Tour-Based Mode Choice Modeling: Using An Ensemble of (Un-) Conditional Data-Mining Classifiers

James P. Biagioni*
Ph.D. Candidate
Department of Computer Science
University of Illinois at Chicago
851 South Morgan Street, Chicago, IL 60607
Phone: (312) 355-0349
E-Mail: jbiagi1@uic.edu

Piotr M. Szczurek
Ph.D. Candidate
Department of Computer Science
University of Illinois at Chicago
851 South Morgan Street, Chicago, IL 60607
Phone: (312) 355-0349
E-Mail: pszczu1@uic.edu

Peter C. Nelson, Ph.D.
Professor, Department of Computer Science
Dean, College of Engineering
University of Illinois at Chicago
851 South Morgan Street, Chicago, IL 60607
Phone: (312) 996-2400
Fax: (312) 996-8664
E-Mail: nelson@uic.edu

Abolfazl Mohammadian, Ph.D.
Associate Professor
Department of Civil and Materials Engineering
University of Illinois at Chicago
842 West Taylor Street, Chicago, IL 60607
Phone: (312) 996-9840
Fax: (312) 996-2426
E-Mail: kouros@uic.edu

Paper submitted for potential presentation and publication at the 88th Annual Meeting of the Transportation Research Board

August 1st, 2008

Word Count: 5,430 + 2 Tables + 6 Figures = 7,430 words

* Corresponding author

Tour-Based Mode Choice Modeling: Using An Ensemble of (Un-) Conditional Data-Mining Classifiers

James P. Biagioni, Piotr M. Szczurek, Peter C. Nelson and Kouros Mohammadian

ABSTRACT

This study aims to take the lessons learned from the history of applying data-mining techniques to mode choice modeling and extend it with the characteristics inherent to tour-based datasets. In doing so, a novel adaptation of existing data-mining methods is developed through the use of an ensemble of conditional and un-conditional classifiers. By defining the notion of an "anchor mode" as the mode selected on the first trip of a tour, this ensemble of classifiers is trained with and without knowledge of the anchor mode respectively. This allows the un-conditional model to make mode predictions without pre-condition for the first trip on a tour, followed by the conditional model which then makes mode predictions for the subordinate trips on a tour, given the knowledge of the selected anchor mode from the previous trip. This method was tested on the new Chicago Travel Tracker Survey dataset, and prediction performance was evaluated across four different data-mining algorithms where the best performing solution was arrived at using a combination of Naïve Bayes for the un-conditional classifier and C4.5 for the conditional classifier. Performance was measured using metrics from the field of information retrieval, and able to demonstrate an appreciable gain by using this method. For the purposes of evaluating this technique compared to traditional discrete choice methods, (un-) conditional multinomial logit models were also constructed and compared to the data-mining based solution. While the performance of the multinomial logit was reasonable, the data-mining solution proved to have better prediction performance overall.

KEYWORDS

Data Mining, Ensemble Method, Travel Mode Choice, Decision Trees, Naïve Bayes, Logistic Regression, Support Vector Machines, Multinomial Logit Models

INTRODUCTION

Mode choice modeling is an integral part of the four-step travel demand forecasting procedure, and as such, has received tremendous attention from statisticians in an attempt to predict the distribution of mode selection across a population. For years, discrete choice models (*1*) have dominated this area of research, however, more recently increasing attention has been paid to data-mining techniques borrowed from the artificial intelligence and machine-learning communities (*2*, *4*, *5*, *6*, *7*), who boast a menagerie of powerful predictive models capable of handling the same multi-attribute data used by traditional models.

This study extends the long lineage of existing research that has successfully applied data-mining methods to the problem of mode choice modeling in comparison to existing statistical techniques. An early example of this comparison can be found in Wets et al. (*2*) where they compare the performance of decision-tree algorithms C4 and CHAID against the multinomial logit model (MNL), for predicting mode choice across the activity-based travel-diary data collected for the development of the Albatross (*3*) system. The data was collected from 1,500 households, of which 2,974 person-day diaries were selected and augmented with complementary data, including shortest-route travel times. After building the three models, experimentation determined that while there was no appreciable difference in terms of prediction accuracy, that the decision-tree methods tend to be more robust than the MNL model, and benefit from being free from predefined utility functions.

