Using Information Gain to Build Meaningful Decision Forests for Multilabel Classification

Kevin Gold
Department of Computer Science
Wellesley College, Wellesley, MA 02481 USA
Tel.: +1-781-697-0732
Email: kgold@post.harvard.edu

Allison Petrosino
Department of Computer Science
Wellesley College, Wellesley, MA 02481 USA
Email: allisonpetrosino@alum.wellesley.edu

***Abstract*—“Gain-Based Separation” is a novel heuristic that modifies the standard multiclass decision tree learning algorithm to produce forests that can describe an example or object with multiple classifications. When the information gain at a node would be higher if all examples of a particular classification were removed, those examples are reserved for another tree. In this way, the algorithm performs some automated separation of classes into categories; classes are mutually exclusive within trees but not across trees. The algorithm was tested on naive subjects’ descriptions of objects to a robot, using YUV color space and basic size and distance features. The new method outperforms the common strategy of separating multilabel problems into L binary outcome decision trees, and also outperforms RAkEL [1], a recent method for producing random multilabel forests.**

I. Introduction

When humans provide labels for a machine learner in the form of freely generated object descriptions, some classifications may be mutually exclusive, and others not. One item may be big and yellow, another small and blue, but no item can be both big and small. In passively acquiring language from natural interactions, it is much more likely for a person to make positive attributions (“bring me that small cup”) than negative ones (“bring me that cup that isn’t big”). If the system is to learn passively without explicit training, most of the information for decision boundaries must come from the contrast of labels in the same category. However, grammar gives no cue as to which adjectives belong to which feature categories, so multimodal learners must somehow decide which labels contrast and which can co-exist. Rather than attempt to anticipate all possible labels and their categories prior to learning, it is desirable to create a machine learner that can extract label categories from the data alone. The problem is difficult for a machine learner, however, because real objects with continuous feature vectors usually differ from each other in almost all of their feature values, making it difficult to determine which features are responsible for a particular label.

The present study explores “Gain-Based Separation,” a new heuristic for creating multiple multiclass decision trees in cases where some labels should be mutually exclusive, while other labels are compatible. When a tree appears to be more informative when particular labels are omitted, these labels are set aside for a new tree. Once the first tree is done, a new tree is begun with the set aside labels, and if any labels seem to be a poor fit for this tree, they are again set aside. This process is repeated until several multiclass trees have been created, each with distinct labels. When it comes time for classification, each tree can provide exactly one label to the example, allowing labels to be mutually exclusive if they are in the same tree, or mutually compatible if they are in different trees.

The representations the method creates are more structured and concise than typical forest-generating ensemble methods, in that labels are not repeated across trees and each tree performs a unique function instead of voting with other trees. Unlike typical decision forest methods [1], [2] but like classic multiclass decision trees, the present method has the advantage that the final “definitions” for each label are easily examined and potentially usable in logical programming, and thus it is a good fit for natural language comprehension and generation.

The present work is a continuation of the TWIG system for learning logical natural language definitions from sensory examples presented to a robot [3]. TWIG made the assumption that all words within a particular grammatical category were mutually exclusive – unlike the present work, which deals with the problem of inferring which words are mutually exclusive and which are not. The mutual exclusivity assumption is common among children learning first words [4], and is helpful for making use of contrast when no explicit negative examples are provided [3], but it is clearly not correct for adjectives and prepositions. Thus, it is possible that this work may provide food for thought in considering how children might apply the Principle of Mutual Exclusivity [4], which does not always hold in the case of adjectives.

A. Related Work

Several recent methods have been developed for dealing with the problem of “multilabel” classification problems, where up to L labels (classifications) can apply to the same example. Two benchmarks for performance still being used [5] include “binary relevance” (BP), the practice of separating the L-label problem into L separate binary classification problems, and “label powerset” (LP), the practice of turning the L-label problem into a single multiclass problem with 2^L labels by using the power set of labels as the set of possible labels. Recent methods that sometimes perform better than these baseline approaches include BP-MLL [6], a modification of backpropagation neural networks that can handle multiple labels as output; ML-kNN, a k-nearest neighbors algorithm for producing multiple labels [7]; and RAkEL [1], a method that renders the LP method more tractable by creating a forest of trees that each use the LP method for a random subset of labels. The present paper will compare Gain-Based Separation to weighted BP, RAkEL, and standard multiclass trees, since these are the methods that can use some variant on decision trees as their underlying algorithm (besides basic LP, which is intractable for the sizes of problems we consider).

