Probabilistic Proactive Timeline Browser

Antti Ajanki[1] and Samuel Kaski[1,2]

[1]Aalto University School of Science
Department of Information and Computer Science
Helsinki Institute for Information Technology HIIT
PO Box 15400, 00076 Aalto, Finland
[2]University of Helsinki, PO Box 68, 00014 University of Helsinki, Finland
{antti.ajanki, samuel.kaski}@tkk.fi

Abstract. We have developed a browser suitable for finding events from timelines, in particular from life logs and other timelines containing a familiar narrative. The system infers the relevance of events based on the user's browsing behavior and increases the visual saliency of relevant items along the timeline. As recognized images are strong memory cues, the user can quickly determine if the salient images are relevant and, if they are, it is quick and easy to select them by clicking since they are salient. Even if the inferred relevance was not correct, the timeline will help: The user may remember if the sought event was before or after a saliently shown event which limits the search space. A user study shows that the browser helps in locating relevant images quicker, and augmenting explicit click feedback with implicit mouse movement patterns further improves the performance.

Keywords: Image retrieval, implicit relevance feedback, interaction, machine learning

1 Introduction

One of the main reasons why image retrieval is difficult is that it is hard to formulate effective queries. In contrast to texts, images cannot be easily automatically decomposed into low level units which the user could combine to form queries. Two common image retrieval approaches are retrieval based on textual metadata, such as image captions, and content-based image retrieval (CBIR). The content-based approach [2] is typically an iterative process where the retrieval system returns a set of images and the user is required to grade their degree of relevance. The retrieval system then updates its estimate of the desired image features and retrieves a new set of images.

Explicit feedback in the form of clicking the relevant images or giving rankings is rather accurate but requires judgement calls by the user and is therefore either laborious or scarce. Use of implicit feedback has been suggested as an alternative. The idea is to measure, as a by-product of normal use, data about how the user interacts with the system, and infer relevance from the measurements that are indirectly related to relevance. Implicit feedback has been shown to be useful in text [9, 10] and image retrieval [4, 7, 8], but not very accurate as the only source of feedback.

When the images are ordered and the order is familiar, new forms of retrieval and feedback become possible. This is the case for life logs [5] and other narratives where the images are strong memory cues for recalling past events. We are in the process of developing an image and event retrieval method that utilizes the capability of humans to very effectively recognize familiar events from cues associated with the event, in our case images.

Our search interface helps in locating images in two ways: first, the images are displayed on a timeline to allow using the temporal neighborhood as a rough search cue. Secondly, the search interface suggests potentially relevant images by making them more salient. The relevance is inferred from explicit and implicit feedback using probabilistic inference. Previous information re-finding methods (e.g. [3]) require manually entering the remembered details as a search query. Our hypothesis is that automatic relevance prediction and visualization of the relevance estimates decreases the effort required to find the correct images.

Unlike a typical CBIR interface, where a small number of images estimated to be relevant is shown at once, our interfaces shows all the images on screen. The screen space is allocated proportional to the estimated relevance. This has the advantage that even the less relevant images are easily accessible, so that the user can correct the relevance prediction by selecting an image whose current estimated relevance is low.

The idea of visualizing the relevance predictions as the size of the images is motivated by Dasher, a predictive text-entry system [11]. Dasher shows all letters of the alphabet entering from the right edge of the screen. The letters can be selected by mouse or by gaze. The sizes of the incoming letters are determined by a language model that predicts likelihood of the next letter given letters selected so far. In our case, the sizes of the images are similarly modulated according to their estimated relevance to make it easier to spot other relevant images.

In the remaining of this paper, we introduce our search interface and the relevance prediction model that combines explicit and implicit feedback. Then we report results of a user study, where the dynamic interface was compared to baseline interfaces which do not try to predict the relevance. We also studied how much the relevance prediction performance improves when explicit feedback (mouse clicks) is combined with implicit mouse movement features, such as time duration of hovering over an image.

2 Timeline Browser with a Relevance Estimator

We introduce a browser for finding images that are ordered on a timeline. We utilize the memory of humans: shown images work as recall cues; seeing an image brings back memories from the time the image was taken or seen.

