M.I.T Media Laboratory Perceptual Computing Section Technical Report No. 382
Also appearing: IEEE Int. Conf. on Image Proc., Lausanne, Sep. 1996.

Modeling user subjectivity in image libraries

Rosalind W. Picard, Thomas P. Minka, Martin Szummer

MIT Media Laboratory, 20 Ames St., Cambridge, MA 02139
picard@media.mit.edu, http://www.media.mit.edu/~picard/

Abstract

In addition to the problem of *which* image analysis models to use in digital libraries, e.g. wavelet, Wold, color histograms, is the problem of *how* to combine these models with their different strengths. Most present systems place the burden of combination on the user, e.g. the user specifies 50% texture features, 20% color features, etc. This is a problem since most users do not know how to best pick the settings for the given data and search problem. This paper addresses this problem, describing research in progress for a system that (1) automatically infers which combination of models best represents the data of interest to the user and (2) learns continuously during interaction with each user. In particular, these two components – inference and learning – provide a solution that adapts to the subjective and hard-to-predict behaviors frequently seen when people query or browse image libraries.

1 Introduction

The earliest systems designed for image retrieval (see [1] for several descriptions) and those that have become commercially available, tend to follow a basic paradigm: (1) precompute features or model parameters for each image, (2) have the user specify which models or ranges of parameters are most important, and (3) have the user select example images to initiate a query. The system then compares the user's query information with all the stored information, and retrieves images it thinks are "similar" according to the constraints specified during step (2).

This basic paradigm is useful in limited data sets and search problems, provided that the user is an expert in how the underlying image similarity processing works. However, it is not suitable for general use. The average person looking for images does not know how to choose model parameters as required in step (2). Moreover, as combinations of models (e.g. multiple color and texture models) become available, the choice of parameters is non-intuitive even for the expert image processing researcher. In short, new image analysis tools are needed that perform model selection and combination. Ideally, the tools work rapidly, so the user can iterate interactively with the system, refining his or her request online.

Additionally, a user should be allowed to be subjective – to give, over time, the same set of imagery different labels, or to give the same labels to different content, e.g. to the category of images "they like." It is desirable that the system be able to adapt itself continuously to the changing requests of the user, e.g. to *learn* how to model mappings between the image data and its labels based on changing feedback from the user.

One of the most challenging test scenarios is when the desired image contents are hard to describe objectively. In the solutions we are researching, the users do not have to select model parameters, but simply choose example images that they like. Fig. 1 illustrates a case of two users trying to find more images they like in a Picasso art database of 320 paintings. Each user selects a few example images, then the system analyzes the characteristics of the examples and retrieves other similar images from the database. The burden is on the system to infer how to measure similarity.

In the figure, User 1 gives two examples of textured, cubist paintings of different colors. The system infers that color is not relevant, and searches for images with similar texture (using a multiscale simultaneous autoregressive texture model from [2]). User 2 also gives two examples. The first image is identical to that of user 1, but the second has a different texture and the same color. In this case, the system determines that color is important and retrieves other images with similar colors (using Euclidean distances on 256-bucket color histograms from the decorrelating color space of [3]). The browser can also combine texture and color for one query, or choose combinations of other available similarity models. To refine the query results, the user simply gives additional examples. This is called "relevance feedback" in the information retrieval community.

It is important to note that modeling subjectivity is an objective problem. The labelings or categories chosen by subjective users result in objective groupings of data over which the performance of the system can be tested objectively.

The next section describes a method of inference for models and their combination. Sec. 3 describes our research on a learning algorithm that addresses the problem of generalization, i.e. taking knowledge learned from one problem and using it to solve another.

2 Model inference and combination

For years many researchers (including the authors) have assumed that there would be "one best" model for solving the problem of perceptual similarity, or image similarity. Working in the area of content-based image retrieval has changed our thinking in this regard. Although there are searches on limited domains where one model may always be best, in general we think the one-model solution will be too brittle, and a relatively small set of models (less than a dozen) will give the best performance.

Of course, which models these should be remains dependent on the data and what is to be done with it. In the Picasso example above, the particular texture model is good at grouping some of Picasso's cubist paintings, but poor at

