. In such cases, most standard classification algorithms tend to have an increased risk of over-fitting. To address this problem, we used the \(L_1\) regularized logistic regression [27] to perform feature selection. The \(L_1\) logistic model that we use for feature selection solves the following optimization problem:

$$\min \frac{1}{m} \sum_{i=1}^{m} \log(1 + \exp(-y_i (x_i^T w))) + \lambda \sum_{i} \|w_i\|$$  

(3)

where the variables are \(w \in \mathbb{R}^n\), and \(\lambda > 0\). The regularization parameter \(\lambda\) controls the number of nonzero components in \(w\), and it is determined by cross-validation. Prior to using \(L_1\) regularized logistic regression for feature selection, features have to be normalized. In our experiments, we found that \(L_1\) regularized logistic regression can yield significant improvement in performance.

Baseline (non-image) Features. Although our focus in this paper is image features, the followings are standard features that are often used universally in other search problems or click prediction problems. We use them for our baseline models.

- **Item features:**
  - **Total cost:** Total cost is the sum of the product price, tax and the shipping cost.
  - **Shipping cost:** The shipping cost of the item.
  - **Condition:** This is a discrete variable denoting if the item is new, used, or refurbished.

- **Query features:**
  - **Query-item title text match:** This score captures the goodness of text match between the query and the item title considering proximity [3] of the query words in the item title. We considered the length of minimum span in the title that contains the query words as a measure of proximity. Note that every item title in the search recall set [3] must contain all the query words.
  - **Query clicks over impression:** This is the number of clicks per impression for a given query. This measures how often clicks happen for that particular query.

- **Seller features:**
  - **Seller reputation:** Seller reputation can be computed based on several factors [23] and it is an important subject of research. We use a couple of different variants. They are computed using user ratings given by buyers.
  - **Seller-item click through rate:** The average click through rate for all items listed by the seller.
  - **Seller-trust:** The probability that a buyer will experience a problem with the seller (e.g., a defective item, delayed shipping, etc.).

4. IMAGE-RELATED FEATURES

We categorize image features into two distinct groups—photographic features and object features. Photographic features are directly derived from image color and intensity values and are oblivious to the contents of the images. In contrast, object features try to capture descriptions of the contents, such as object shapes, colors and textures.

In the following subsections, we describe each of these features in details.

4.1 Photographic Features

Our photographic features are largely divided into two types—global features and regional features. We describe each of these features in details.

4.1.1 Global Features

Aspect Ratio: This is the image height divided by its width.

Brightness: This is the average of gray-scale intensity values of all the pixels. We used the following standard expression to convert from RGB values to gray-scale values: 

$$0.3R + 0.6G + 0.1B$$

Dynamic Range: This is the range of the gray-scale values defined as \((max - min)\). We discard outlier pixels to make this score more robust.

Contrast: This quantity represents visual properties that make an object appear clearer. There are different kinds of
In addition to the global features described in the previous section, we generate the following region-based photographic features. Due to their dependence on image segmentation, there are some object content elements to these features. Nonetheless, in our experiments, these features are grouped as photographic features rather than object features.

**Lightness of Background:** We observe that a lot of good product images have a light-colored background. We compute the RGB distances of the background pixels from a pure white pixel and take their mean and standard deviation.

**Uniformity of Background:** A good product image typically does not have a high variance in the background. This feature is computed as the standard deviation of the gray-scale intensity values of the background.

**Colorfulness of Foreground:** There has been extensive research on perceptual quality of color and there are many variants of colorfulness. We use an expression that first appeared in a paper by Hasler and Susstrunk [15] based on sRGB color space.

**Ratio of Background to Foreground Area:** In this feature, we compute the ratio between the foreground area and the background area. A larger ratio indicates that the product size of the product is larger in the frame.

**Background and Foreground Brightness Difference:** This feature attempts to capture the difference of brightness between the background and the foreground. A bigger value usually means that the object is more clearly visible.

**Background and Foreground Contrast Difference:** This captures the contrast between the background and the foreground. A higher contrast usually accentuates the product better.

## 4.2 Object Features

We have three types of features related to objects in an image: color histograms, texture histograms, and shape features. We provide details of these features below.

### 4.2.1 Color Histogram

Once an image is segmented, foreground color pixel values are extracted in the HSV (Hue Saturation Value) format. Then, these pixel HSV values are quantized into an 11 bucket histogram.