Hensher and Ton (*4*) then compared the performance of nested logit models against that of artificial neural networks, for the purpose of predicting commuter mode choice. Models were built based upon a stated-choice experimental dataset collected in Sydney and Melbourne, Australia, as part of a broader effort to examine the potential impact of two new transport modes. The researchers point out that while the nested logit models outperform neural networks at an aggregate level, neural networks outperform nested logit models at an individual level – giving no clear indication as to which method is better overall. Despite this ambivalence however, they do note that neural network models might be preferable in certain situations, as they feature a greater resistance to noise in the data.

In Xie et al. (*5*) they compare the performance of decision-tree algorithm C4.5 and the multi-layer feed-forward neural network (MLFNN), against the multinomial logit model, for the purpose of predicting work travel mode choice. Models were built on the two-day travel diaries of the San Francisco Bay Area Travel Survey (BATS) 2000 data set, using two sets of variables: individual/household socio-demographic and trip level-of-service. Experimental results between the data mining techniques show that although the MLFNN model outperforms the C4.5 decision-tree in terms of prediction accuracy, it shows worse transferability due to overfitting the data. When compared to the MNL model, experiments show similar performance across all three methods on an aggregate dataset, while the MLFNN model performs best on an individual prediction level. The authors conclude that both data-mining methods outperform the MNL model overall, with the MLFNN model being best for prediction accuracy, while the C4.5 model best for interpretability.

Zhang and Xie (*6*) then compared the performance of support vector machines (SVM) and multi-layer feed-forward neural networks (MLFNN) against the multinomial logit model, for the purpose of predicting commute mode choice over the San Francisco Bay Area dataset, consisting of 5029 home-to-work commute trips. This paper is the first to introduce SVM's for the purpose of mode choice modeling, driven by the theoretical underpinnings of the method, which ensure a globally optimal solution and greater generalization ability than other existing machine-learning algorithms. In their experiments they find that SVM has better performance than the other two models in predicting mode choice; even though MLFNN's fit the training data better, they suggest it might be overfitting as its predictions on unseen data are worse. The authors favor the fact that SVM's do not assume a model structure apriori, but dislike the fact that it is a black box model – especially when compared to the transparency of the MNL model. They suggest that work on sensitivity analysis of neural networks might be applied to SVM's in order to determine its econometric features.

Lastly, Moons et al. (*7*) compared the performance of support vector machines (SVM) and classification and regression trees (CART) against (semi-) linear statistical models, for the purpose of predicting mode choice across the activity-based travel-diary data collected for the development of the Albatross (*3*) system. The dataset, which consists of 1025 observations, is split into three sets: one for prediction of slow transport, one for prediction of public transport, and one for prediction where the automobile is used to drive. Explanatory variables include person and household characteristics, as well as characteristics of the work pattern. In their experiments on the public transport dataset they find that the logistic regression model provides the best prediction quality, with SVM coming in second (although with obvious overfitting problems), followed by CART. On the slow transport dataset, SVM performs the best with no obvious overfitting this time, followed by logistic regression in second and CART in third. On the auto-drive dataset they again find that SVM performs the best, this time followed by CART, with parametric models performing worst. The authors conclude that on very skewed datasets the (semi-) linear models usually outperform SVM and CART, while on better balanced datasets the performance of SVM and CART is comparable, and often somewhat better than the (semi-) linear models.