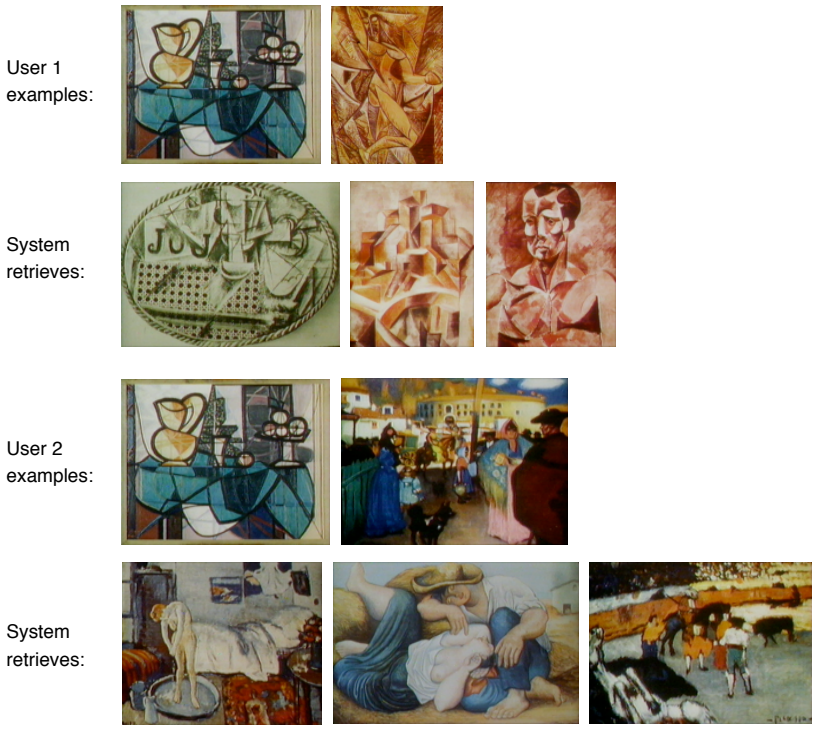

Figure 1: Different users select different positive examples for the category "favorite Picasso images." Image analysis aims to infer underlying features common to the user-defined category, and then use these features to predict other images of interest to the user. The inference is always based on the present set of positive and negative examples given by a user.

grouping portraits. Another model, or combination of models, might perform better still.

With no claims of starting with the best models, but with evidence that combining suboptimal ones can outperform a single one [4], we describe the following method for making combinations which can improve joint performance.

We have explored many ways for combining models. Initially, we considered direct linear combinations of model features – the traditional approach. However, concatenating model features causes an exponential growth in the space used to represent the data, and has a variety of other problems, especially the problem when features from one model are of a different scale than features from another model, and simple re-scaling of them destroys their discrimination properties [5]. To date, the most successful combination method we have found (for avoiding the scaling and dimensionality problems, and for running in interactive time) is based on quantization of the feature spaces followed by a learning algorithm, such as set cover [4].

The model features or parameters are used only initially during quantization, which is represented as hierarchical trees. (The trees provide an organization of the data which may also be interesting for the user to browse.) The use of trees provides not only a searching efficiency advantage, but also takes care of the problem that there are "many ways to segment an image." People group image contents differently – that is a manifestation of subjectivity and of differing goals. With the tree representation, different segmentations can be made by simply choosing a different partition of the nodes. The hierarchical trees are representations that make segmentations, as opposed to a representation that is a fixed segmentation. Once the user is in the system loop, the system can decide which of its possible segmentations best suits the user's desires.

Once the trees are constructed, the similarity problem changes from one of metric distances on image features to one of distances in a hierarchy-induced metric. Different model parameter ranges and dimensionalities cease to be an issue. The set-cover combination method then proceeds by looking for the simplest set of nodes that covers all the user's positive examples and none of their negative examples. Additional criteria can also be added depending on the domain, and performance can be improved by adding the ability to learn these criteria. (See [4] for details.)

An example of using this method, starting with three models (and a database of only six elements) is shown in Fig. 2.

The inference method presently acts on both positive and negative examples. A limitation is that the user cannot yet give feedback such as, "I don't like this particular spatial arrangement of these colors." This is an area for continued development.

The current inference/combination method processes about 5 examples per CPU second on an HP 735/99 using a database with thousands of leaf elements and about a half dozen models. Details on its complexity are in [4]. While we generally don't like to impose speed requirements on research algorithms, the ability to perform model selec-