The use of multiple decision trees for classification has a precedent in the use of "decision forests" [2] and boosting [8], but these methods are not designed to provide multiple, different classifications that all apply simultaneously. In both cases, the use of multiple decision trees is used to produce a more accurate single classification, rather than produce multiple labels for the same example. There are many more variants on decision trees, such as decision graphs [9] or methods for producing more complex decision surfaces [10]; though the method presented here might be adapted to these variants, this is beyond the scope of the current paper. The case of multiple compatible labels which address different aspects of the feature space has also been addressed in the case where the labels belong to a known taxonomy [11].

Robotic language learning work has typically been most successful in adjective learning by clustering sensory examples of each word, thereby creating a finite region of feature space in which each word applies [13], [14]. These methods are arguably less desirable than those that partition the entire feature space, since finite regions counterintuitively reject extreme examples (an object bigger than any seen before would not qualify as *big*), but the aforementioned studies also address many practical issues that are beyond the scope of this paper, such as learning audio segmentation. The problem of how to learn multiclass classifiers from "positive examples" alone was addressed in [15], which taught neural networks meanings for prepositions from visual examples using the strategy of treating non-target words as weak negative examples. Several other studies take the approach of treating grounded language learning as a problem of statistical translation between natural language and a concept/symbol language [16], [17]. The present study instead treats the adjective learning problem as that of simultaneously learning partitions of feature space and the mapping of labels to these partitions – in other words, allowing labels to help shape concept boundaries, rather than simply referring to them. Color space in particular tends to be partitioned differently in different languages, suggesting that language shapes these concept boundaries [18], though some have provided evidence that certain points in color space may be privileged for receiving definitions [19]. Regardless, from a machine learning perspective, it is useful to have algorithms that can work with as little a priori information as possible, so in the experiments below, the algorithm is given no a priori information about which labels belong to which categories, nor where boundaries are likely to lie.

II. Separating Decision Trees

A. The Problem

The algorithm presented here is for the following general learning problem. There is a set of "adjective" functions $f_1, f_2, \ldots, f_N$, where N is not known in advance. Each function f_i maps a continuous-valued object vector $\mathbf{x}$ to a label (word) in its corresponding category C_i: $f_i : \Re^n \rightarrow w \in C_i$. We assume the categories are disjoint. The learner is given "overheard" descriptions of each object, modeled as pairs $(\mathbf{x_j}, w_j)$, but does not know which function was used to produce each word w_j. The goal is then to reconstruct the functions f_i, and implicitly, the categories C_i that are the ranges of each function. In the real world the training set may be corrupted by noise and error, and there may be words in the corpus that cannot be mapped to the feature space at all, but this abstraction captures the essential details of the problem.

Of particular interest is the case in which each function f_i uses a different subspace of the feature space. For example, a decision tree for color labels should not contain any decisions about distance features, and vice versa. The algorithm that follows is most successful when this assumption holds, but it does not necessarily require the assumption to be true, nor does it impose this structure on the representation.

B. Algorithm: Gain-Based Separation

The algorithm with the separation heuristic applied is almost the same as the standard decision tree algorithm [20], but with the difference that if the information gain would be greater at a decision point if a particular label were omitted from the tree altogether, it is reserved for a different tree.

In greater detail: the evidence at each level of the tree consists of a set of labels, each paired with a list of observed facts about the referent. These lists function as the input vectors $\mathbf{x_j}$, and can contain boolean, discrete, or continuous information. The information content at a particular node of the tree being built is given by

$$I(T) = \sum_i -P(\ell_i) \log_2 P(\ell_i) \qquad (1)$$

where T is a list of paired labels and input vectors, and $P(\ell_i)$ is the evident probability of the label, $|\ell_i|/|T|$. If a binary decision D partitions the evidence set into subsets T_p and T_n based on whether each example satisfies the decision's test, the new expected information $I(D, T)$ is:

$$I(D,T) = \frac{|T_p|}{|T|} I(T_p) + \frac{|T_n|}{|T|} I(T_n) \qquad (2)$$

The information gain given by D is then simply $Gain(D, T) = I(T) - I(D, T)$. Potential decisions are generated in continuous spaces by allowing each sample point to be a potential decision threshold in either direction, for each feature.