### 4.2.2 Texture Histogram

The horizontal and vertical Sobel filters are applied on the gray-scale image. Then, the magnitude of the edge response at each position is computed. We then build an 8-bucket histogram that captures edge responses within the foreground.

### 4.2.3 Shape Features

Using an extension of Restricted Boltzmann Machine (RBM) [19], we automatically learn filters that capture representative object shapes and are robust to certain transformations such as translations.

**Restricted Boltzmann Machine:** The RBM is a generative probabilistic model that is often used to learn features or to initialize deep neural networks in an unsupervised manner [20, 5, 33]. Specifically, the RBM is a two layer undirected graphical model that models joint probability distribution of visible and hidden variables; the visible nodes represent the data while the hidden nodes represent the features discovered by training the RBM. If the visible nodes are real-valued, the joint probability distribution of RBM can be described using the following equations:

\[
P(v, h; \theta) = \frac{1}{Z(\theta)} \exp(-E(v, h; \theta))
\]

\[
E(v, h; \theta) = \frac{1}{2} v^T W v - b^T v - a^T h
\]

where \(v\) and \(h\) denote visible and hidden variable vectors respectively, and \(Z(\theta)\) is the normalization constant. The parameters of the RBM (that need to be learned) are \(\theta \triangleq \{W, b, a\}\). Specifically, \(W\) is the weight matrix between visible and hidden units (i.e., \(W_{ij}\) represents the symmetric interaction between \(v_i\) and \(h_j\)); \(b\) and \(a\) are the bias terms for visible and hidden units, respectively.

The RBM has symmetric connections between the two layers denoted by a weight matrix \(W\), but no connections within hidden nodes or visible nodes. Therefore, it is easy to compute the conditional probability distributions when \(v\) or \(h\) is fixed. In practice, the following conditional probability of the hidden nodes are used as the features representing the input data \(v\):

\[
p(h_k = 1|v) = \text{sigmoid} \left( \sum_l W_{lk} v_l + a_k \right)
\]

where \(\text{sigmoid}(x) = 1/(1+\exp(-x))\) is the sigmoid function.

Unsupervised training of the RBM can be performed by approximately maximizing the following log-likelihood with a technique called contrastive divergence approximation [18].

\[
\log P(v; \theta) = \log \left( \frac{1}{Z(\theta)} \sum_h \exp(-E(v, h; \theta)) \right)
\]

In addition, stacks of RBMs can be greedily trained layer by layer to construct the Deep Belief Network (DBN). For detailed discussions of RBMs and DBNs, see [19, 4].

**Convolutional Restricted Boltzmann Machine:** Lee et al. [26] have proposed the convolutional RBM (CRBM) where the filters are shared among hidden units that correspond to different locations in the image. They have also shown that stacks of multiple CRBM can be used to learn translation invariant object-shape filters from images of arbitrary sizes beyond small image patches. Their model uses sparsity regularization that encourages hidden nodes to have sparse activations and filters to learn interpretable patterns [25]. They further developed a technique called probabilistic max-pooling that achieves local invariance and allows upper layers to learn increasingly larger shapes.

In short, CRBM can be used to learn image filters that capture representative shapes from the training data without exact spatial alignments among visual patterns. Therefore, we choose to use CRBM to learn shapes in our item categories.

1. In the context of object image features, the visible nodes corresponds to vectorized pixel values of an image patch.
2. For the case of binary-valued visible nodes, the energy function can be defined as \(E(v, h; \theta) = -v^T Wh - b^T v - a^T h\).
Prior to training CRBM, the images are first converted to gray-scale and then whitened to remove pair-wise correlations between nearby pixels [35]. This process is known to encourage learning shapes as opposed to learning intensity variations in the image.

We train two sets of filters using different scale/layer combinations. For the first set, we learn filters that encode rough overall shapes of items. To do this, rather than building multiple layers of filters (as done in [26]) we scale down 140x140 item images into small 24x24 images and then train a single layer CRBM with 200 15x15 filters (If item images are not square images, then the shorter side is appended with border pixels to make them square images.). This is computationally much cheaper than learning upper layers and yet allows us to learn many filters that capture overall shapes of items. Figure 2 shows examples of the single-layer filters learned from a few eBay categories. It is apparent that these filters capture shapes from each category.