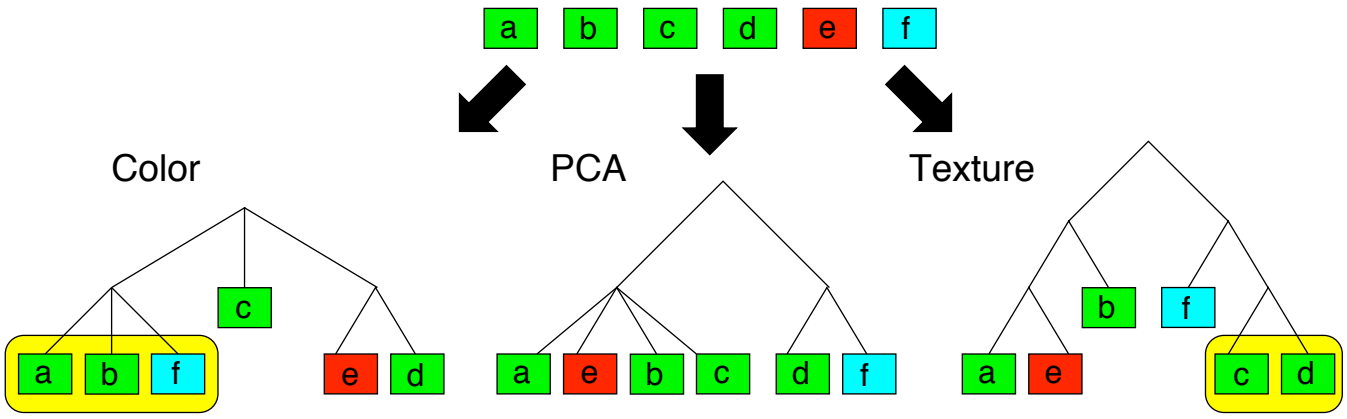

Figure 2: Method for combining multiple models. Each model constructs a tree of possible segmentations. The user provides examples: (a, b, c, d are positive, e is negative.) The system chooses tree nodes which most efficiently describe the positives and not the negatives, resulting in the two groupings shown (shaded). The result is a combination of color and texture to characterize the user's examples. Note that example f is inferred as being of interest to the user.

tion interactively is a benefit in a retrieval system, where the formulation of a query is well-suited to an iterative process of providing a few examples, seeing what is retrieved, modifying the example set, and so forth. This ability obviates the need for the user to "think of everything" before posing a query. In reality, users often modify what they want after seeing more of the database.

3 Learning and generalization – modeling subjectivity

After you show somebody how to do something, and you indicate that they are doing it to your satisfaction, you assume they have "learned" this problem, and will remember its solution (within reason) if they confront it again. Most current retrieval systems, however, have no such memory. When people repeatedly ask similar queries of them, the system appears to be "stupid" because it doesn't learn.

Therefore, on top of the set-covering algorithm, we've added a dynamic bias. The *bias* of a learner is defined to be any basis for choosing one generalization over another, other than strict consistency with the training examples [7]. Having the right bias is crucial to successful learning, especially when a small number of examples (as desired in an interactive setting) leaves open many possible solutions. The "FourEyes" browser improves its bias over time [4]. When the system sees a problem similar to one it has seen before, it automatically switches to the bias that it learned for that problem. When it sees a significantly new problem, FourEyes learns a new bias. It therefore behaves differently over time, depending on what it has been exposed to.

FourEyes has three stages that learn at different rates, from interactive-speed online learning, to longer-term offline learning, the latter of which is analogous to human "reflection" or "dreaming" in its abilities to process image information over a broader scope.

One of the critical tests of a learning system is how well does it generalize? Traditional image processing has been concerned with generalization from a training set to a test set. The problem in image retrieval systems is that the same test set might have more than one "true" interpretation, depending on what the user wants at the moment (or what the next user wants, after the system has adapted to the present user.) The user's subjectivity, in the form of changing feedback, creates a signal processing problem analogous to non-stationary signal detection, where the category of signals you are trying to detect, e.g., "images you like," may change its signature in time. But they also may not change. The difficulty is to track the changes, while preserving performance on the parts that do not change.

The system tracks the changes with online clustering of the bias, represented as weights on the tree nodes. (These weights are used in the set-cover process.) Another aspect of the bias is the shape of the trees, corresponding to the quantization of the space. This is also learned during interaction with the user. (See [4] for details.)

We have applied a test of generalization to FourEyes, to evaluate how its learning mechanism performs on problems it hasn't seen before. Each different "problem" can be thought of as a labeling in a user's head – all users label the same data, but do so differently, or one user may do so differently over time. A test of FourEyes' performance on such a task is shown in Fig. 3.

Fig. 3 simulates ten different learning problems on $N = 1008$ images in the Brodatz Database. Each problem has two labeling categories of size 504 images. The ten problems are similar in that each category is a random merging of the 112 9-image classes of Brodatz patterns (112 classes x 9 images/class = 1008 images total; 56 classes were merged to form each category). Although the categories were not chosen by people, they simulate categories that might have been chosen by people.

Training on only one problem and testing on that same problem gives the best performance – the "ideal" points shown in Fig. 3. Generalization is more challenging – involving training on one problem, and testing on another.

