Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence

Discovering Latent Strategies

Xiaoxi Xu
Department of Computer Science
University of Massachusetts Amherst
xiaoxi@cs.umass.edu

This research is motivated by how case similarity is assessed in retrospect in law. In the legal domain, when both case facts and court decisions are present, assessing case similarity by taking account of both case facts and court decisions is more intuitive than considering case facts alone. Discovering similar mappings of case facts to court decisions, or similar strategies that courts used to decide cases based on evaluating case facts (i.e., similar conditional dependency of court decisions on case facts), is an interesting and yet unexplored research problem.

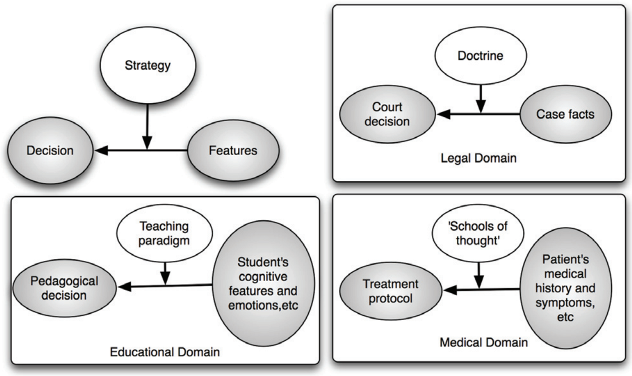


Figure 1: The exemplar relationship among strategy, decision, and features in various domains.

Indeed, judging similarity or difference based on dependency is not unique to law. In politics, presidential administrations are judged to be similar or different based on their strategies used to make decisions – decisions about war and peace, about budgetary funding priorities, and about which political candidate to support along with innumerable other choices. In medicine, judgements on similarity about physicians are based on their strategies used to prescribe a treatment after physicians evaluate a patient's previous medical complications, reported symptoms, results of various tests, etc. The similarity of decision-making software agents is also assessed based on dependency. For example, intelligent tutoring systems (ITS) are *felt* differently by students they teach when they provide pedagogical decisions driven by different teaching paradigms.

When the outcomes of decisions are observable (e.g., radical mastectomy leads to shorter/longer recovery time from breast cancer than lumpectomy does.), uncovering decision-making processes is highly desirable. This is because, once strategies are discovered, they can shed light on how to achieve a desired outcome and avoid an unwanted one, and can also allow for strategy comparison. This new area of research about discovering strategies in decision-making is what we call *Strategy Mining*.

In this paper, we formulate the strategy-mining problem as a clustering problem, called the **latent-strategy problem**. We define the problem below; example domains are illustrated in Figure 1.

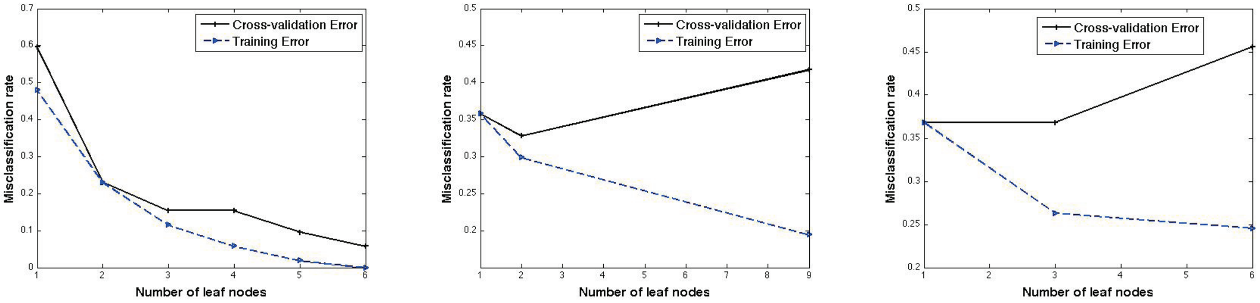
- *Definition*: In a latent-strategy problem, a corpus of data instances I is given, each of which is represented by a set of features F and a decision label D. The inherent dependency of the decision label on the features is governed by a latent strategy S. The objective is to find clusters, each of which contains data instances governed by the same strategy.

In the latent-strategy problem, the clustering target is dependency. Dependency-based clustering differs from conventional object-based clustering in a notable way. Object-based clustering assesses similarity by examining the joint distribution of all features in a non-discriminating feature space, whereas dependency-based clustering evaluates similarity based on the class-conditional distribution in a discriminating feature space.