For the second set of filters, we train the second layer filters that use the first layer outputs as inputs. The first layer outputs come from 24 10x10 natural first-layer bases (oriented edge filters). The pooling ratio used is 3—roughly speaking, this means that each pixel learned from the second layer, in fact, represents three pixels in the original raw image.

There are 250 second layer filters per category, and the filter size is 14x14. Since the pooling ratio is 3, these filters represent shapes as large as 42x42 pixels in the original images. These second layer filters encode item part shapes that retain more details than the first-layer overall-shape filters. Figure 3 visualizes the second layer filters selected from a few categories.

After the filters are learned, the actual image features are extracted from each image by computing the hidden unit responses. In typical RBM, the hidden unit response is computed as the probability of the hidden unit being ‘on’ given the visible unit values (Equation 6). In our experiments, the feature for a filter is chosen as the maximum response value among convolutional unit responses over the image. This has an intuitive interpretation of shape detection within an image with a sliding window.

### 4.3 Image Segmentation

Before computing photographic and object features from an image, we first segment the image into the foreground and the background. We use an automated segmentation process that is based on GrabCut [34]. The original GrabCut is an interactive image segmentation tool where the user has to provide a rectangular bounding box for the foreground object. All the pixels outside of the bounding box are marked as the background. The algorithm computes the distribution of the background colors and then iteratively estimates the foreground portion and the background portion within the bounding box.

We automate this algorithm by simply choosing a bounding box that is slightly smaller than the item image, leaving the edge pixels of the item image to represent the background colors. In cases where this process fails (for instance, all or the majority of the image is segmented as the background), the algorithm will simply choose a rectangle surrounding the center of the image as the foreground. The size of each dimension of the rectangle is chosen as the half of the corresponding image dimension length. Despite such a simple modification, our automated image segmentation process works surprisingly well. Figure 4 shows automatically segmented examples.

### 4.4 Query Image Relevance Score

Since our task is to learn relationship of an item to a particular query, it would be helpful to have a feature that can relate item image features to a query. A visual dictionary that maps text tokens to image features could be useful. Such a dictionary is often used in image retrieval.
systems [37]. In such systems, one typically builds a probabilistic model for each concept, \( \Pr(X | \text{concept}) \), which describes the distribution of image features for a particular concept. We can use a dictionary of such concepts to measure relevance of a query to a particular item image.

We use a heuristic approach where we map each text token to a vector of image features. This vector of image features is computed as an average feature vector of all the images that a token is associated with.

In order to build a visual dictionary, one needs a large database of annotated images. We take advantage of the fact that eBay already has a large number of annotated images—every item image that a seller uploads to the site can be considered an annotated image since it is accompanied by the item’s title. Therefore, we perform the following steps to build visual dictionaries from eBay image data.

1. Collect a set of items, their titles and their corresponding images. The items in the set would be from the same category.
2. Find out the most frequent tokens used in the titles of the items. Remove stop words and punctuation. This collection of tokens is the keys in our dictionary.
3. For each item, extract all the object image features mentioned in previous sections from the image.
4. Extract word tokens from the item title.
5. Look up individual title tokens in our dictionary. If a token is found, then the item’s image features are used to compute the average feature vector for the particular token.

Once an average feature vector is obtained for a particular token, then we measure the relevance of a query to a particular item image by looking up individual tokens of the query in the visual dictionary for their average feature vectors, and then computing their inner products with the image features of the item in question. If there are multiple tokens in the query, then we simply take the average of inner product results of all the tokens.

Although the way we build the dictionary is simple, our experiment shows that it works reasonably well and has a strong correlation with item clicks.

5. EXPERIMENTS

In this section, we describe our data sets, experiments and results.

5.1 Data Collection and Preparation

Our data are collected from eBay logs spanning one month. Each sample corresponds to either a click or a skip in the search result page, where a skip refers to an event where an un-clicked item appears before another item that is clicked (i.e. a skipped item appears at a higher rank than a clicked item). Each sample is associated with a query and an item. Additionally, each item associated with a query is a multi-quantity, fixed-price item. In eBay, there are two different types of items - an auction type and a fixed-price type. User behaviors on auction items are very different from fixed-price items in that a large number of clicks may happen close to the auction ending time. In contrast, multi-quantity, fixed-price items are typically long-lasting items that are not affected by duration or purchases. We do not use auction items in our experiments. We also only consider samples with at least 100 combined skips and clicks. Finally, the data are collected from specific item categories because we are interested in building category dependent click predictors. Potential bot crawl events are also filtered out.

