Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence

Using Domain Knowledge to Systematically Guide Feature Selection

William Groves
Computer Science and Engineering
University of Minnesota
groves@cs.umn.edu

Abstract

The effectiveness of machine learning models can often be improved by feature selection as a pre-processing step. Often this is a data driven process only and can result in models that may not correspond to true relationships present in the data set due to overfitting. In this work, we propose leveraging known relationships between variables to constrain and guide feature selection. Using commonalities across domains, we provide a framework for the user to express model constraints while still making the feature selection process data driven and sensitive to actual relationships in the data.

Motivation: When building prediction models to solve real world problems, feature selection is often not considered directly. Instead, many machine learning algorithms do feature selection implicitly as part of the learning process. If performance is not satisfactory, explicit feature selection can be performed as a pre-processing step. Alternatively, feature selection can be performed ad-hoc by a human, but this is discouraged because of complexity and because humans may (and often do) make sub-optimal selections. We propose a middle road where feature selection is data driven but the search for a better feature set is guided by domain knowledge from the user.

Background: Data driven feature selection has received significant attention and several techniques are used in practice. [Hall, 2000] presents CFS (correlation-based feature selection) to perform a filter-based feature selection using a merit heuristic (Pearson's correlation). The algorithm uses best-first search to incrementally add features. The output of feature selection is used as input to a machine learning algorithm. Wrapper-based feature selection is an approach similar to CFS that uses the desired machine learning algorithm is in-situ as the merit heuristic [Kohavi and John, 1997].

Another approach toward prediction in multivariate domains is to use time-series methods that incorporate time-delayed relationships (i.e. lagged) implicitly. Vector autoregressive moving average (ARMA) and multivariate regression are effective techniques [Martens and Næs, 1992]. ARMA methods can be sensitive to collinear variables especially when there are many variables and the data set size is small, so feature selection is a useful pre-processing step.

My Work: Often the user has knowledge that can facilitate feature selection. In this framework, the user provides 1) a categorization of all features in the domain into distinct *feature classes* and 2) partial ordering constraints between feature classes, this is called the *feature class hierarchy*. We emphasize that the information provided in the feature class hierarchy need not be expert-level knowledge to be effective. Even basic fundamental knowledge of the domain is sufficient to reduce the feature set search and can improve results. Also, constraint relationships can be domain specific: for example, the constraint between two classes (denoted A→B) could mean that A must be included in the feature set before is included B (because A is more relevant to the target).

This information facilitates a search over all feasible combinations of feature classes. Each valid configuration (satisfies the constraints) is called a *lag scheme* and will instantiate a unique feature set. We search over the lag schemes and choose the scheme with the best validation set performance. The purpose of the constraints is to exclude feature selection configurations that would violate domain knowledge. This promotes feature selection efficiency while being careful not to exclude potentially interesting relationships.

The proposed method is also applicable to non-temporal domains as well. In such domains, variables will still have different levels of relevancy with respect to the target which can be encoded as feature class constraints. To encode a non-temporal domain, each feature class would contain at most one possible time lag (the current observation).

This technique is useful in domains with many observable features (variables), regular and time-ordered observations, and known or suspected relationships between features are available to practitioners in the domain.

Completed Work: In [Groves and Gini, 2013], we applied this framework to predicting airline ticket prices. The paper compares existing domain-specific techniques from the literature, data driven feature selection techniques, and our user-guided framework. In the experiments for a specific route, there are 92 input features observed on each day. Also, there are time delayed relationships between some variables: for example, some airlines react slowly when adjusting prices, so the model should reflect this. Experimentally, it was found that time lags of up to 8 days were chosen during the lag scheme search. The feature classes, constraints (for example, A→B denotes that feature class A must have more time

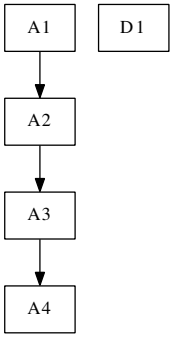

Figure 1: Airline ticket feature class hierarchy

Class	# Vars	Variable List
D1	8	(*deterministic features*, included only once) days-to-departure, day-of-week
A1	3	minimum price, mean price, and quote count for all airlines
A2	9	statistics for non-stop, one-stop and two-stop flights
A3	18	statistics for each airline
A4	54	statistics for each airline and # stops