To the best of our knowledge, no prior work has been done on solving the latent-strategy problem. Existing clustering algorithms are inappropriate to cluster dependency because they either assume feature independency (e.g., K-means (MacQueen 1967)) or only consider the co-occurrence of features without explicitly modeling the special dependency of the decision label on other features (e.g., Latent Dirichlet Allocation (LDA) (Steyvers and Griffiths 2007)).

We propose a baseline algorithm for dependency clustering. Our algorithm is based on the following *assumption*: data instances with similar features but different decision labels come from different conditional distributions.

Our algorithm, in a nutshell, models conditional dependencies with decision trees and iterates between an assign-



(a) AM-DTM

(b) K-means

(c) LDA

Figure 2: Ten-fold cross-validation & training errors of decision trees built using 1 of the 3 data clusters given by (a) AM-DTM, (b) K-means, and (c) LDA. The selected clusters are representative to the general truth of all clusters.

ment step and a minimization step to learn a mixture of decision tree models that represent latent strategies. We call this algorithm Assignment Minimization for Decision Tree Mixtures (AM-DTM). AM-DTM starts from partitioning data into K disjoint datasets, which are used to build K initial decision trees. Techniques should be used to avoid overfitting. The main body of AM-DTM consists of two iterative steps: an assignment step (A-step) and a minimization step (M-step). In the A-step, instances are assigned to clusters based on decision trees' classification results. The assignment strategy is as follows. If an instance in a cluster is correctly classified by the decision tree built from that cluster, it will stay in the original cluster; otherwise, it will move to a cluster whose decision tree correctly classifies it. When there are more than one decision tree that correctly classifies a misclassified instance, that instance will move to the cluster whose decision tree yields the highest classification probability for it. Further ties are broken by preferring the decision tree whose leaf node has a greater number of instances underneath. If there is no decision tree that correctly classifies an instance, that instance will stay in its original cluster. In the M-step, decision tree learning is performed and the total training error of all decision trees is minimized under the assumption that the assignment from the A step is correct. To ensure and speedup convergence, we replace an older decision tree with a new one for a cluster only when the new tree has a lower training error. This process is repeated until no instance is moved. The goal of the iteration is to minimize the overall training error so that the learned decision trees representing coherent concepts can be found accurately. Similar to the Expectation Maximization algorithm (Little and Rubin 2002), AM-DTM follows the empirical risk minimization principle from PAC learning theory.

We carried out a set of experiments to evaluate AM-DTM in a legal domain. Our dataset (Rissland and Xu 2011) contains 151 actual cases taken from a variety of jurisdictions in the United States and in the United Kingdom. Although the legal doctrine used for deciding each case was not recorded at dataset construction, domain knowledge tells us that each case in the dataset was decided by one of three known doctrines. Initial results showed that (1) AM-DTM converged within a few iterations (5 iterations on average of 10 runs), (2) its learned decision trees are compact (5 leaf nodes, on average, which conforms to the fact that legal doctrines are usually not complex rules), with low training errors (0.03 on average) and low cross-validation errors (0.06 on average), (3) the learned decision trees in overall resemble the doctrines well, and (4) AM-DTM significantly outperformed K-means and LDA on clustering dependency as shown in Figure 2 (e.g., decision trees learned from clusters given by K-means and LDA have high cross-validation errors).

AM-DTM has three notable characteristics. First, it is irrelevant-feature resistant, because the decision tree model used by AM-DTM can automatically select key features that significantly influence the decision. Second, it is a glass-box learning algorithm, because one can easily evaluate and explain clustering results by examining the *look* of the learned decision trees. Finally, the outputs of AM-DTM are predictive, because the learned decision trees allow for similarity-based retrieval and classification tasks on new data. In future work, we will develop algorithms that use other non-parametric statistical models, parametric discriminative models, and parametric generative models to cluster conditional dependency, and compare them with AM-DTM.

Acknowledgments

The author is especially grateful to David Jensen for his guidance. The author also thanks Edwina Rissland for her feedback and the reviewers for their constructive comments.

References

Little, R. J. A., and Rubin, D. B. 2002. *Statistical Analysis with Missing Data, Second Edition*. Wiley-Interscience.

MacQueen, J. 1967. Some methods for classification and analysis of multivariate observations. *Proc. of the Fifth Berkeley Symposium on Mathematics, Statistics and Probability* 1(5):281–296.

Rissland, E. L., and Xu, X. 2011. Catching gray cygnets: An initial exploration. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Law (ICAIL)*. To appear.

Steyvers, M., and Griffiths, T. 2007. Probabilistic topic models. *Handbook of Latent Semantic Analysis* 22:424–440.