Inference using Probabilistic Concept Trees

Doug Fisher **Doug Talbert**

{dfisher|dat}@vuse.vanderbilt.edu

Department of Computer Science
Box 1679, Station B
Vanderbilt University
Nashville, TN 37235 USA

1 Introduction

Discussions of 'probabilistic reasoning systems' often presuppose a *belief network*, which represents the joint probability distribution of a domain, as the primary knowledge structure. However, another common knowledge structure from which the joint probability distribution can be recovered is a hierarchical probabilistic clustering or *probabilistic concept tree* (Fisher, 1987). Probabilistic concept trees are a target structure for a number of clustering systems from machine learning such as COBWEB (Fisher, 1987) and systems by Hadzikadik and Yun (1989), Gennari, Langley, and Fisher (1989), Decaestecker (1991), Anderson and Matessa (1991), Reich and Fenves (1991), Biswas, Weinberg, and Li (1994), De Alte Da Veiga (1994), Kilander (1994) Ketterlin, Gançarski, and Korczak (1995), and Nevins (1995). Related probabilistic structures are produced by systems such as AUTOCLASS (Cheeseman, Kelly, Self, Stutz, Taylor, & Freeman, 1988), SNOB (Wallace & Boulton, 1968; Wallace & Dowe, 1994), and systems by Hanson and Bauer (1989) and Martin and Billman (1994). These systems can be easily adapted to form probabilistic concept trees of the type we describe. This paper will not focus on clustering systems *per se*, but on characteristics and capabilities of probabilistic concept trees, particularly as they relate to inference tasks often associated with belief networks. As 'object-centered' knowledge structures, probabilistic concept trees nicely complement the 'variable-centered', belief network structure.

2 Probabilistic Concept Trees

Probabilistic concept trees are typically constructed from data (objects, observations, entities) via clustering. We assume that an object is a vector of values, V_{ij} along distinct (finite-valued) variables, A_i. A hierarchical clustering is a tree-structured collection of clusters (classes), where sibling clusters partition the observations covered by their common parent, and actual objects are represented at leaves. As in COBWEB, AUTOCLASS (Cheeseman, et. al., 1988), and other systems (Anderson & Matessa, 1991), we will assume that clusters, C_k, are described probabilistically: each variable value has an associated conditional probability, $P(A_i = V_{ij}|C_k)$, which reflects the proportion of observations in C_k that exhibit the value, V_{ij}, along variable A_i.

Probabilistically-described clusters arranged in a tree form a hierarchical clustering known as a probabilistic concept (or categorization) tree. Each set of sibling clusters partitions the observations covered by the common parent. The probability (proportion) of $A_i = V_{ij}$ at a cluster C_k is given by: $P(A_i = V_{ij}|C_k) = \sum_l P(C_{kl}|C_k)P(A_i = V_{ij}|C_{kl})$, where C_{kl}'s are children of C_k. There is a single *root* cluster, identical in structure to other clusters, but covering all observations. Figure 1 gives an example of a probabilistic concept tree (i.e., a hierarchical clustering) in which each node is a cluster of observations summarized probabilistically. Observations are at leaves and are described by three variables: `Size`, `Color`, and `Shape`.

3 Constructing Probabilistic Concept Trees from Data

Clustering systems form clusterings (probabilistic concept trees and other structures) guided by some measure that favors clusterings with high (intra-cluster) similarity between objects within the same clusters and low (inter-cluster) similarity between objects of differing clusters. We will not discuss this process in depth, but see Fisher (1996) for details. Suffice it to say that probabilistic concept trees constructed in experiments reported here were constructed top-down: a set of observations was partitioned into clusters,