The primary change to this familiar algorithm is to calculate at each decision the alternate gain $Gain(D, T_{\bar{k}}) = I(T_{\bar{k}}) - I(D, T_{\bar{k}})$, where $T_{\bar{k}}$ is the set of all examples that are not

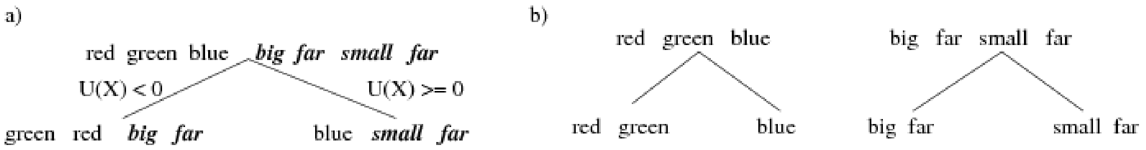

Fig. 1. (a) Labels irrelevant to the prevailing category of the current tree (bold italics) tend to occur on both sides of the best decision branch (here, a color feature in YUV space), reducing the decision's information gain. All labels that reduce the information gain are removed and reserved for another tree. (b) The new tree will tend to contain labels about aspects of the object different from the first tree. The algorithm here would next remove "far" for a third tree.

labeled ℓ_k, for all labels ℓ_k such that $T \neq T_{\bar{k}}$. That is, for each label, the information gain is recomputed as if the examples with that label did not exist. All examples with labels for which $Gain(D, T_{\bar{k}}) > Gain(D, T)$ are removed from the tree and reserved for the next tree. The algorithm checks all labels at every decision, and if multiple labels at a single node satisfy the criterion, they are all removed everywhere they appear in the tree. No optimal decisions in the tree are recalculated, however, on the assumption that removing irrelevant words should not change the best decisions.

As usual, to finish the original tree, decisions are greedily and recursively chosen until a chi-square test determines that the best decision is no longer statistically significant (here, $p > 0.01$). The final label at a leaf is determined from the examples by a majority function.

The next tree is built according to the same heuristics, using the examples that had been removed from the original tree. Reserved labels from the second tree are set aside for a third tree, and so on until labels are no longer flagged for removal. If there are only one or two distinct labels, omitting a label results in an information gain of 0, so labels can only be reserved when there are more than two different labels in a tree's evidence set. It is possible and common for the final tree to contain no informative decisions, in which case these labels are considered too uncertain to produce a meaningful definition, and they are not used for production.

Each tree then corresponds to one reconstituted function F_i. When an example $\mathbf{x}$ is given to the learned representation, the algorithm produces as a description the set of labels $\cup i\{\ell_j : F_i(x) = \ell_j\}$.

C. Comparison to RAkEL

Gain-Based Separation is most usefully compared to RAkEL [1], a recent ensemble method designed to produce a forest that can provide multilabel classifications. One possible way to transform any multilabel problem with label set L into a single-classification problem is to create a new set of mutually exclusive labels $P(L)$ that consists of the power set of those labels. With $2^{|L|}$ possible labels in the transformed problem, this method is obviously impractical for our scenario in which words can be freely provided. RAkEL renders this brute force approach into a tractable ensemble method. RAkEL stands for "random k-labelsets," and the idea is to create m trees that each use the powerset of only k of the labels, randomly chosen. Each tree then produces a compound label such as "yellow-tall" or "far-short," and each tree that contains a label in its k-labelset receives a vote as to whether a label should be included in the final production. If a majority of relevant trees produce a label as part of a compound label, then that label is included in the final production. Similar to boosting, then, the forest that is produced contains many trees that in themselves do not necessarily contain very meaningful definitions, but taken as an aggregate, produce the right results. (For the experiments presented below, $k = 5$ and $m = 10$ produced the best results, though values between 5 and 10 were tried for k, and between 5 and 20 for m. The range was chosen to achieve roughly the same representational complexity as the present method.)

The present method attempts to find more structure in the problem by only putting labels in the same tree if the decisions of that tree appear to be relevant to those labels. Rather than divvying labels that refer to the same features among multiple trees, each tree should receive more or less one category of mutually exclusive labels. The trees do not vote, but each produces one label. In this way, a more concise representation that better matches the underlying definitions is produced than typical ensemble methods. It was hypothesized that this better match of representation to definition would result in better classification performance.

