Proceedings of the Twenty-Third International Florida Artificial Intelligence Research Society Conference (FLAIRS 2010)

Towards an Ensemble Framework for Assisting in Synthesis Tasks*

Joseph Kendall-Morwick
Indiana University
Lindley Hall 215
150 S. Woodlawn Ave.
Bloomington, IN 47405-7104

Introduction

It is a common practice when problem solving to seek the advice of others as to what decision to make next. One may seek advice from a single trusted colleague, but commonly one seeks advice from multiple sources, weighing the advice from each before making a decision (Polikar 2006). This behavior is mimicked by ensemble methods for machine learning which combine outputs from multiple independent components to arrive at conclusions which, if the ensemble was constructed properly, can be more accurate and reliable than its individual components.

Ensemble methods have enjoyed much recent attention from machine learning researchers. Classification systems have been studied extensively through ensemble methods, and though efforts have also been made to study clustering and regression ensembles, some have suggested wider application of ensemble methods (Rokach 2009).

We seek to assist in synthesis tasks involving the design of structures, examples including plans (Kim and Blythe 2003; Aha, Breslow, and Munoz-Avila 2001) and workflows (Leake and Kendall-Morwick 2009). As such tasks can involve a high level of sophistication and complexity, it is not straightforward to apply machine learning techniques aimed towards analytical tasks (e.g., classification, regression) (Aha and Wettschereck 1997). To simplify the design task, incremental refinements are often sought involving explicit aspects of an incomplete or incorrect structure. These refinements can be presented as recommendations in a user-driven process where the AI system provides assistance to a human author. We simplify such recommendations to two core components, the problem and the solution, in order to reduce the problem of generating recommendations into two analytical tasks, described in detail in the following section.

Most studies of ensemble methods have focused on varying the training set, feature set, or random values initializing a single inducer to produce a diverse set of classifiers. Even though there is no change in the inductive bias for each component of the ensemble, ensemble methods can increase the accuracy or diversify the class of decision boundaries learnable by a particular technique. We focus on a different sort of ensemble technique, what is sometimes called hybridization, in order to support synthesis tasks.

Synthesis tasks are complex, often involving a disparate collection of goals and potential problems. Such tasks may also involve several sources of knowledge, such as user-generated content (tags, ratings, etc.), prior completed structures, prior interaction episodes, expert knowledge, etc. One could produce a complex system aimed at encompassing strategies addressing every conceivable aspect of the task, but there is value in involving multiple techniques which may either specialize in incorporating knowledge, such as a case-based reasoner whose cases record interrelationships, or specialize in addressing a particular domain-specific issue, such as a rule-based reasoner. An ensemble framework simplifies the design of such a hybrid system and allows for easy ablation studies and analyses of refinements to components. Additionally, ensemble systems can learn weights for individual components which normalize the support offered from potentially unrelated rules or techniques.

Recommending Problems and Solutions

Such an ensemble framework is motivated by previous work towards building an intelligent assistant for workflow authorship by combining four independent components (Leake and Kendall-Morwick 2009). Through this study we found that the task of identifying and solving problems in a partially-completed workflow can be somewhat cumbersome and components could be further specialized if this task were decomposed into two independent tasks: recommending problems and solutions. This complication is common to synthesis tasks and is explored in this section through a running example comparing a complex yet familiar synthesis task, text editing, with a simple classification task, product recommendation.

In the product recommendation task, the problem to be solved is known: '*What product would this customer most want to buy*', and the solution is simply a product or product category. In contrast, text revisions would not take the form of a simple classification, since such a solution lacks the context required to understand the specifics of the problem to be solved. In the latter case, recommendations of revisions must also provide some further specification of the

*This material is based on work supported by the National Science Foundation under Grant No. OCI-0721674. I thank David Leake, Jay Powell, and the anonymous reviewers for their helpful comments.

problem itself, identifying the relevant context.