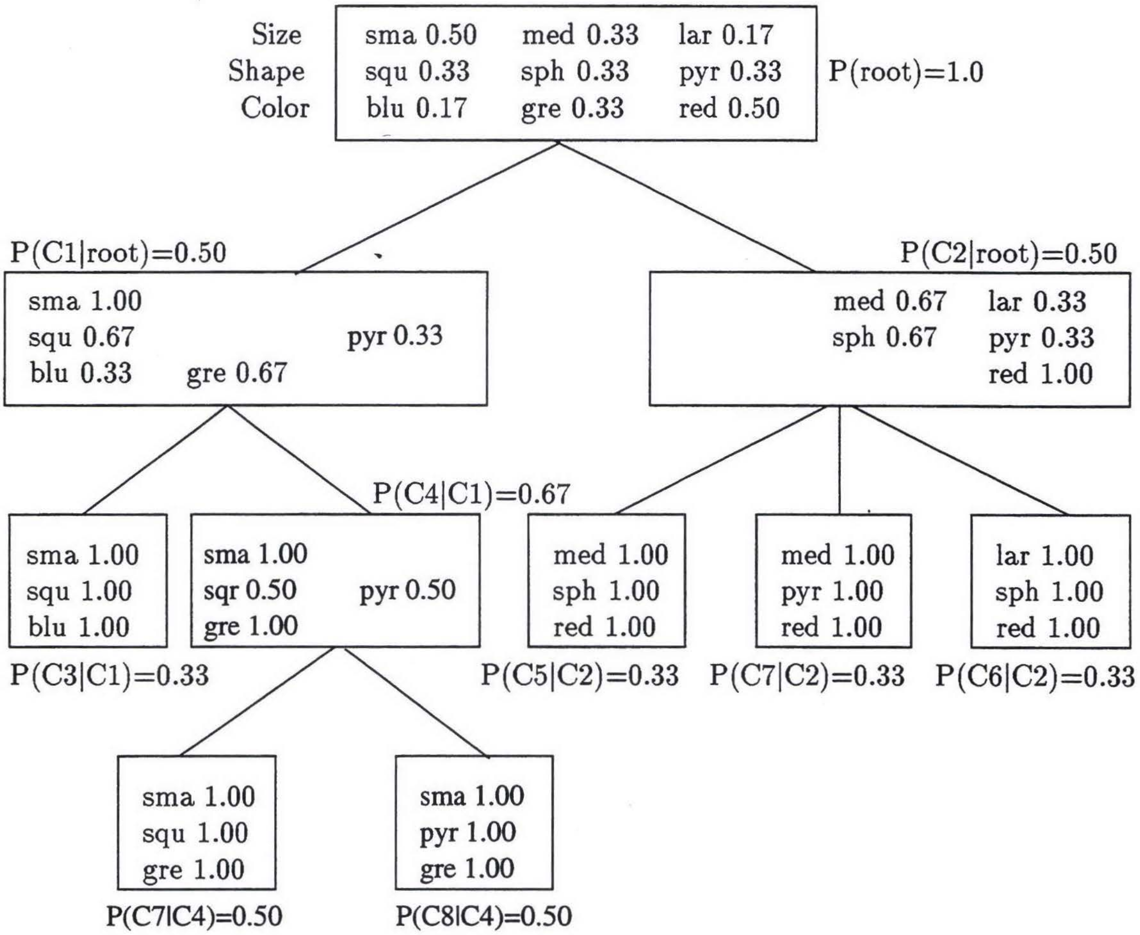

Figure 1: A probabilistic concept tree.

C_k, in an attempt to optimize

$$\frac{\sum_k^N P(C_k) \sum_i \sum_j [P(A_i = V_{ij}|C_k)^2 - P(A_i = V_{ij})^2]}{N}$$

which is a measure based on work by Corter and Gluck (1992). N is the number of clusters in the partition. $P(A_i = V_{ij})$ is the unconditioned probability of V_{ij} (over the whole population of observations), and is stored at the root of the tree. Intuitively, the measure favors a partition that maximizes the average increase in the 'purity' of attributes across the clusters. Fisher (1996) notes some problems with the measure and suggests some others that are based on split measures used for decision tree induction. Bayesian (Cheeseman, et al, 1988; Tirri, Kontkanen, & Myllymäki, 1996) and MML (Wallace & Boulton, 1968) measures are other alternatives that more directly seek the most probable partition.

The observations of each cluster, C_k, in the top-level partition are then partitioned further (using the same measure), with C_k playing the role of 'root'. This process continues, finally decomposing the data into singleton (single-object) clusters.

