

A KNOWLEDGE ENGINEERING APPROACH FOR IMAGE CLASSIFICATION BASED ON PROBABILISTIC REASONING SYSTEMS

Seungyup Paek and Shih-Fu Chang

Columbia University, 1306 S.W. Mudd, 500 W. 120th St., Department of Electrical Engineering
New York City, N.Y. 10027, U.S.A
syp@ee.columbia.edu, sfchang@ee.columbia.edu

ABSTRACT

We present a knowledge engineering approach for image classification that is based on probabilistic reasoning systems. The approach gives the knowledge engineer a systematic way to integrate multiple probabilistic classifiers. A case study of applying this approach to the domain of consumer photographs is presented.

1. INTRODUCTION

Based on our experience in a previous Web image search engine, WebSEEK, we found that subject hierarchy browsing is a popular user operation in interactive image retrieval [1]. Image classification systems can be used to map images to specific classes in the subject hierarchy.

We introduce a knowledge engineering approach for image classification that is based on probabilistic reasoning systems. The approach gives the knowledge engineer a systematic way to integrate multiple probabilistic classifiers for a given domain. This is essential in building image classification systems for two reasons.

Firstly, for a given image classification problem, a variety of classifiers based on information of different modalities (visual, textual) and different techniques can be developed. The proposed approach provides a systematic way to integrate multiple classifiers developed for a given image classification problem.

Secondly, for a given domain, several related image classification problems can be defined. The proposed approach provides a systematic way to integrate classifiers developed for different image classification problems, in order to exploit interactions that may exist between classes.

The paper is organized as follows. Firstly, we briefly describe how to capture uncertain knowledge in a natural and efficient way by using a belief network. The belief network data structure is the basis for a probabilistic reasoning system. Secondly, we overview the block matching classifier, which we will use as a component of the probabilistic reasoning system. Finally, we present a knowledge engineering approach to designing image classification systems that is based on the probabilistic reasoning system. We conclude by giving a case study in which we apply the approach to building an image classification system for the domain of consumer photographs.

2. PROBABILISTIC REASONING SYSTEMS

In this section, we begin by showing how to capture uncertain knowledge in a natural and efficient way by using a data structure called a belief network. Belief networks form the basis of a

probabilistic reasoning system [5]. In section 4, we will show how probabilistic reasoning systems form the basis of a knowledge engineering approach for image classification.

2.1. Representing Knowledge in an Uncertain Domain

Probability theory shows us that the joint probability distribution of a given domain can be used to answer any question about a domain. However, the joint probability distribution can become intractably large as the number of variables grows. To deal with this problem, we can use a data structure called a *belief network* to give a concise specification of the joint probability distribution of a set of random variables for a given domain.

Belief networks greatly reduce the number of conditional probabilities that need to be specified. This is possible by using Bayes' rule to exploit conditional independence relationships among random variables.

For image classification problems, the random variables of a given domain are used to characterize the image classes and the outputs of image classifiers. This will be discussed in detail in section 4. In the remainder of this section, we will present the structure and semantics of belief networks.

A belief network is a graph in which the following conditions hold. Firstly, a set of random variables makes up the nodes of the network. Secondly, a set of directed links connect pairs of nodes. The intuitive meaning of an arrow from node X to node Y is that X has a direct influence on Y . In this case, X is called the parent of Y . Thirdly, each node has a conditional probability table (CPT) that quantifies the effects that the parents have on the node. Fourthly, the graph has no directed cycles. It is a directed a-cyclic graph.

2.2. Semantics of Belief Networks

To explain the semantics of a belief network, we first start with the claim that a belief network that is properly constructed for a domain provides a complete and concise description of the domain. Every entry in the joint probability distribution can be calculated from the information in the network. A generic entry in the joint is the probability of a conjunction of particular assignments to each variable, such as $P(X_1 = x_1 \wedge \dots \wedge X_n = x_n)$. We use the notation $P(x_1, \dots, x_n)$ as an abbreviation for this. Given a belief network, the value of this entry is defined to be given by the following formula:

$$P(x_1, \dots, x_n) = \prod_{i=1}^n P(x_i \mid \text{Parents}(X_i)) \quad (1)$$

Thus, each entry in the joint is represented by the product of the appropriate elements of the conditional probability tables (CPTs) in the belief network. The CPTs therefore provide a decomposed representation of the joint.