Table 1: Feature classes from general to specific and the corresponding raw domain variables

(a) New York → Hong Kong (any flight)

Class	Lagged Offsets							
	0	1	2	3	4	5	6	7
D1	•							
A1	•	•	•	•	•	•	•	•
A2								
A3								
A4								

(b) New York → Hong Kong (non-stop only)

Class	Lagged Offsets							
	0	1	2	3	4	5	6	7
D1	•							
A1	•	•	•	•	•	•	•	•
A2	•	•	•	•	•	•	•	
A3	•	•	•	•				
A4	•	•	•	•				

Figure 2: Sample of the best lag schemes found for the airline domain

lags included in the feature set than B), and a sample of best performing lag schemes are shown in Table 1, Figure 1, and Figure 2, respectively. This configuration has 8518 valid lag schemes; without constraints there are 10^{200} ($\approx 2^{(92*8)}$) combinations of the original features. The lag scheme constraints make the search feasible.

The results were computed using simulated purchases guided by the predictions to optimize purchase timing; this technique was also used in [Etzioni *et al.*, 2003]. Two benchmarks used were: the *earliest purchase* cost (purchasing as early as possible to avoid price rises) and the *optimal purchase* policy (best possible result given perfect knowledge). This is a difficult problem: the optimal policy achieves an average savings of 11.0% off of the earliest purchase for 7 routes under study. Using purely data-driven feature selection, the best method (CFS) achieves only a 0.687% reduction. Using our method, the savings is 7.25%, a large savings. Beyond prediction performance, the lag scheme search can provide domain knowledge. In Figure 2, two prediction targets on the same route are shown. The more specific target (b) requires more features to achieve its best performance.

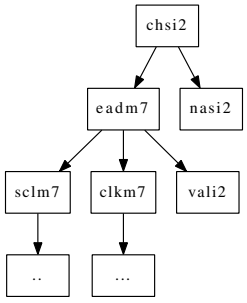

Figure 3: Physical relationships of observation sites for downstream prediction target CHSI2

Future Work: As a next step, we propose to apply the lag scheme search to stream flow prediction. Stream flow is a natural domain for the lag scheme constraints because there is a directed graph to describe the water flows between observation sites, a portion of a river network is provided in Figure 3. Each gauge site is represented by a feature class and the feature class constraints recognize the natural temporal relationships (for example, a downstream site should have a time lag that is less than the time lag on an upstream site).

Sometimes the structure of the lag scheme constraints may not be enough to make the feature selection search tractable. In the stream flow domain, an exhaustive search of all lag schemes can be infeasible if the number of feature classes is large. In future work, we will explore the greedy approaches, often used by data-driven methods, for lag scheme search when necessary for tractability and efficiency.

Conclusions: There are two primary contributions of this work, and we believe that it represents an alternative to traditional feature selection. First, keeping information that is known apriori (the model constraints) can facilitate better and more reliable machine learning models. Statistical methods (such as cross-validation) and resampling methods (such as boosting) can also address these risks but can inject noise when the number of observations is small. Second, relationships observed in the lag scheme output can be meaningful for domain practitioners: specifically, the method can elucidate unusual or unexpected relationships found in the data.

References

[Etzioni *et al.*, 2003] Oren Etzioni, Rattapoom Tuchinda, Craig Knoblock, and Alexander Yates. To buy or not to buy: mining airfare data to minimize ticket purchase price. In *SIGKDD Conf. on Knowledge Discovery and Data Mining*, pages 119–128, 2003.

[Groves and Gini, 2013] W. Groves and M. Gini. Optimal airline ticket purchasing using automated user-guided feature selection. In *IJCAI '13: Proc. 23rd Int'l Joint Conf. on Artificial Intelligence*, 2013.

[Hall, 2000] Mark A. Hall. Correlation-based feature selection for discrete and numeric class machine learning. In *Int'l Conf. on Machine Learning*, pages 359–366, 2000.

[Kohavi and John, 1997] Ron Kohavi and George H. John. Wrappers for feature subset selection. *Artif. Intell.*, 97(1-2):273–324, 1997.

[Martens and Næs, 1992] Harald Martens and Tormod Næs. *Multivariate Calibration*. John Wiley & Sons, July 1992.