4 Using Probabilistic Concept Trees for Prediction

Given an established probabilistic concept tree, and a partial object description where certain *evidence* variables have known values, we can categorize the partial object description relative to the concept tree and predict values along the unknown (*query*) variables of the object. Thus, the tree is used to facilitate a task akin to pattern completion. As in decision tree induction, it is typically, but not necessarily the case, that categorization proceeds down one path of the tree. However, categorization using a probabilistic concept tree is generally a polythetic process at each node – based on an observation's values along many attributes. In contrast, categorization with a decision tree proceeds by looking to only a single variable at each node to direct the (monothetic) process.

For example, we might evaluate a new observation's similarity (using a suitable measure) to the probabilistic summary (or prototype) of each cluster and classify the observation as a member of the most similar cluster.

Instead, we use the same measure that was used to construct clusters by placing a (partially-described) observation into a cluster among a set of siblings, (temporarily) updating the attribute distribution of the cluster based on the observation's values, then evaluating the quality of the resulting partition. We do this for each sibling, and choose to classify the observation as a member of the cluster that maximizes the partition score. The observation is recursively classified in this way along one path in the tree until a leaf is reached. In theory different measures of partition quality or object-to-cluster similarity will lead to different classification behavior, but in practice it is probable that a large number of measures will lead to roughly the same behavior.

We have systematically experimented with a limited form of the pattern-completion task: (1) a partial object description is classified down a single 'best-matching' path of the concept tree, and (2) there are $N-1$ evidence variables and 1 query variable on each prediction trial (Fisher, 1996). In pursuing one path, classification is terminated at a selected node (cluster) along the classification path, and the variable value of highest probability at that cluster is predicted as the unknown variable value of the new observation. Naively, classification might always terminate at a leaf (i.e., an object), and the leaf's value along the specified variable would be predicted as the variable value of the new object. This strategy can actually be viewed as an implementation of instance-based reasoning (and learning through clustering). Tirri et al (1996) and Fisher (1989) both take this view, though they differ in specifics. Both recognize, however, that together with a variety of other implementations of instance-based reasoning, this scheme suffers from a number of drawbacks, notably overfitting.

Thus, we adapt a simple resampling strategy known as *holdout* (Weiss and Kulikowski, 1991) to determine whether a variable might be better predicted at some internal node in the classification path. In particular, given a hierarchical clustering and a *validation* set of observations, the validation set is used to identify an appropriate *frontier* of clusters for prediction of each variable. Figure 2 illustrates that the preferred frontiers of any two variables may differ, and clusters within a frontier may be at different depths. For *each* variable, A_i, the objects from the validation set are each classified through the hierarchical clustering with the value of variable A_i 'masked' for purposes of classification; at each cluster encountered during classification the observation's value for A_i is compared to the most probable value for

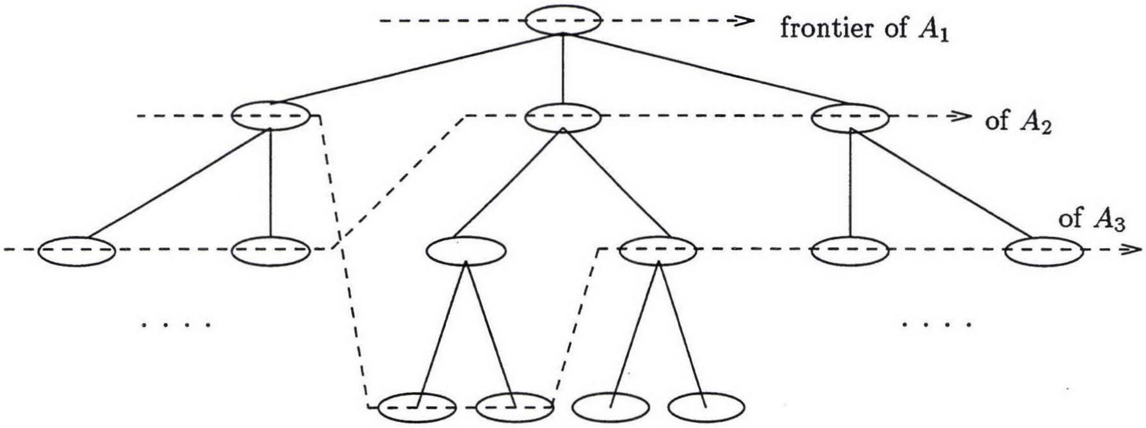

Figure 2: Frontiers for three variables in a hypothetical clustering.