This study aims to take the lessons learned from the history of applying data-mining techniques to mode choice modeling and extend it with the characteristics inherent to tour-based datasets. Close attention has been paid to the relationship between mode and trip-tours in the transportation literature, in particular Cirillo and Axhausen (*8*) analyzed mode choice at the tour level using the Mobidrive data set, which consists of six-week travel diaries from 145 individuals and 67 households. Examination of the dataset revealed that travelers tend not to change modes during a tour, or even for a whole day – a finding which supports the long-held belief that travelers maintain their mode during a tour, especially if they use an individual vehicle (car, motorcycle or bicycle). The authors suggest that mode choice models take this into consideration when building predictive models, going so far as to build their own discrete choice models wherein the assumption is explicitly made that travelers do not switch modes within a tour.

In light of these findings, Miller et al. (*9*) constructed a tour-based model of travel mode choice, using an agent-based architecture to represent individual trip-makers within the context of their household demands. This model establishes the concept of an anchor point, which is used as a decision point for individual agents to decide whether or not to take a vehicle for their current tour. If the agent decides to take a vehicle, they are bound to using it for the entire trip-chain, whereas if they select another mode of transportation they are free to select any other mode independently throughout the tour. In addition to this mode selection logic, the model also takes into account household demands in the form of vehicle allocation (whereby overall household utility is maximized if there is contention over which agent should take the vehicle), ride-sharing and joint travel tasks (i.e. where household members share the auto for whole/part of a tour). The authors' prototype implementation featured the mode-choice and vehicle allocation logic (leaving the ride-sharing and joint travel components for future work), allowing it to work successfully as part of a microsimulation framework.

This study investigates a novel adaptation of the data-mining methods aforementioned, by using an ensemble of (un-) conditional classifiers, focused around the notion of an anchor mode for tour-based trips, in order to predict mode choice at the trip level. This research aims to provide a boost in predictive power for mode choice modeling using data-mining techniques, without necessitating the development of an explicit agent-based framework such as that found in Miller et al. (*9*). For the purposes of evaluating this technique compared to traditional discrete choice methods, (un-) conditional multinomial logit models are also constructed, and a performance comparison to the data-mining method given.

(UN-) CONDITIONAL CLASSIFIERS

Taking the key lesson learned from Cirillo and Axhausen (*8*) – that travelers tend not to change modes during a tour – the notion of an "anchor mode" is used in this study. Similar to the concept of an "anchor point", as used in Miller et al. (*9*), anchor mode simply refers to the mode selected when departing from an anchor point (usually home). To incorporate this information into the raw data, it is augmented to include the anchor mode attribute for all subordinate trips in a tour.

A distinction must be drawn between the way "anchor mode" is used in this study, as opposed to the way it is used in Cirillo and Axhausen (*8*): namely, rather than mandating that each subordinate trip in a tour use the anchor mode as the selected mode, it is simply included as an additional attribute in the dataset, allowing the data-mining algorithms to draw data-driven conclusions about the relationship between anchor mode and the selected mode for a trip, rather than assuming a 1:1 correspondence exists apriori. Such an assumption would be particularly undesirable for cases where a mode such as public transport is chosen as the anchor mode, as Cirillo and Axhausen (*8*) point out: a lot of mode variability exists in such tours, as travelers can easily include walking or taxi trips as they do not have to return to their parked vehicles.

The key component of the mode choice prediction method developed in this study, is the notion of using an ensemble of un-conditional and conditional classifiers. "(Un-) conditional" classifiers, refers to the manner in which they are used for classifying instances: namely, an un-conditional classifier is used to predict the selected mode used for the first trip on a tour without any pre-conditions (i.e. where no anchor mode exists), followed by the conditional classifier, which then incorporates (i.e. is conditioned on) the predicted anchor mode in order to predict the selected mode used for each subsequent trip on the tour. In this way, the model incorporates the anchor mode in a flexible, data-driven (i.e. it relies on the relationship derived between anchor and selected mode by the data-mining algorithms) manner, that doesn't necessitate the development of an explicit agent-based mode-choice framework.