The browser includes a mouse-operated fish-eye lens. When the mouse is moved over an image, the image and its close neighbors are grown to allow inspecting the images more closely. The size of the images is restored when mouse moves away from the images.

The sizes of the images are also affected by their estimated relevance. Relevance is estimated from explicit feedback (mouse clicks) and implicit browsing patterns. Images estimated to be relevant are shown in a larger size. The relevances are recomputed and sizes are updated dynamically after each click. Our hypothesis is that emphasizing the relevant images makes it easier and quicker to find the correct images. Figure 1 shows snapshots of the interface.

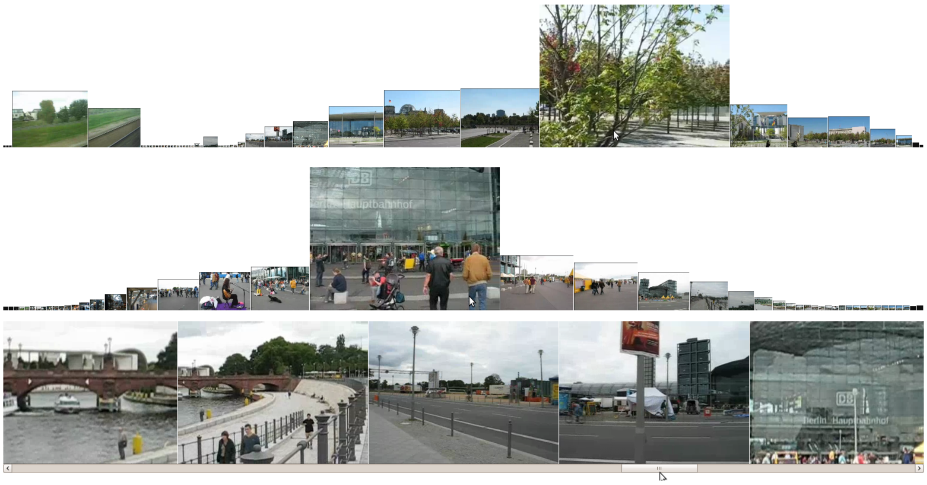

Fig. 1. The three interface variants from the experiments. From top to bottom: our new proactive interface where image sizes are proportional to the estimated relevance, zooming interface that magnifies the image on which the cursor is located and its neighborhood, and a scrollable simple timeline. The images are snapshots from a Creative Commons licensed movie by Adam Wojtanek.

2.1 Relevance Prediction

To be able to emphasize potentially relevant images, we need to estimate the relevance based on clicks and implicit feedback. Next, we will introduce a probabilistic generative model for this purpose.

We assume that relevance is reflected in a latent (potentially multidimensional) variable $\boldsymbol{z}$. The latent variable $\boldsymbol{z}_i$ of image i is assumed to be generated by a linear regression from image's observed content feature vector $\boldsymbol{f}_i$ to the latent space. The values in the unobserved regression matrix $\boldsymbol{Q}$ capture the user's "implicit" query by weighting the content dimensions appropriately. This prior distribution constrains images that are similar with respect to the latent query $\boldsymbol{Q}$ to have similar relevances. We further assume a user model where the image click counts $\boldsymbol{y}$ are drawn from a multinomial distribution. The weights of the multinomial are softmax-normalized values of the latent variables $\boldsymbol{z}$. The plate diagram of the model is shown on the left-hand side of Fig. 2.

We also consider the case where implicit mouse movement feedback, in addition to the clicks, is available for the relevance prediction. The implicit browsing behavior on an image i is encoded as a feature vector $\boldsymbol{x}_i$. We assume that the relevance of an image is related to what is common between the explicit clicks and the implicit mouse movements. Therefore, we consider a model where both the click $\boldsymbol{y}$ and movement features $\boldsymbol{x}$ are generated by the common latent relevance variables $\boldsymbol{z}$. The linear mapping from $\boldsymbol{z}$ to $\boldsymbol{x}$ is a parameterized by a latent matrix $\boldsymbol{W}$. The model structure, where two observed variables are generated by a common latent variable, is similar to the Bayesian CCA (see [6]). This variant of the model is depicted on the right-hand side of Fig. 2.