We randomly chose four eBay categories — Digital Cameras, Women’s Dresses, Women’s Boots and Chairs. However, our experiments can easily be applied to any other categories. Table 1 shows the number of the total number of clicks and skips collected from each category.

We use a separate dataset to train CRBM features. About 30,000 to 80,000 images from each category are used to generate the CRBM shape filters. These images and their corresponding item titles are also used to construct the visual dictionary of text tokens, as described in Section 4.4.

5.2 Prediction Models

We evaluated a baseline model using all the non-image features described in Section 3. In addition, we evaluated our three models augmented with image features as follows:

- The first model uses all the baseline features, all the photographic features, and the query image relevance feature described in Section 4.4.
- Our second model uses all the baseline and image features. They include 469 object features that describe colors, textures, and shapes of items (via CRBM).
- Our third model applies L1-regularized logistic regression feature selection to all the images and baseline features.

The best values we found for the L1-regularization parameter \( \lambda \) were typically between 3 and 30, depending on categories. To find the best values, we ran L1-regularized logistic regression against training folds several times with different \( \lambda \) values, yielding different sets of selected features. For each set of selected features, we trained unregularized logistic models against the training folds and tested them against validation folds. In the end, we chose the \( \lambda \) values that yielded the best performing models against validation folds. The average number of selected features ranged from 20 (in "Chairs" category) to 460 (in "Dresses" category).

For all the models mentioned above, we use logistic regression as the classifier and validate the results using 10-fold cross validations.

5.3 Performance Measure

We evaluated the performance of each model with the precision and recall rate. We also calculate Area under the ROC curve (AUC) for each model. In addition, we measured average NDCG (Normalized Discounted Cumulative Gain) at positions 5 and 10 over validation query-result sets. The

<table>
<thead>
<tr>
<th>Categories</th>
<th>Clicks</th>
<th>Skips</th>
</tr>
</thead>
<tbody>
<tr>
<td>Digital Cameras</td>
<td>53383</td>
<td>4678570</td>
</tr>
<tr>
<td>Chairs</td>
<td>9828</td>
<td>73383</td>
</tr>
<tr>
<td>Dresses</td>
<td>179637</td>
<td>18514341</td>
</tr>
<tr>
<td>Boots</td>
<td>19836</td>
<td>-3180167</td>
</tr>
</tbody>
</table>

Table 1: Dataset composition
Table 2: AUC (area under the ROC curve) in different categories (best performance shown in bold)

<table>
<thead>
<tr>
<th>Categories</th>
<th>Baseline + Photo &amp; Query-Image Features</th>
<th>Baseline + All Image Features</th>
<th>Feature Selection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chairs</td>
<td>0.5765</td>
<td>0.5927</td>
<td>0.5946</td>
</tr>
<tr>
<td>Boots</td>
<td>0.5824</td>
<td>0.5932</td>
<td>0.6001</td>
</tr>
<tr>
<td>Dresses</td>
<td>0.5553</td>
<td>0.5689</td>
<td>0.5925</td>
</tr>
<tr>
<td>Digital Cameras</td>
<td>0.6029</td>
<td>0.6315</td>
<td>0.6423</td>
</tr>
</tbody>
</table>

Table 3: NDCG at positions 5 and 10 in different categories (best performance shown in bold)

<table>
<thead>
<tr>
<th>Categories</th>
<th>Position</th>
<th>Baseline + Photo &amp; Query-Image Features</th>
<th>Baseline + All Image Features</th>
<th>Feature Selection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chairs</td>
<td>5</td>
<td>0.4737</td>
<td>0.4916</td>
<td>0.4936</td>
</tr>
<tr>
<td>Boots</td>
<td>10</td>
<td>0.5622</td>
<td>0.5924</td>
<td>0.5693</td>
</tr>
<tr>
<td>Dresses</td>
<td>5</td>
<td>0.4462</td>
<td>0.4578</td>
<td>0.4499</td>
</tr>
<tr>
<td>Digital Cameras</td>
<td>10</td>
<td>0.5462</td>
<td>0.5546</td>
<td>0.5528</td>
</tr>
</tbody>
</table>