DATA MINING

In order to build the un-conditional and conditional classifiers used in this study, four different data-mining algorithms were leveraged and tested with respect to their predictive performance: decision trees, naïve bayes, simple logistic and support vector machines. A description of each, including pointers to further information, follows.

Decision Trees

Decision trees, one of the most widely used data-mining techniques, are constructed by means of repeated attribute partitioning. At each level of the tree (starting with the root), the algorithm selects the attribute whose partitioning will maximize the class-homogeneity among the data instances being used for construction (typically through the use of a heuristic function, such as information gain ratio). The ultimate goal of the decision-tree algorithm is to partition all instances into purely homogenous subgroups – allowing the series of partitions to form convenient If-Then rules from root to leaf, that fully-describe all of the instances contained therein. In order to avoid overfitting the training data, the tree is then typically pruned in order to increase the generalizability of the decision structure. See (*2*, *5*, *6*, *8*, *10*) for more information.

Naïve Bayes

The Naïve Bayesian approach to data-mining takes a purely probabilistic perspective on things. Rather than attribute partitioning (like decision-trees), classification is simplified into the task of estimating class posterior probabilities, i.e. for an example d (a vector of attributes: $<A_1 = a_1, A_2 = a_2, \ldots A_n = a_n>$), compute $\Pr(C = c_j \mid d = <A_1 = a_1, A_2 = a_2, \ldots A_n = a_n>)$, for all classes c_j and see which one is most probable (*11*). By making the assumption of conditional independence (i.e. all attributes are conditionally independent given the class $C = c_j$), Bayes' rule simplifies into:

$$\Pr(C = c_j \mid d = <A_1 = a_1, A_2 = a_2, \ldots A_n = a_n>) = \Pr(C = c_j) \prod_{i=1}^{n} \Pr(A_i = a_i \mid C = c_j)$$

Since we only need a decision on the most probable class for each instance, we only keep the numerator (since the denominator is the same for every class). From here, the prior probabilities $\Pr(C = c_j)$ and conditional probabilities $\Pr(A_i = a_i \mid C = c_j)$ can be easily estimated from the data by occurrence counts (for nominal attributes), and class predictions made. In case of numeric attributes, the probability density function needs to be estimated. For this, the use of kernel density functions is used in this work. See (*10*) for more information.

Simple Logistic

The Simple Logistic method, as its name implies, is based on simple linear logistic regression. However, as a means to boost performance, the logistic regression model is supported by the LogitBoost algorithm (*11*) which fits a succession of logistic models, each of which learns from the classification mistakes made by the previous model, in order to fine-tune the model parameters and find the best (least error) fit. The LogitBoost algorithm also performs cross-validation on the dataset as a means to automatically select the best attributes for prediction, resulting in a simplified, best-fit logistic regression model. See (*11*) for more information.

Support Vector Machines

Support vector machines (SVMs) are binary classifiers that work by finding the maximum margin hyperplane that can separate two classes. Finding the maximum margin hyperplane is posed as a quadratic programming optimization problem which can be solved relatively efficiently. To handle cases where data is not linearly separable, the use of soft margins is employed, which modify the formulation of the problem to allow for misclassifications in the data. The degree to which misclassifications are allowed is specified via a tunable complexity parameter. Additionally, the so-called “kernel trick” can be used in support vector machines to allow non-linear separation boundaries. The “kernel trick” refers to replacement of the dot product used in the problem formulation with a kernel function, which allows for transforming the data into a higher dimensional space. The idea is, that data which is not linearly separable in the given dimension, will become linearly separable in some higher dimensional space. Support vector machines are thus capable of handling a variety of data in a robust manner. While being inherently binary classifiers, SVMs can also be used for multi-class problems by using techniques such as pairwise coupling. See (*12*, *13*) for more information.

Ensemble Method

In an attempt to increase the accuracy of classification, ensemble methods build multiple classifiers and use their outputs as a form of voting for final class selection. One of the most popular of these methods is known as AdaBoost. This algorithm works by training a sequence of classifiers, each of which is dependent on the previous one by re-weighting the dataset to focus on the previous one's errors (*10*). Typically, those examples that are classified incorrectly are given higher weights. Classification is performed by passing each example to the set of previously built classifiers, and combining their (weighted) output to determine the final class. See (*10*, *14*) for more details.