To summarize, we make the following distributional assumptions:

$$\begin{aligned}
\boldsymbol{z}_i|\boldsymbol{Q},\boldsymbol{f} &\sim N(\boldsymbol{Q}\boldsymbol{f}_i,\sigma^2\boldsymbol{I}) \\
\boldsymbol{x}_i|\boldsymbol{W},\boldsymbol{z} &\sim N(\boldsymbol{W}\boldsymbol{z}_i,\sigma_x^2\boldsymbol{I}) \\
\boldsymbol{y}|\boldsymbol{\alpha},\boldsymbol{z} &\sim \text{Multinomial}([\ldots,\frac{\exp(\boldsymbol{\alpha}^T\boldsymbol{z}_i)}{\sum_j \exp(\boldsymbol{\alpha}^T\boldsymbol{z}_j)},\ldots])
\end{aligned}$$

Columns of $\boldsymbol{W}$ and $\boldsymbol{Q}$ and the vector $\boldsymbol{\alpha}$ are drawn from Gaussian distributions with zero mean and diagonal variance.

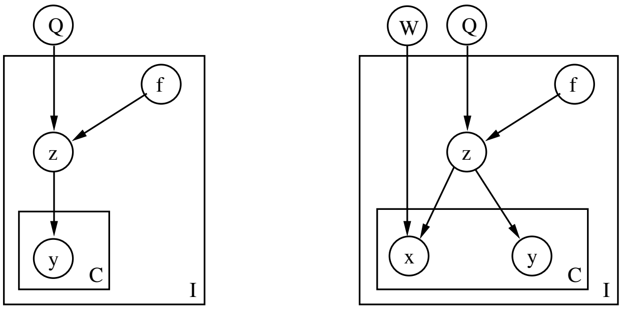

Fig. 2. Plate diagram of the relevance prediction model. The simpler model on the left uses only information about which documents were clicked to give explicit feedback for the prediction. The model on the right includes mouse movement features $\boldsymbol{x}$, as well. The plate I is over the images and the plate C is over the feedback rounds.

We use adaptive MCMC for inferring the relevances. The sampler is initialized from the MAP estimate.

3 Experiments

We tested the model in a video recall task. Three test subjects were asked to view a 17 minute video and, after viewing, recall certain events from it. To aid the recall, the test subjects were able to browse a timeline of snapshots from the video. We recorded the mouse movements and clicks during the browsing. The test subjects completed six recall tasks where they were asked to find certain number of scenes with a specific set of people in specific places. Specifically, they were asked to select one image per suitable shot by clicking it and, after selecting enough images, to shortly describe out loud what they remember about each selected scene. This video recall setup was chosen to simulate retrieval from a personal database, where the content is already somewhat familiar beforehand.

The test subjects were randomly assigned to one of three interface conditions (Fig. 1) in each task. The first condition was the interface described in Sec. 2, which dynamically changes the size of the images to reflect the predicted relevance. We will refer to this interface as *proactive* below. To quantify the effect of altering image saliency during the search, we had a second interface (called *zooming*) which was otherwise identical to the first except that the sizes of the images were not changed in according to the relevance predictions. The third interface (called *scrollable*) was a simple baseline where images were shown in a static size on a scrollable window. The order of the tasks and interface conditions was balanced between the test subjects.

The video was compressed into a series of snapshots taken at 5 second intervals. The timeline, which the users saw, included every second snapshot. The rest of the snapshots, which were not shown, were held out as a test set. There were 104 and 105 images in the timeline and hold out sets, respectively.

The image content was encoded as binary indicators of specific people, objects and locations in the image. For this experiments, the features were constructed by manually tagging the images. Similar features could be constructed without manual tagging using face and object recognition algorithms with the expense of a lower accuracy. For example, in lifelogging type of applications a wearable recording device can recognize people and objects in the image [1].

3.1 Comparing the Interfaces

An interface is efficient if it allows a quick access to the relevant images without having to view too many non-relevant images. We measure the effort required to find the relevant images among the non-relevant ones using the standard information retrieval measure of mean average precision. We consider the viewed images as an ordered list, where the clicked images are labeled as positive and all others as negative. The average precision is the average of the precisions computed at each positive rank. In the proactive and zooming interfaces, an image is considered viewed when the user moves the mouse over it. In the scrollable interface, an image is considered viewed when it is scrolled into the view.