III. Experiment

A. Methods

The first experiment was performed with 4 human subjects giving descriptions to a Surveyor corporation Blackfin robot [21]. The subjects were presented with 32 objects belonging to three categories (candles, books, and toy letters), in 9 groups of 3-5 objects, and were asked to describe the objects in such a manner that the robot would know which object to drive to. The subjects were told that the robot could not read, but could understand words related to size, color, and distance. These free descriptions were recorded for offline processing by the algorithms to be tested.

Once the descriptions to be used in training were collected, a second data set was collected for evaluation. To obtain ground truth for the purposes of measuring precision and recall, the subjects were asked for each object to describe its color, as well as whether it was tall, short, or neither; big, little, or neither; fat, skinny, or neither; thin, thick, or neither; large, small, or neither; and near, far, or neither. In this way, the algorithms could be evaluated not by what the subject originally said, but by how well they produced descriptions that the subjects would agree with.

The subjects' original freeform descriptions (not the complete ground truth descriptions) were given to the robot paired with the experimental conditions that elicited each description. On receiving a description, the robot turned off its laser and performed a depth-first-search based color segmentation of the image starting from where the laser point had been, showing the result in a GUI window. If the segmentation was correct, this was confirmed with "yes, that's a ...". The robot then recorded noisy estimates of 6 features: a color sample in YUV space, estimated depth from the position of the laser in the camera's field of view, and height and width taken from the pixel counts, scaled by the estimated depth.

The phrases were paired with the recorded features and presented to four algorithms for comparison. The first algorithm was a standard multiclass decision tree learner. The second produced one tree per label, treating all examples of other labels as weak (0.25 weight) negative examples. The third algorithm was RAkEL, described below, and the fourth was the algorithm described in this paper, "gain-based separation." For each algorithm, the phrases were broken into their constituent words, such that each training example consisted of a word label and a vector of continuous features.

All algorithms were tested under two conditions: one which omitted all non-adjectives or adjectives which the robot could not represent as "stop words," and one which contained no such distinctions. All algorithms were trained on all freeform data, then tested against each subject's ground truth estimations, which served as test sets. For RAkEL, only $k = 5$ and $m = 10$ (the values that produced the best F-measure) are reported for conciseness, and results are reported for both a pruned ($p < 0.01$ at each node, as with the other methods) and unpruned version, on the theory that unpruned trees would perform better when the powerset labels could be very uncommon.

B. Results

Table I shows the results for the basic case in which words irrelevant to the robot's sensing capabilities were ignored. Precision was calculated as the fraction of labels the algorithm generated that subjects agreed with (but did not necessarily produce in the training corpus), and recall was the fraction of labels the subject agreed with that were produced by the algorithm. Some leeway was granted for near-synonyms that the subjects were not asked about; if a subject said an object was "close," it was counted as "near," though subjects were not specifically asked about the word. The F-measure used here is the harmonic mean of precision and recall.

The low recall of the single-tree method is not surprising, since it could only produce one label per example; this was mostly a check to ensure the overall F-measure improved by using the new method, which it did. The one tree per label method performed better, but not as well as the new method. RAkEL performed remarkably similarly to these baselines – when pruned, it performed similarly to the single tree method, and when not, its performance was very similar to the one tree per label method.

Algorithm	Precision	Recall	F
Single Multiclass Tree	0.80	0.16	0.27
One Tree Per Word	0.77	0.25	0.38
RAkEL, no pruning	0.76	0.26	0.39
RAkEL, with pruning	0.82	0.14	0.24
Gain-Based Separation	0.54	0.42	0.47

TABLE I

PERFORMANCE ON THE ADJECTIVE CORPUS PRODUCED BY SUBJECTS IN EXPERIMENT 1, WITH IRRELEVANT WORDS OMITTED.

Including the "stop words" did not change the ranking of the results' F-measures, besides dragging unpruned RAkEL down to be almost as bad as its pruned version. The F-measures were 0.25 for the multiclass tree, 0.34 for one-tree-per-word, 0.24 for unpruned RAkEL, 0.18 for pruned RAkEL, and 0.39 for gain-based separation. RAkEL was particularly adversely affected by the stop words, since the chance of choosing meaningful labels for the k-labelset was reduced. All algorithms were generally adversely affected in precision because of the additional labels that could be mistakenly produced. The full precision and recall table is omitted for space reasons.