DATA

In order to build the classification models developed in this study, a large activity-based dataset was drawn from the Chicago Travel Tracker Survey, provided by the Chicago Metropolitan Agency for Planning (CMAP). It consists of 1- and 2-day activity diaries from 32,118 people among 14,315 households in the 11 counties neighboring Chicago, and features 35,548 tours decomposed into 218,005 trip-links, complete with rich socio-demographic and trip-based attributes. This study marks one of the first uses of this data for building experimental mode-choice models.

Pre-Processing

The raw survey data was pre-processed (i.e. cleaned) to remove attributes and rows that contained a large number of missing values. Great care was taken to ensure that complete tours were preserved – thus, those rows that contained missing values were removed along with their containing tour in order achieve this goal. As a result of the pre-processing steps completed, the cleaned data used for experimentation contained a total of 19,118 tours decomposed into 116,666 trip-links.

Supplementary Data

In order to supply values for non-selected modes, supplementary data was obtained from CMAP that contained inter-zonal travel times. Based on the geocoded activity-locations in the Travel Tracker Survey data, a mapping between these two datasets was easily established.

Dependent Variable

The dependent variable in this study is the selected mode for a given trip. From the Travel Tracker Survey data eight modes were identified for predictive modeling: walk, bike, auto-drive, auto-passenger, Chicago Transit Authority (CTA) bus (i.e. urban bus), CTA train (i.e. urban train), Pace bus (i.e. suburban bus), Metra train (i.e. commuter rail).

Independent Variables

Three sets of independent variables were used in this study: 1) trip-level variables: *departure time, arrival time, activity duration, primary trip purpose* and *total number of people traveling*; 2) mode-specific variables: *travel time* and *average out-of-pocket cost*; and 3) individual/household socio-demographic variables: *gender, age, disability status, driver's license status, employment status, student status, number of vehicles in the household, age of vehicles, number of bicycles in the household, number of people in the household* and *household income*.

CLASSIFIER IMPLEMENTATIONS

Data Mining Toolkit

For the purpose of implementing the data mining algorithms, the Weka Data Mining Toolkit (*15*) API was leveraged within a custom-built Java application. This API provides common implementations of all of the data mining methods used in this paper. Parameter selection for the algorithms was carried out by experimentation in order to optimize classification performance.

Discrete Choice Model

The discrete choice modeling software, Biogeme (*16*), was used for building the multinomial logit models in this study. Both conditional and unconditional MNL model specifications were developed, with and without the anchor mode attribute respectively. They were then tested in an experimental framework that combines and evaluates the MNL models in a manner identical to that of the data mining algorithms, on the same tour-based dataset.

PERFORMANCE MEASURES

In order to appropriately evaluate the performance of the data-mining algorithms and multinomial logit model, three metrics from the information-retrieval (IR) literature will be leveraged: accuracy, precision and recall. Precision and recall are commonly used when interest centers around classification performance on a particular class (*10*) – in this case, we are interested in the classification performance on 8 particular classes (modes) – as it allows a fine-grained view of how precise and complete the classification is. Accuracy complements precision and recall as it provides an aggregate representation of performance across all classes.

Accuracy

Whereas precision and recall allow classification performance to be evaluated at an individual-class level, accuracy essentially summarizes these statistics into one comprehensive measure that describes the percentage of correctly classified instances in a dataset. It is computed as the number of *correctly* classified instances, divided by the total number of instances classified (*10*).

Precision

Precision measures the proportion of correctly classified instances among all of those instances that were similarly classified (*18*). It is computed as the number of *correctly* classified instances of a particular class A, divided by the total number of instances classified as class A (*10*).