Overlapping Problems

Consider the text-editing sub-task of suggesting word replacements, potentially including suggestion of a more appropriate term or avoiding repetitive verbs. For this sub-task, a **problem** can be as general as '*replace a word in this text*' or as specific as '*replace the n-th word in the m-th paragraph in this text*'; in every case the problem consists of a sub-sequence of words in the text, and some problems generalize over other **sub-problems**. Additionally, we can also consider the sub-task as a further specification of the problem, where '*replace a word in this text*' and '*delete a word in this text*' are two unique problems addressable in the text revision task. Recommended problems must be **answerable**, in that they identify all of the context necessary and only the context necessary for providing a **solution**, which, in this case, would be the replacement word.

Withholding Recommendations

An additional consideration is whether any recommendation should be made or not. In the shopping scenario, it is known that the customer is looking to buy *something*. Otherwise, browsing an e-commerce site would not be a very productive use of his or her time. Therefore, a product recommender will make a recommendation if it can, only withholding recommendations in the case that it has insufficient confidence in any recommendation, not because it believes the customer is not interested in making a purchase. However, in the text revision scenario, not every potential problem reflects an actual, existing problem, and additionally one may seek assistance only to confirm a belief that the text is in no need of further revision. Such is also the case in any other incremental design task.

Requirements for an Ensemble Framework

Facilitation of Specialization

We have identified two separate tasks for the framework to address: recommending problems and solutions. Components of the framework can specialize in one or the other. For instance, analyzing the frequency of re-used words can identify some potential word-replacement problems, while a thesaurus can identify potential word-replacement solutions. The framework must provide a clear separation of these tasks and facilitate communication between such components.

Confidence Estimates

Components tasked with identifying problems can err both in falsely identifying a non-problem (false-positive), or by failing to identify an actual, existing problem (false-negative). Adding additional components will reduce the number of false-negatives (increase coverage), but will also increase the number of false-positives, potentially at a greater rate. Adding components to the ensemble should improve performance rather than degrade it, therefore an additional requirement is considered: Recommendations must include a confidence estimate. By ranking recommendations by confidence estimate, the user can selectively view only the most viable problems, avoiding scanning through high numbers of false-positives. The framework is then also responsible for regression over confidence estimates generated by each component to insure comparability and proper weighting.

Contra-recommendation

Another requirement reducing false-positives is allowing components to recommend *against* a problem or solution. Allowing **contra-recommendation** increases the range of strategies that can be incorporated into an ensemble, provides for further decomposition of complex strategies, and increases the benefit of confidence thresholding by reducing the confidence in likely erroneous recommendations. One important concern is the complication this adds to combining confidence estimates, though we have begun to address this issue (Leake and Kendall-Morwick 2009).

Framework Sketch

A framework meeting our requirements would create recommendations in 2 phases, each consisting of 4 steps. Problems are recommended in the first phase, and accompanying solutions are recommended in the second. Each phase is outlined as follows:

1. Each component generates positive recommendations
2. Each component reviews the existing recommendations and may generate contra-recommendations
3. Confidence estimates for each recommendation are normalized through component-specific regression
4. confidence estimates are combined for overlapping or competing recommendations

References

Aha, D. W., and Wettschereck, D. 1997. Case-based learning: Beyond classification of feature vectors. In *ECML '97: Proceedings of the 9th European Conference on Machine Learning*, 329–336.

Aha, D.; Breslow, L.; and Munoz-Avila, H. 2001. Conversational case-based reasoning. *Applied Intelligence* 14:9–32.

Kim, J., and Blythe, J. 2003. Supporting plan authoring and analysis. In *IUI '03: Proceedings of the 8th international conference on Intelligent user interfaces*, 109–116.

Leake, D., and Kendall-Morwick, J. 2009. Four heads are better than one: Combining suggestions for case adaptation. In *ICCBR '09: Proceedings of the 8th International Conference on Case-Based Reasoning*, 165–179.

Polikar, R. 2006. Ensemble based systems in decision making. *IEEE Circuits and Systems Magazine* 6(3):21–45.

Rokach, L. 2009. Taxonomy for characterizing ensemble methods in classification tasks: A review and annotated bibliography. *Computational Statistics & Data Analysis* 53(12):4046–4072.