Gain-based separation was also faster than most of the other methods, sometimes by a wide margin. When run on the Java 1.6 VM of a MacBook with a 2 GHz Intel Core Duo processor, the algorithm running with no stop words omitted took 13.5s, compared to 652s for the "one tree per word" approach. The algorithm was also faster than RAkEL, which took 27s using prepruning and 43s without, and not much slower than a single multiclass tree (13.3s). The methods' running times changed by no more than 4 seconds when the stop words were omitted, except for the one-tree-per-word method, which took 408s without extraneous labels.

The actual definitions formed by the trees were close to the trees that might be created by hand, but with some labels and trees irrelevant. Figure 2 shows the set of trees generated by training on all four subjects' utterances, with the decision attributes at interior nodes and labels at the leaves. (The decisions' numerical thresholds have been omitted for clarity; when the correct feature was chosen, a reasonable threshold was also chosen, and when an irrelevant feature was chosen, the exact threshold is irrelevant.)

C. Follow-up analysis

To gain a better understanding of gain-based separation's success, binary decision trees which used only category-appropriate features for each word were computed from the data, using standard C4.5 decision trees [20] that only contrasted labels within the same category. The features and words were both separated into 3 categories – color, distance, and size. A binary-valued tree was trained for each word using only words within the same category as negative examples. Precision and recall were computed using these binary forests on the test set. Irrelevant words were omitted. The resulting precision, recall, and F-measure (Table II) show that the gain-based separation method actually performed slightly better

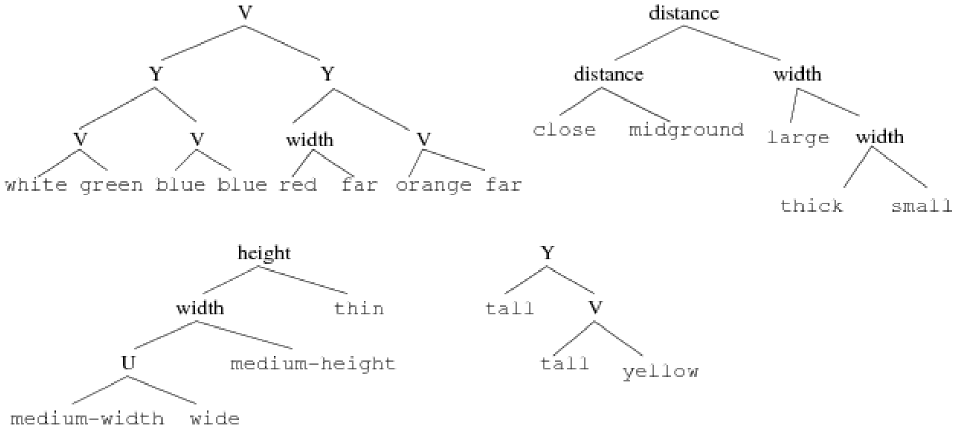

Fig. 2. A sample set of trees generated by the algorithm from subject utterances. Labels are in bold, with decision features shown at the interior nodes. Numerical decision thresholds are omitted for readability. Y is a brightness feature, and U and V mix hue and saturation. The trees are generally more sensible and compact than those produced by random forest methods.

Algorithm	Precision	Recall	F
Gain-Based Separation	0.54	0.42	0.47
Binary-Valued Trees, Categories Known	0.37	0.26	0.30
Binary-Valued Trees, Categories Greedy	0.30	0.16	0.21

TABLE II
FOLLOW-UP EXPERIMENT RESULTS, COMPARING THE ALGORITHM TO TREES THAT USED ADDITIONAL INFORMATION ABOUT THE CATEGORIES.

than these category-informed trees. This result shows that being able to concisely represent the data is more useful than even knowing the relevant features ahead of time; the gains made by being multiclass outweigh the mistakes made in inferring the categories.

A second follow-up analysis was run to suggest whether the algorithm might be improved by changing it to look for an optimal forest, instead of choosing decisions greedily. In the absence of a developed algorithm, we instead examined the assumption that the nature of a whole tree could be inferred from the first decision. As in the previous follow-up experiment, the words were separated by category, but for the first decision, any feature could be used to split. Thereafter, the remaining decisions could only use features within the same category as the first decision. The results here (Table II) show that the greedy assumption does detract from performance, suggesting that gain-based separation could itself benefit from a less greedy approach.