	Unvalidated	Validated
Soybean (small)		
Leaves	18.00 (0.00)	13.10 (1.59)
Accuracy	0.85 (0.01)	0.85 (0.01)
Ave. Frontier Size	18.00 (0.00)	2.75 (1.17)
Soybean (large)		
Leaves	122.00 (0.00)	79.10 (5.80)
Accuracy	0.83 (0.02)	0.83 (0.02)
Ave. Frontier Size	122.00 (0.00)	17.01 (4.75)
House		
Leaves	174.00 (0.00)	49.10 (7.18)
Accuracy	0.76 (0.02)	0.81 (0.01)
Ave. Frontier Size	174.00 (0.00)	9.90 (5.16)
Mushroom		
Leaves	400.00 (0.00)	96.30 (11.79)
Accuracy	0.80 (0.01)	0.82 (0.01)
Ave. Frontier Size	400.00 (0.00)	11.07 (4.28)

Table 1: Characteristics of optimized clusterings before and after validation. Average and standard deviations over 20 trials.

A_i at the cluster; if they are the same, then the observation's value would have been correctly predicted at the cluster. A count of all such correct predictions for each variable at a cluster is maintained. Following classification for all variables over all observations of the validation set, a preferred frontier for each variable is identified that maximizes the number of correct counts for the variable. This is a simple, bottom-up (post-order) procedure that insures that the number of correct counts at a node on the variable's frontier is greater than or equal to the *sum* of correct counts for the variable over each set of mutually-exclusive, collectively-exhaustive descendents of the node. Our method of validation is inspired by retrospective pruning strategies in decision tree induction such as *reduced error pruning* (Quinlan, 1993).

During testing of a validated clustering, each variable of each test observation is masked in turn; when classification reaches a cluster on the frontier of the masked variable, the most probable value is predicted as the value of the observation; the proportion of correct predictions for each variable over the test set is recorded. For comparative purposes, we also use the test set to evaluate predictions stemming from the unvalidated tree, where all variable predictions are made at the leaves (singleton clusters, objects) of this tree.

Table 1 shows results from 20 experimental trials in four domains using unvalidated and validated clusterings, where the initial probabilistic concept trees were generated through some 'reasonable' process described in Fisher (1996). The first row of each domain lists the average number of leaves (over the 20 experimental trials) for the unvalidated and validated trees. A leaf in a validated clustering is a cluster (in the original clustering) that is on the frontier of *at least one* variable, and none of its descendent clusters (in the original clustering) are on the frontier of any variable. For example, if we assume that the tree of Figure 2 covers data described only in terms of variables A_1, A_2, and A_3, then the number of leaves in this validated clustering would be 7. Prediction accuracies in the second row of each domain entry are the mean proportion of correct predictions over *all* variables over 20 trials. Predictions were generated at leaves (singleton clusters) in the unvalidated hierarchical clusterings and at appropriate variable frontiers in the validated clusterings. In all cases, validation can be used to substantially prune the concept tree without diminishing 'pattern-completion' accuracy.

Finally, the third row shows the 'average frontier size', which gives the average number of clusters per variable frontier (averaged over all variables, 20 trials). This gives a better measure of the compression possible than 'Number of Leaves', which does not distinguish concept trees with only one or a few deep frontiers from a concept tree with uniformly deep frontiers.

5 Probabilistic Inference with Probabilistic Concept Trees

Undoubtedly, exploiting only one path during inference yields good results because (1) our tests condition classification and inference using $N-1$ (of N) evidence variables, and (2) there is considerable structure is the (i.e., sparse) domains that we examined. Developers of systems like AUTOCLASS (Cheeseman et. al., 1988) and SNOB (Wallace & Dowe, 1994) have long advocated probabilistic assignment of objects to classes during classification, thus necessitating the combination of evidence from various paths for purposes of inference. We have not experimented as yet along these lines, but we expect that probabilistic classification is desirable for accurate estimates of joint/conditional distributions when only a few evidence variables are specified and/or structure in the data is present, but 'weak'.