Equation 1 defines what a given belief network means. It does not, however, explain how to construct a belief network such that the resulting joint distribution is a good representation of a given domain. Equation 1 implies certain conditional independence relationships that can be used to guide the knowledge engineer in constructing the topology of the network. First, by using the definition of conditional probability, we can rewrite the joint in the following form:

$$P(x_1, \dots, x_n) = \prod_{i=1}^n P(x_i | x_{i-1}, \dots, x_1) \quad (2)$$

Comparing this with Equation 1, we see that the specification of the joint is equivalent to the general assertion that

$$P(\mathbf{X}_i | \mathbf{X}_{i-1}, \dots, \mathbf{X}_1) = P(\mathbf{X}_i | \mathbf{Parents}(\mathbf{X}_i)) \quad (3)$$

Provided that $Parents(X_i) \subseteq \{x_{i-1}, \dots, x_1\}$. This last condition is easily satisfied by labelling the nodes in any order that is consistent with the partial order implicit in the graph structure. What the preceding equation says is that the belief network is a correct representation of the domain only if each node is conditionally independent of its predecessors in the node ordering, given its parents. Hence, in order to construct a belief network with the correct structure for the domain, we need to choose parents for each node such that this property holds. Intuitively, the parents of node X_i should contain all those nodes in X_1, \dots, X_{i-1} that directly influence X_i .

The general procedure for incremental network construction is shown below. This will form the core of the knowledge engineering approach to developing image classification systems (section 4).

1. Choose the set of relevant variables X_i that describe the domain.
2. Choose an ordering for the variables.
3. While there are variables left, pick a variable X_i and add a node to the network for it. Set $Parents(X_i)$ to some minimal set of nodes already in the net such that the conditional independence property (3) is satisfied. Finally, define the conditional probability table for X_i .

Even in a locally structured domain, constructing a locally structured belief network is not a trivial problem. We require not only that each variable is directly influenced by only a few others, but also that the network topology actually reflects those direct influences with the appropriate set of parents. Because of the way the construction procedure works, the “direct influencers” will have to be added to the network first if they are to become parents of the node they influence. Therefore, the correct order to add nodes is to add the “root causes” first, then the variables they influence, and so on until we reach the “leaves,” which have no direct causal influence on other variables.

As well as being a complete and non-redundant representation of the domain, a belief network can often be far more compact than the full joint. This property is what makes it feasible to handle a large number of pieces of evidence without the exponential growth in conditional probability values.

2.3. Inference in Belief Networks

The basic task for any probabilistic inference system is to compute the posterior probability distribution for a set of query variables, given exact values for some evidence variables. That is, the system computes $P(\mathbf{Query} | \mathbf{Evidence})$. Of course, belief networks are flexible enough so that any node can serve as either a query or an evidence variable. There has been a wealth of research to design efficient inference mechanisms for answering queries given a belief network. The development of these algorithms are discussed in [5].

3. COMPONENTS

In the previous section, we showed how to capture uncertain knowledge in a natural and efficient way by using a belief network. The belief network data structure is the basis for a probabilistic reasoning system. We mentioned that for image classification problems, the random variables characterize image classes and image classifiers. In this section we present the block matching classifier, first proposed in [8], as one example of an image classifier that can be used as a component of the probabilistic reasoning system. Other classifiers that can be used with the approach presented in this paper are presented in [4, 6, 7, 9]. In section 4 we will show how the classifier and the probabilistic reasoning system are used in the knowledge engineering approach to building image classification systems.