Taken together, the results also suggest that the relatively middling F-measure was the result of a small sample size, since either category-informed method should theoretically have been able to produce the correct trees with a large enough data set.

IV. CONCLUSIONS

Though decision forests and other ensemble methods have typically used very large representations that do not contain much meaningful structure in any one tree, the present results suggest that it can be useful to find concise representations that better capture the implicit contrasts between labels, and avoid contrasting labels that seem mutually compatible. Gain-based separation is a fast heuristic for deciding which labels should be mutually exclusive and therefore contrasted against each other, and which are compatible and should not be used as implicit negative examples. The method's attractiveness only improves in cases where examples often do not receive labels, since it does not produce meaningless distinctions to attempt to explain why examples did not receive a particular label, but only adds decisions to explain labels that should genuinely contrast with each other. The method is generally more successful at producing multiple correct labels than either the binary predicate ("one tree per label") method or the recent RA*k*EL method [1], and outscores both on F measure when precision is valued equally with recall. It also has the advantage of being parameterless, besides any parameters used for pruning.

The method is promising for automatically learning logical definitions and groupings from the structure of the trees, which is an advantage over methods that might produce less semantically meaningful representations. Though the separation into categories was not perfect, it was good enough for several words such that the definitions would be usable in a natural-language parser that produced logical meanings for sentences, as in [22].

These experiments also show that the practice of using non-positive examples as weak negative examples for binary classifiers in multiclass problems may be inferior to methods that can decide when labels are incompatible with each other. Though the assumption that non-positive examples are negative examples has proven successful in adapting other decision tree variants when the classes are mutually exclusive [23], and the use of non-positive examples as weak negative examples has proven a useful heuristic for learning prepositions [15], the present study shows that it may be possible to exploit more structure in the problem to decide when a class is a "negative" example. In general, the results here can be interpreted as supporting Occam's Razor, with smaller representations leading to better performance.

The automatic separation of categorization problems into several mutually exclusive problems may be applicable to other machine learning algorithms besides decision trees. Neural networks and support vector machines are both essentially methods for partitioning feature spaces, and so they similarly make use of contrast for learning. This would be an interesting line of inquiry to pursue.

Gain-based separation's three great weaknesses currently are its reliance on linear, axis-aligned boundaries between regions of feature space; its inability to understand synonyms; and its greedy approach that often results in a meaningless final tree. The first weakness is clearly not intrinsic to the recursive separation idea, but was the result of using basic decision trees as the underlying algorithm to modify. This choice was more based on clarity of exposition and ease of implementation than anything else, and it should be obvious that the method could be applied to any variant on decision trees that uses information gain as its decision criterion. Synoynms are a somewhat trickier issue; one cannot simply classify all labels that appear at the same leaf of the decision tree as synonyms, because this would leave the algorithm very vulnerable to noise. This may not be a very important problem, however, since children commonly implicitly assume there is no such thing as a synonym, and that this seems to be a useful heuristic for early word learning [25]. The final weakness, its greedy approach with no way to backtrack and put labels back into trees they were extracted from, is a good direction for improvement on the algorithm.

In short, the automatic determination of *which* classes ought to contrast is an exciting potential direction for machine learning. The algorithm presented here is demonstrably better than a one-versus-many approach at producing multiple labels and meaningful structures, while retaining most of the precision of the single multiclass decision tree. The present work also suggests that more structured and concise forests could produce better results than typical ensemble or multilabel methods that achieve good performance from replication and voting. A less greedy algorithm and extending the basic idea to other machine learning algorithms are potential future directions.

Acknowledgments

The authors would like the family of Norma Wilentz Hess for their support of this research.

References

[1] G. Tsoumakas and I. Vlahavas, "Random k-labelsets: An ensemble method for multilabel classification," in *Proceedings of the 18th European conference on machine learning (ECML 2007)*, Warsaw, Poland, 2007.

[2] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[3] K. Gold, M. Doniec, C. Crick, and B. Scassellati, "Robotic vocabulary building using extension inference and implicit contrast," *Artificial Intelligence*, vol. 173, no. 1, pp. 145–166, 2009.