A recursive procedure, based on a strategy used with decision trees (Quinlan, 1993), examines multiple paths of a probabilistic concept tree to infer joint probabilities: the probabilities of a conjunctive outcome can be additively combined across (mutually-exclusive) sibling clusters, and probabilities of individual conjuncts can be multiplicatively combined within 'appropriate' clusters (nodes) and subtrees. These 'appropriate' nodes correspond to nodes along variable frontiers; a variable's frontier indicates the nodes/clusters at which the variable in conditionally independent of other variables (though our current implementation indirectly tests/identifies such nodes by determining where performance peaks – classification proceeds deep enough to exploit dependencies, without venturing to a point where overfitting to spurious correlations detracts from performance).

6 Probabilistic Trees and Belief Networks

Trivially, a probabilistic concept tree can be viewed as a belief network: each cluster is a binary-valued hidden variable (member, nonmember). The mutual exclusion of sibling clusters makes the conditional probability table at each cluster (with exactly one parent) quite simple. However, clusters also have primitive, observable variables as children. Each cluster may be viewed as the parent of a primitive variable 'node'. However, because of ancestral relationships among nodes, it is only the frontier of most specific (deepest) clusters that point at a variable that are important. If this frontier (of parents) for each variable is the set of singleton clusters (leaves), then the network does not generalize the data. However, we postulate that if the 'frontier of parents' for a variable corresponds to the variable frontier as identified Section 4, then good estimations of the joint distribution can be expected with a sparse network (in many natural domains).

Clusters of the original clustering that are on the frontier of several variables are parents to several variables in the network. Once the clustering is constructed, the corresponding belief network need not retain most of the $P(A_i = V_{ij}|C_k)$'s. Once translated, inference with the belief network can proceed by computing the posterior distribution of query variables given evidence variables. This process is considerably different than the strategy of top-down classification used with hierarchical clusterings, though we expect that the performance of the network will be very similar to performance with the procedure outlined in Section 5. Connolly (1993) has also used clustering as a tool for belief network construction, though his procedures for using clustering to construct hidden variables in a Bayesian network differs from ours. Our view is that clustering creates the network, which then only needs to be simplified and 'cleaned up.'

7 Concluding Remarks

We plan to experiment with evidence combination along multiple paths of a probabilistic concept tree as outlined in Section 5, and to flesh out and implement the clustering-to-network translation scheme outlined in Section 6. Independent experimental dimensions of interest include the number of evidence and number of query variables within a trial. Dependent experimental dimensions include a measure of error of estimated joint distributions. Our previous experiments with variable frontiers suggest that small probabilistic concept trees can yield good estimates of the joint distribution in many domains, just as very sparse belief networks often give good estimates.

Acknowledgements

We thank the reviewers for helpful comments and suggestions. Pat Langley motivated us to explore the relationship between probabilistic concept trees and belief networks. Figure 2 and Table 1 are taken from (Fisher, 1995) and used with permission from AAAI Press. Subsequent drafts of this and related work and data can be found at
http://cswww.vuse.vanderbilt.edu/~dfisher/tech-reports.html.

References

Anderson, J. R., & Matessa, M. (1991). An interative Bayesian algorithm for categorization. In D. Fisher & M. Pazzani (Eds.), *Concept formation: Knowledge and experience in unsupervised learning.* San Mateo, CA: Morgan Kaufmann.

Biswas, G., Weinberg, J., & Li, C. (1994). ITERATE: A conceptual clustering method for knowledge discovery in databases. In B. Braunschweig and R. Day (Eds.) *Innovative Applications of Artificial Intelligence in the Oil and Gas Industry.* Editions Technip.

Breiman, L., Friedman, J., Olshen, R., & Stone, C. (1984). *Classification and Regression Trees.* Belmont, CA: Wadsworth.

Cheeseman, P., Kelly, J., Self, M., Stutz, J., Taylor, W., & Freeman, D. (1988). AutoClass: A Bayesian classification system. *Proceedings of the Fifth International Machine Learning Conference* (pp. 54–64). Ann Arbor, MI: Morgan Kaufmann.

Connolly, D. (1993). Constructing hidden variables in Bayesian networks via conceptual clustering. *Proceedings of the Tenth International Conference on Machine Learning* (pp. 65–72). San Fransisco, CA: Morgan Kaufmann.