The details of the learner and of this particular general-

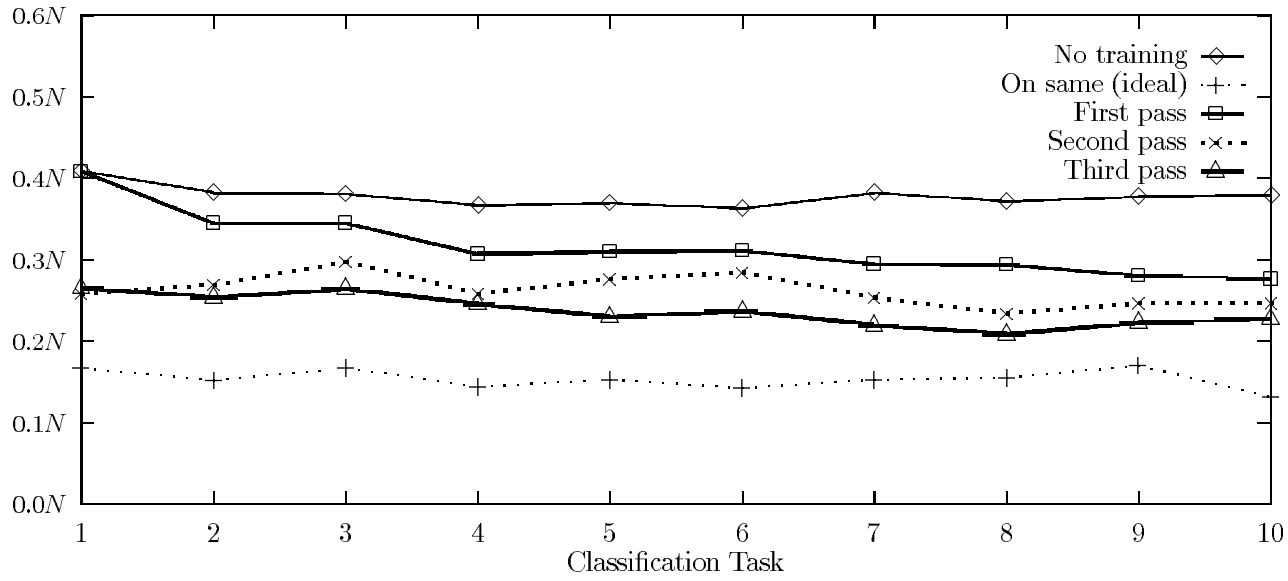

Figure 3: Test of generalization on ten similar problems. The vertical axis is the learning time, measured in terms of how many examples had to be provided before the problem was solved. The "no training" and "ideal" curves are the results of running the learner once, and then again on the same problem. The other curves are the result of sequential training – running on problem 1, then problem 2, etc. They tilt slightly downward to the right, indicating the presence of generalization.

ization study are in [4], which also contains additional evaluations. To summarize the results here, the curves above indicate that the time needed to learn decreases from left to right, showing that the learner gains performance on the later problems, even before having seen them. For this evaluation, training on the first nine problems is half as good as training on the tenth by itself.

4 Summary

This paper has highlighted results from our recent research focusing on image analysis for (1) model inference and combination, and (2) learning for generalization. The underlying premise is that a subjective human is in the loop with the image analysis and retrieval system. Not only is the human user unlikely to know how to set all the model parameters optimally, but his or her subjectivity leads to the same data needing to be treated in different ways.

A method was described for automatically choosing combinations from multiple image models. This releases the human from the task of adjusting image features or model parameters. Instead, the human interacts with the system by providing a stream of positive and negative examples.

An example was provided of a learning system with the ability to generalize what it has learned across new problems. This is intended to simulate training after interacting with one user, and then having to perform well while working with another user or with the same user who is behaving differently over time. To a user, this behavior would make the system appear "smarter and faster" with increasing use.

5 Acknowledgements

This work was sponsored in part by BT, P.L.C., by Hewlett-Packard Research Labs, by Interval Research Corp., by NEC, and by the Television of Tomorrow Consortium.

6 References

[1] B. Furht, S. W. Smoliar, and H.-J. Zhang, *Video and Image Processing in Multimedia Systems*. Kluwer Academic Publishers, 1995.

[2] J. Mao and A. K. Jain, "Texture classification and segmentation using multiresolution simultaneous autoregressive models," *Patt. Rec.*, vol. 25, no. 2, pp. 173–188, 1992.

[3] Y.-I. Ohta, T. Kanade, and T. Sakai, "Color information for region segmentation," *Comp. Graph. and Img. Proc.*, vol. 13, pp. 222–241, 1980.

[4] T. P. Minka, "An image database browser that learns from user interaction," Master's thesis, MIT, Cambridge, MA, February 1996. EECS.

[5] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. Wiley-Interscience, 1973.

[6] T. P. Minka and R. W. Picard, "Interactive learning using a 'society of models'," *Pattern Recognition*, 1996. To appear. Also appears as MIT Media Lab Perceptual Computing TR#349.

[7] T. M. Mitchell, "The need for biases in learning generalizations," Computer Science CBM-TR-117, Rutgers University, New Brunswick, NJ, May 1980.