[4] E. M. Markman and G. F. Wachtel, "Children's use of mutual exclusivity to constrain the meanings of words," *Cognitive Psychology*, vol. 20, pp. 121–157, 1988.

[5] A. Dimou, G. Tsoumakas, V. Mezaris, I. Kompatsiaris, and I. Vlahavas, "An empirical study of multi-label learning methods for video annotation," in *7th International Workshop on Content-Based Multimedia Indexing*. Crete: IEEE, 2009.

[6] M.-L. Zhang and Z.-H. Zhou, "Multi-label neural networks with applications to functional genomics and text categorization," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 10, pp. 1338–1351, 2006.

[7] ——, "ML-KNN: A lazy learning approach to multi-label learning," *Pattern Recognition*, vol. 40, no. 7, pp. 2038–2048, 2007.

[8] Y. Freund and R. E. Shapire, "Experiments with a new boosting algorithm," in *Machine learning: Proceedings of the thirteenth International Conference*. San Francisco: Morgan Kaufman, 1996, pp. 148–156.

[9] P. Tan and D. Dowe, "MML inference of decision graphs with multi-way joins and dynamic attributes," in *LNCS:AI 2003: Advances in Artificial Intelligence*, T. D. Gedeon and L. C. C. Fung, Eds. Berlin: Springer, 2003, pp. 269–281.

[10] S. Qing-Yun and K.-S. Fu, "A method for the design of binary tree classifiers," *Pattern Recognition*, vol. 16, pp. 593–603, 1983.

[11] F. Wu, J. Zhang, and V. Honavar, "Learning hierarchical classifiers with class taxonomies," in *LNAI:Abstraction, Reformulation, and Approximation*, J.-D. Zucker and L. Saitta, Eds. Berlin: Springer, 2005, vol. 3607, pp. 313–320.

[12] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proceedings of the eleventh annual conference on computational learning theory*. New York, NY: ACM, 1998, pp. 92–100.

[13] D. K. Roy and A. P. Pentland, "Learning words from sights and sounds: a computational model," *Cognitive Science*, vol. 26, pp. 113–146, 2002.

[14] C. Yu and D. H. Ballard, "A multimodal learning interface for grounding spoken language in sensory perceptions," *ACM Transactions on Applied Perceptions*, vol. 1, no. 1, pp. 57–80, July 2004.

[15] T. Regier, *The Human Semantic Potential: Spatial Language and Constrained Connectionism*. Cambridge, MA: MIT Press, 1996.

[16] J. M. Siskind, "Lexical acquisition in the presence of noise and homonymy," in *Proceedings of the 12th National Conference on Artificial Intelligence (AAAI-94)*. MIT Press, 1994, pp. 760–766.

[17] R. J. Kate and R. J. Mooney, "Learning language semantics from ambiguous supervision," in *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence (AAAI-07)*. Menlo Park, CA: AAAI Press, 2007.

[18] A. Gellatly, "Colourful Whorfian ideas: Linguistic and cultural influences on the perception and cognition of colour, and on the investigation of them," *Mind and Language*, vol. 10, no. 3, 1995.

[19] P. Kay and T. Regier, "Resolving the question of color naming universals," *PNAS*, vol. 100, no. 15, 2003.

[20] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, pp. 81–106, 1986.

[21] Surveyor corporation site, http://www.surveyor.com/, retrieved 12/14/08.

[22] F. C. N. Pereira and S. M. Shieber, *Prolog and Natural-Language Analysis*. Menlo Park, CA: CSLI/SRI International, 1987.

[23] H. Hong, W. Tong, R. Perkins, H. Fang, Q. Xie, and L. Shi, "Multiclass decision forest – a novel pattern recognition method for multiclass classification in microarray data analysis," *DNA and Cell Biology*, vol. 23, no. 10, pp. 685–694, 2004.

[24] S. E. Fahlman and C. Lebiere, "The cascade-correlation learning architecture," in *Advances in Neural Information Processing Systems*, D. S. Touretzky, Ed. San Mateo, CA: Morgan Kaufmann, 1990, vol. 2, pp. 524–532.

[25] E. Clark, "The principle of contrast: A constraint on language acquisition," in *Mechanisms of language acquisition*, B. MacWhinney, Ed. Hillsdale, NJ: Lawrence Erlbaum Assoc., 1987, pp. 1–33.