Table 4: Top-ranked Features after feature selection

<table>
<thead>
<tr>
<th>Digital Cameras</th>
<th>Chairs</th>
<th>Boots</th>
<th>Dresses</th>
</tr>
</thead>
<tbody>
<tr>
<td>ShippingCost</td>
<td>SellerRep1</td>
<td>SellerRep1</td>
<td>ShippingCost</td>
</tr>
<tr>
<td>MichelsonContrast</td>
<td>Layer2CRBM94</td>
<td>DynamicRange</td>
<td>SellerTrust</td>
</tr>
<tr>
<td>Layer2CRBM74</td>
<td>Layer2CRBM155</td>
<td>MichelsonContrast</td>
<td>Texture7</td>
</tr>
<tr>
<td>Saturation</td>
<td>Layer2CRBM227</td>
<td>Brightness</td>
<td>Texture0</td>
</tr>
<tr>
<td>Brightness</td>
<td>SellerRep2</td>
<td>BGFGAreaRatio</td>
<td>QueryImageScore</td>
</tr>
<tr>
<td>DynamicRange</td>
<td>Layer2CRBM59</td>
<td>Texture1</td>
<td>Layer2CRBM221</td>
</tr>
<tr>
<td>WhiteColor</td>
<td>Layer2CRBM141</td>
<td>Texture2</td>
<td>Layer2CRBM96</td>
</tr>
<tr>
<td>BGLightness</td>
<td>Layer2CRBM144</td>
<td>RMSContrast</td>
<td>Layer2CRBM92</td>
</tr>
<tr>
<td>ColorHue7</td>
<td>Layer2CRBM190</td>
<td>Layer2CRBM234</td>
<td>Texture1</td>
</tr>
<tr>
<td>SellerRep2</td>
<td>Layer2CRBM3</td>
<td>SellerRep2</td>
<td>Texture2</td>
</tr>
</tbody>
</table>

NDCGs were averaged with an equal weight for each query. We use the following version of DCG:

\[ DCG_p = \text{rel}_1 + \sum_{i=2}^{p} \frac{\text{rel}_i}{\log_2 i} \]  

(8)

The normalized DCG is defined as DCG/IDCG where IDCG is computed using the ideal ranking. We assume that the ideal ranking for each query is to rank recall items by their click-through rates. Therefore, we set each query-item sample’s relevance as \( \text{clicks}/(\text{clicks} + \text{skips}) \). This is similar to DCG usage in [1].

5.4 Experimental Results

For each of the prediction methods and four product categories, we report the click prediction performance using area under the ROC curve (AUC), click prediction rate (precision) vs. recall curve, and NDCG.

The AUC values of different prediction methods are shown in Table 2. The precision vs. recall curves are shown in Figure 5. With both metrics, we observe improvements across all the categories with the image features. In all these results, we can see that the L1-regularized logistic regression feature selection process is highly effective.

The results of NDCG measurements are shown in Table 3. We can see that ranking performance also improves with image features in all categories.

We performed paired t-tests on baseline and the best results and found that these improvements are statistically significant with p-values well below 0.001.

5.5 Image Features after Feature Selection

In order to find out which features are most important in our tasks, we examined coefficient magnitudes of selected features in each category. To make a fair comparison, the feature values were properly normalized prior to the feature selection process. In Table 4, we report top 10 features in each product category.

Somewhat surprisingly, we can see that image features dominate the list. In other words, except for a few seller features and the shipping cost, all the other features in the list are image-related features. In addition, both photographic features and object features are well represented in this list. Another interesting fact is that prominent CRBM features are mostly the second layer features. This is consistent with other observations in machine learning literature [20, 5, 19, 4, 26] that higher-layer features often exhibit stronger discriminative power than lower-level features.

Our results clearly suggest that images play a very important role in attracting user’s attention in product listing pages.
Figure 5: Click Prediction (precision) vs recall curves for different models. In eBay, typical click-through-rate is around 0.02 and 0.03.

6. CONCLUSIONS
In this paper, we introduce image features into click prediction model for product search. We generate many kinds of image features, such as color, texture and shape features generated by convolutional RBM. Additionally, we introduce query-to-image relevance score. Given that there are many features, we conduct automatic feature selection with L1-regularized logistic regression.

We show that image features, along with the feature selection method, can significantly improve our click prediction model in both precision and recall.

7. REFERENCES
[6] B. Carterette and R. Jones. Evaluating search engines by modeling the relationship between relevance and