In the block matching classifier, all the images of the training and test sets are first divided into a set of 8×8 block regions. For each block region, we generate a set of color and texture related visual features, described below. In our work, we are using the HSV color histograms [2] as the color feature, and the edge direction histogram as the texture feature [6]. All the images in the training set are manually labeled as belonging to one of a set of classes.

The HSV color representation is attractive because it represents with equal emphasis the three color attributes that correspond to the human perception of colors: hue (H), saturation (S), and value (V). Value corresponds to the brightness of a color. For each block of an image, a histogram is computed by recording the number of occurrences of each quantized HSV color in the pixels of the block.

To generate the edge direction histogram, an edge detection algorithm is used to extract the edges in a block. For the histogram, a total of 73 bins are used to represent the edge direction histogram of an image. The first 72 bins are used to represent edge directions quantized at 5 degree intervals and the last bin represents a count of the number of pixels that did not contribute to an edge.

For an image to be classified, we compare each block of the image to all the blocks of the images in the training set and compute the k nearest neighbor blocks. This is done for each visual feature by using the histogram intersection similarity measure. Each of the nearest neighbor blocks from the training set are labeled with a class. After doing this for all the blocks of the image to be classified, we take a majority vote of the classes in the nearest neighbor blocks.

4. KNOWLEDGE ENGINEERING APPROACH

In this section we will present a knowledge engineering approach for image classification based on the probabilistic reasoning system presented in the previous sections. In the following section,

we give a case study of how this knowledge engineering approach can be used to build an image classification system. The knowledge engineering approach is outlined in the following steps.

4.1. Decide on the Classes

For a given domain, we decide on the classification problems to be solved. For each classification problem we also decide on the classifiers to be used for the problem. A classification problem is characterized by a set of classes c_1, c_2, \dots, c_n . One or more classifiers can be employed for a given classification problem. The goal of a classifier is determine the correct class $C \in c_1, c_2, \dots, c_n$ for each image.

Deciding on the classification problems to be solved for the domain is guided by two principles. First, we have to determine the classes that are *useful* and *interesting* for a given domain. This depends on an understanding of the domain and of how the images of a given domain will be used in different applications. Secondly, we have to understand the capabilities and tradeoffs of different classifiers. This depends on an understanding of how each classifier works and what kinds of classification problems they have been used to solve successfully.

4.2. Decide on a Vocabulary of Random Variables

We define a boolean random variable X_i for each classification problem defined for the domain. We also define a random variable Y_j for each classifier j that is used in the domain. The values that X_i and Y_j can take on correspond to the classes associated with the classification problem.

4.3. Encode General Knowledge

We encode general knowledge about the dependence between the different classes by constructing a belief network for the domain. Constructing a locally structured belief network not only requires that each random variable is directly influenced by only a few others, but also that the network topology actually reflects those direct influences with the appropriate set of parent random variables.

Because of the way the construction procedure works, as discussed in section 2, the “direct influencers” will have to be added to the network first if they are to become parents of the node they influence. For image classification systems, the random variables corresponding to classifiers are always children nodes of the associated classification problem node.

There are domains in which each image class can be influenced directly by all the others. The reduction in information that occurs in practice comes about because real domains have a lot of structure, which networks are very good at capturing. In some domains, there will be slight dependencies that should strictly be included by adding a new link. But if these dependencies are very tenuous, then it may not be worth the additional complexity in the network for the small gain in accuracy.

After the qualitative part where we have specified what classes are dependent on what others, there is a quantitative part, where we specify the Conditional Probability Tables (CPTs) of each node in the belief network. The values can come either from the knowledge engineer’s subjective experience, or from measurements of frequencies in a database of training images, or from some combination of the two.