Figure 3 shows the mean average precisions for the three interfaces in each task. We are interested in comparing the zooming and the proactive interfaces, which behave equally until the first click. Therefore, the figure displays values computed after discarding all views up to and including the first clicked image.

The proactive interface attains higher mean average precision than the zooming interface in all tasks. This shows that modulating the saliency according to the relevance estimates helps the users to locate the relevant images faster.

The performance of the scrollable interface has the largest variance between the tasks. In task number 2 it is clearly better than the alternatives. The reason is that the scrolling interface is initially showing the first images on the timeline and there happened to be a few scenes which are relevant in the second task, in the beginning of the video. In five tasks the scrollable interface is the worst by a large margin. They correspond to the typical case where one has to scroll through many images before finding relevant ones, whereas in the other two interfaces it is possible to easily skip over a sequence of images.

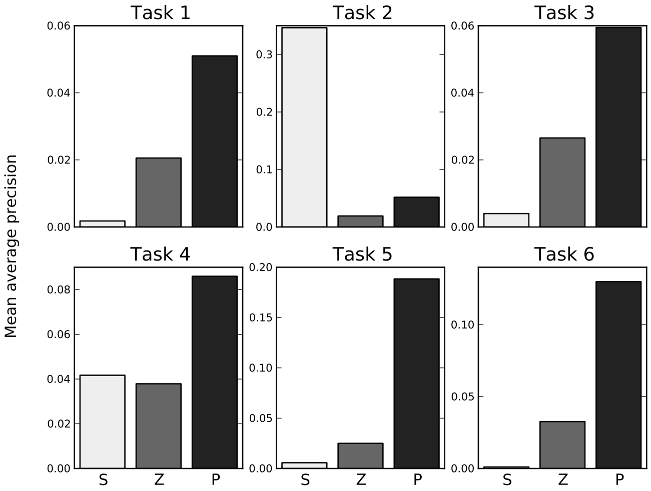

Fig. 3. Mean average precisions in image viewing sequences after the first click for the scrolling (S), zooming (Z) and proactive (P) interfaces. Higher values mean that fewer non-relevant images were viewed between the relevant (clicked) images. The proactive interface is better in all tasks than the zooming interface, which is identical except for the prediction.

3.2 Integrating Explicit and Implicit Feedback

We also study if implicit feedback from the mouse movement patterns can improve inference over the explicit clicks. We encode the implicit feedback as a feature vector for each image. The features are listed in Table 1. The evidence from the explicit and implicit feedback is combined using the model from Sec. 2.1. We compared the performances of the combined feedback and the explicit-only models. They were trained using data collected with the zooming and proactive interfaces. The data from the scrollable condition is discarded because, in that condition, the mouse is mostly used to interact with the scrollbar and less with the images.

Table 1. Implicit mouse movement features designed for capturing the browsing patterns.

Description	Type
Number of visits to the image	Integer
Total hover duration on the image (ms)	Continuous
Was this image already visited during the last 15 visits?	Binary
Both left and right neighbors visited during the last 4 visits	Binary
Average hovering duration on the previous 2 images (ms)	Continuous

To select the dimensionality of the latent variable $\boldsymbol{z}$ we train models of different dimensionality on the observations excluding the last click of a task, and compare the log-likelihoods when predicting the last click. The log-likelihoods averaged over the tasks are plotted in Fig. 4. If the dimensionality is too low (less than 6), the model is not flexible enough to model both the features $\boldsymbol{x}$ and the clicks $\boldsymbol{y}$, and hence the performance of the combined feedback variant is much worse than that of the explicit-only model. When the dimensionality is 6, including the implicit feedback improves the performance slightly. Combined feedback model with $\dim(\boldsymbol{z}) = 8$ has the best log-likelihood. In summary, the performance is improved when the implicit feedback is combined with explicit feedback.