Corter, J., & Gluck, M. (1992). Explaining basic categories: feature predictability and information. *Psychological Bulletin, 111*, 291–303.

Decaestecker, C. (1989). Incremental concept formation with attribute selection. *Proceedings of the Fourth European Working Session on Learning* (pp. 49–58). Montpellier, France.

Fisher, D. H. (1987a). Knowledge acquisition via incremental conceptual clustering. *Machine Learning, 2*, 139–172.

Fisher, D. H. (1989). Noise-tolerant conceptual clustering. *Proceedings of the International Joint Conference Artificial Intelligence* (pp. 825–830). Detroit, MI: Morgan Kaufmann.

Fisher, D. H. (1995). Optimization and simplification of hierarchical clusterings. *Proceedings of the First International Conference on Knowledge Discovery and Data Mining* (pp. 147–180). Menlo Park, CA: AAAI Press.

Fisher, D. H. (1996). Iterative optimization and simplification of hierarchical clusterings. *Journal of Artificial Intelligence Research, 4*, 147–180.

Fisher, D. H., & Langley, P. (1990). The structure and formation of natural categories. In G. H. Bower (Ed.), *The Psychology of Learning and Motivation.* San Diego, CA: Academic Press.

Fisher, D., Xu, L., Carnes, J., Reich, Y., Fenves, S., Chen, J., Shiavi, R., Biswas, G., & Weinberg, J. (1993). Applying AI clustering to engineering tasks. *IEEE Expert, 8*, 51–60.

Gennari, J., Langley, P., & Fisher, D. (1989). Models of incremental concept formation. *Artificial Intelligence, 40*, 11–62.

Hadzikadic, M., & Yun, D. (1989). Concept formation by incremental conceptual clustering. *Proceedings of the International Joint Conference Artificial Intelligence* (pp. 831–836). Detroit, MI: Morgan Kaufmann.

Hanson, S. J., & Bauer, M. (1989). Conceptual clustering, categorization, and polymorphy. *Machine Learning, 3*, 343–372.

Hanson, R., Stutz, J., & Cheeseman, P. (1991). Bayesian classification with correlation and inheritance. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence*, 692–698. San Mateo, CA: Morgan Kaufmann.

Ketterlin, A., Gançarski, P., J. Korczak (1996). Hierarchical clustering of composite objects with a variable number of components. In D. Fisher and H.-J. Lenz (Eds.), *Learning from Data: Artificial and Statistics V*, New York: Springer.

Kilander, F. (1994). *Incremental Conceptual Clustering in an On-Line Application.* (Doctoral Dissertation, Report No. 94-014). Department of Computer and Systems Sciences, Stockholm University.

Lebowitz, M. (1987). Experiments with incremental concept formation: UNIMEM. *Machine Learning, 2*, 103–138.

Martin, J., & Billman, D. (1994). Acquiring and combining overlapping concepts. *Machine Learning, 16*, 121–155.

Nevins, A. J. (1995). A branch and bound incremental conceptual clusterer. *Machine Learning, 18*(1), 5–22.

Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning.* San Mateo, CA: Morgan Kaufmann.

Reich, Y., & Fenves, S. (1991). The formation and use of abstract concepts in design. In D. Fisher & M. Pazzani (Eds.), *Concept formation: Knowledge and experience in unsupervised learning.* San Mateo, CA: Morgan Kaufmann.

Tirri, H., Kontkanen, P., & Myllymäki (1996). Probabilistic Instance-Based Learning. In L. Saitta (Ed.), *Proceedings of the Thirteenth International Conference on Machine Learning* (pp. 507–515). San Fransisco, CA: Morgan Kaufmann.

Wallace, C. S., & Boulton, D. M. (1968). An information measure for classification. *Computer Journal, 11*, 185–194.

Wallace, C. S., & Dowe, D. L. (1994). Intrinsic classification by MML – the SNOB program. *Proceedings of the 7th Australian Joint Conference on Artificial Intelligence*, (pp. 37–44). UNE, Armidale, NSW, Australia: World Scientific.

Weiss, S., & Kulikowski, C. (1991). *Computer Systems that Learn.* San Mateo, CA: Morgan Kaufmann Publishers.