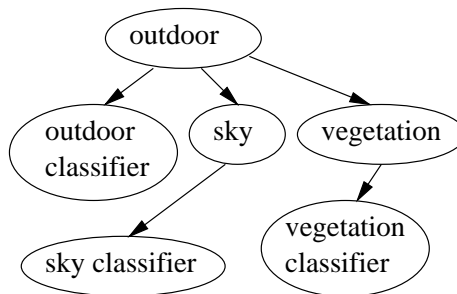


Figure 1: Initial Belief Network for Consumer Photographs

5. CASE STUDY: CONSUMER PHOTOGRAPHS

In this section, we present a system for the classification of consumer photographs. Figure 1 shows the belief network for this domain. In the domain of consumer photographs, indoor and outdoor classes are an intuitive way to organize the photographs. Examples of the indoor and outdoor photographs can be seen in the *inLumine* image classification system online demo [3]. The sky and vegetation classes are not as likely to be used by users in browsing photographs, but may be used for refined searches.

For training and evaluation, a set of 1708 consumer photographs from Kodak were labeled into a variety of image classes on the basis of human judgements. Each photograph was labeled as being *indoor* or *outdoor*, *sky* or *no sky* and *vegetation* or *no vegetation*. Table 1 summarizes the performance evaluation.

First, all the images were labeled as indoor or outdoor by building an indoor-outdoor block matching classifier based on leave-one-out cross validation, to maximize statistical accuracy. In this method, each image is classified by using all the remaining images as the training data. This process was repeated by building a sky block matching classifier and a vegetation block matching classifier. At the end of this process, each image is automatically labeled as being *indoor* or *outdoor*, *sky* or *no sky* and *vegetation* or *no vegetation*.

Second, the set of all images was divided into two sets, set_a and set_b . A belief network, bn_a was constructed by using the statistics of set_a . In other words, the conditional probability tables of bn_a were based on the labels of the images in set_a . Similarly, a belief network, bn_b was constructed by using the statistics of set_b . Afterwards, belief network bn_a was used to classify the images in set_b , and belief network bn_b was used to classify the images in set_a (two-fold cross validation for the performance evaluation of the overall system).

Based on two-fold cross validation of the overall system, the image classification system based on combining multiple classifiers in a belief network gave a classification accuracy of 84.54%, for an improvement of 1.5%. In other words, for the set of all images, the belief network based classifier had 25 fewer errors than the results of the indoor-outdoor block matching classifier. For the vegetation-no vegetation classifier, there was an improvement of 3.3% by using the belief network, as opposed to the block matching classifier alone.

Finally, a belief network bn was constructed by using all the images. This belief network was used to classify all the images. This was done to approximate leave-one-out cross validation for the evaluation of the overall system, which is currently being implementing.

| Classifier | Single Classifier | Integrated (2-fold) | Integrated (approx. of leave-1-out) |
|-------------------|-------------------|---------------------|-------------------------------------|
| <i>Outdoor</i> | 83.08 % | 84.54 % | 86.30 % |
| <i>Sky</i> | 78.69 % | 78.00 % | 79.39 % |
| <i>Vegetation</i> | 71.14 % | 74.40 % | 76.17 % |

Table 1: Performance evaluation

Based on this approximation of leave-one-out cross validation of the overall system, the image classification system based on combining multiple classifiers in a belief network gave a classification accuracy of 86.3%, for an improvement of 3.2%. In other words, for the set of all images, the belief network based classifier had 55 fewer errors than the results of the indoor-outdoor block matching classifier. For the vegetation-no vegetation classifier, there was an improvement of 5.0% by using the belief network, as opposed to the block matching classifier alone.

It is important to state that the classification accuracy we report for the indoor-outdoor block matching classifier is significantly lower than the results reported in [8], for the same set of Kodak consumer images. Essentially, the Kodak photographs consist of photos coming from different rolls of film. We found that within a roll, there were many photographs of the same scene, with slight variations in perspective. If we do not account for this “duplication”, the block matching classifier obtains approximately 90% classification accuracy, as reported in [8]. However, taking this into account, we obtained the significantly lower classification accuracies.