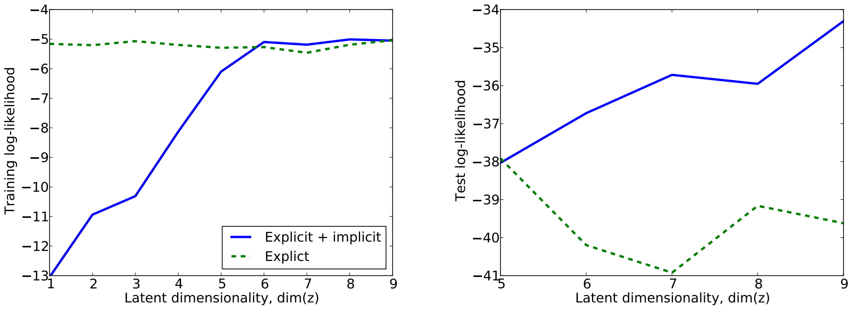

Fig. 4. Average log-likelihood of the clicks $\boldsymbol{y}$ for training (on left) and testing (on right) sets as a function of $\dim(\boldsymbol{z})$, the dimensionality of the latent variable. Larger values mean better prediction performance.

4 Discussion

We have introduced an image browser that modulates the saliency (size) of images according to their predicted relevance. The relevance is estimated online by observing explicit and implicit mouse interaction patterns. We performed a small scale user study where we showed that changing the image size in proportion to the relevance predictions helps in finding the relevant images with less effort. We also showed that complementing explicit feedback with implicit mouse movement patterns improves the relevance prediction further.

Acknowledgements. The authors belong to Adaptive Informatics Research Centre at Aalto University. This work has been funded by Aalto MIDE programme (project UI-ART) and by the PASCAL2 Network of Excellence, ICT 216886.

References

1. Ajanki, A., Billinghurst, M., Gamper, H., Järvenpää, T., Kandemir, M., Kaski, S., Koskela, M., Kurimo, M., Laaksonen, J., Puolamäki, K., Ruokolainen, T., Tossavainen, T.: Contextual information access with augmented reality. In: Proc. MLSP 2010. pp. 95–100. IEEE Press, New York (2010)
2. Datta, R., Joshi, D., Li, J., Wang, J.Z.: Image retrieval: Ideas, influences and trends of the new age. ACM Comput. Surv. 40, 5:1–5:60 (2008)
3. Dumais, S., Cutrell, E., Cadiz, J., Jancke, G., Sarin, R., Robbins, D.C.: Stuff I've seen: a system for personal information retrieval and re-use. In: Proc. SIGIR 2003. pp. 72–79. ACM Press, New York (2003)
4. Faro, A., Giordano, D., Pino, C., Spampinato, C.: Visual attention for implicit relevance feedback in a content based image retrieval. In: 6th Symposium on Eye-Tracking Research & Applications. pp. 73–76. ACM Press, New York (2010)
5. Gemmel, J., Bell, G., Lueder, R.: MyLifeBits: a personal database for everything. Commun. ACM 49, 88–95 (2006)
6. Klami, A., Kaski, S.: Probabilistic approach to detecting dependencies between data sets. Neurocomput. 72, 39–46 (2008)
7. Kozma, L., Klami, A., Kaski, S.: GaZIR: Gaze-based zooming interface for image retrieval. In: Proc. The Eleventh International Conference on Multimodal Interfaces and The Sixth Workshop on Machine Learning for Multimodal Interaction (ICMI-MLMI). pp. 305–312. ACM Press, New York (2009)
8. Oyekoya, O., Stentiford, F.: Perceptual image retrieval using eye movements. In: Zheng, N., Jiang, X., Lan, X. (eds.) IWICPAS 2006. LNCS, vol. 4153, pp. 281–289. Springer, Heidelberg (2006)
9. Puolamäki, K., Ajanki, A., , Kaski, S.: Learning to learn implicit queries from gaze patterns. In: Proc. ICML 2008. pp. 760–767. Omnipress, Madison (2008)
10. Puolamäki, K., Salojärvi, J., Savia, E., Simola, J., Kaski, S.: Combining eye movements and collaborative filtering for proactive information retrieval. In: Proc. SIGIR 2005. pp. 146–153. ACM press, New York (2005)
11. Ward, D.J., MacKay, D.J.: Fast hands-free writing by gaze direction. Nature 418, 838 (2002)