6. RELATED RESEARCH

[10] first proposed a bayesian framework for semantic content classification. The framework was used to classify approximately 100 video clips of a movie “Circle of Friends” into the following classes *action* (e.g. whether a shot contains a significant amount of action), *close up*, *crowd* and *set* (e.g. nature vs. suburban).

As we described in section 3, [8] present a block matching classifier that classifies indoor and outdoor images on the basis of color and texture features. [6] also present a system that uses naive bayesian classification to classify images as indoor or outdoor, on the basis of color and texture features. Furthermore, they also classify images into other semantic classes, such as city/suburb vs. nature landscape.

[7] use a multi-stage system to classify images into several classes, sequentially assigning images to *type* (e.g., color graphics, black and white), *domain* (e.g., center surround, silhouette) and *semantic* classes (e.g., beach, buildings, nature, sunsets). Image semantics were determined by a novel system which matches the arrangements of regions in the images to composite region templates.

For restricted classification, [4] detect naked people in an image, using a representation (*body plan*) for people and animals, which is adapted to segmentation and to recognition in complex environments. The representation is an organized collection of grouping hints obtained from a combination of constraints on color and texture and constraints on geometric properties such as the structure of individual parts and the relationships between parts.

There are two crucial difference between these systems and the image classification system that we propose. Firstly, for a

given image classification problem, our approach allows us to systematically integrate a variety of classifiers. Secondly, for a given domain, our approach allows us to systematically integrate classifiers that are developed for different image classification problems, maximizing on interactions that may exist between different image classes.

7. CONCLUSION AND FUTURE WORK

We have presented a knowledge engineering approach for image classification based on probabilistic reasoning systems. Firstly, for a given image classification problem, our approach allows us to systematically integrate multiple classifiers. Secondly, for a given domain, our approach allows us to systematically integrate classifiers for image classification problems, to exploit interactions between different image classes. The approach provides an integration framework

In our current research, we are applying this approach to other domains, in particular, we are conducting extensive research on the news photograph domain, in which we are integrating both visual and text based classifiers [9].

8. REFERENCES

- [1] S.-F. Chang, J. R. Smith, M. Beigi and A. Benitez *Visual Information Retrieval from Large Distributed On-Line Repositories* Communications of the ACM, vol. 40, no. 12, 1997. Special issue on Visual Information Management.
- [2] J.R. Smith and S.-F. Chang *Tools and Techniques for Color Image Retrieval* Proceedings of the Symposium on Electronic Imag: Science and Technology-Storage and Retrieval for Image and Video Databases IV, IS&T/SPIE, 1996.
- [3] S. Paek *Advent Demo: inLumine Image Classification System* <http://www.ee.columbia.edu/advent>
- [4] D. A. Forsyth and M. M. Fleck *Finding Naked People* Proceedings of the European Conference on Computer Vision, 1996.
- [5] S. Russell and P. Norvig *Artificial Intelligence: A Modern Approach* Prentice Hall, 1995.
- [6] A. Vailaya, A. Jain and H. J. Zhang *On Image Classification: City vs. Landscape* Proceedings of the IEEE Workshop on Content-Based Access of Image and Video Libraries, 1998.
- [7] J. R. Smith and S.-F. Chang *Multi-Stage Classification of Images from Features and Related Text* Proceedings of the Fourth DELOS Workshop, 1997.
- [8] Martin Szummer and Rosalind W. Picard *Indoor-Outdoor Image Classification* IEEE Workshop on Content Based Access of Image and Video Databases (CAIVD-98), 1998.
- [9] S. Paek, C. Sable, V. Hatzivassiloglou, A. Jaimes, B. Schiffman, S.-F. Chang and K. McKeown *Integration of Visual and Text Based Approaches for the Content Labeling of Photographs* ACM SIGIR '99, 1999. Special Session on Multimedia Indexing and Retrieval.
- [10] N. Vasconcelos and A. Lippman *A Bayesian Framework for Semantic Content Characterization* Proceedings of the IEEE Computer Vision and Pattern Recognition Conference, 1998.