Recall

Recall measures the proportion of instances of a particular class that are correctly classified (*18*). It is computed as the number of *correctly* classified instances of a particular class A, divided by the total number of actual class A instances in the dataset (*10*).

For the purposes of evaluating performance on mode choice prediction, recall is the most important evaluation metric of the three, as it precisely measures the behavior we are striving to achieve: being able to correctly identify the distribution of trips among the modes. It is superior to the accuracy measure, since the mode choice problem is not so much a classification task, but rather, a problem of distribution estimation. As such, a good measure of performance for this problem would clearly be a sum of the deviation for each mode from the real distribution, which is captured by the mean recall measure.

EVALUATION

The classifiers were evaluated in several stages. To test the usefulness of the anchor mode (ACRMODE) attribute for determining mode of transportation for a given trip, several classifiers were built with and without knowing the anchor mode. While in reality, the anchor mode will never be known with 100% certainty, these tests provided the upper bound for any expected performance gain that could be achieved. The classifiers tested were: C4.5 decision trees, Naïve Bayes, Simple Logistic, and SVM. The C4.5 decision trees were built using the default parameters supplied by the Weka software, which includes a confidence factor of 0.25 and minimum of 2 instances per leaf. Naïve Bayes classifier was implemented using kernel density functions for numeric attributes. The Simple Logistic classifier also used default Weka values: cross validation is used and the maximum number of boosting iterations is 500. Support Vector Machines employed default values from Weka, except for the following: logistic models were built to support proper probability estimates. Several kernel functions were tested, with the linear kernel demonstrating the best overall performance.

The results, shown in Figures 1 through 3, show that the anchor mode is, in fact, a key attribute for determining mode for a given trip. For every performance measure, knowing the anchor mode improves the classification performance of all tested techniques.

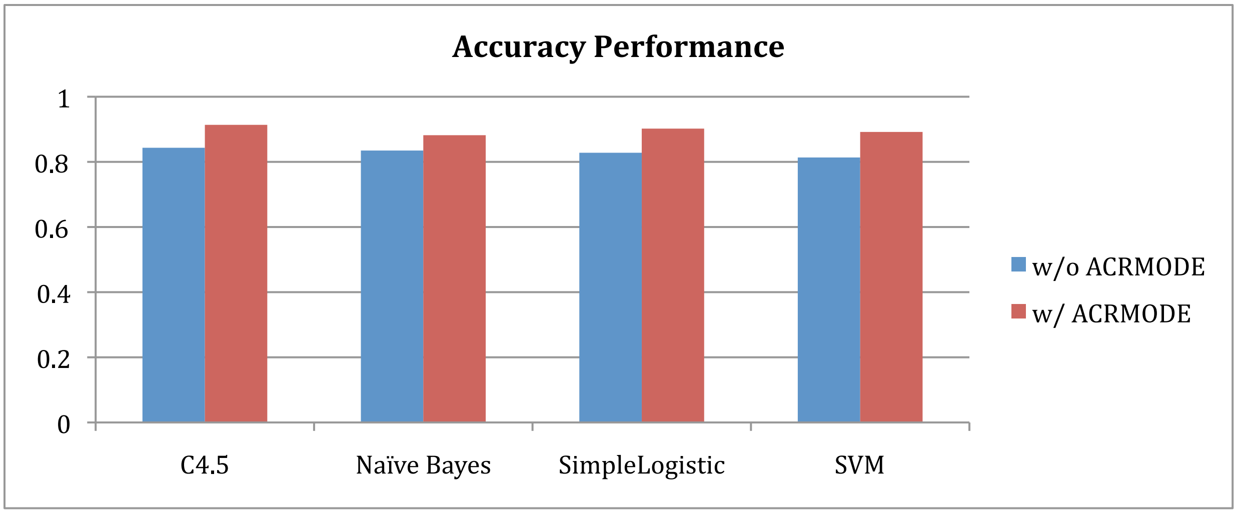

FIGURE 1 Accuracy Performance with and without anchor mode attribute.

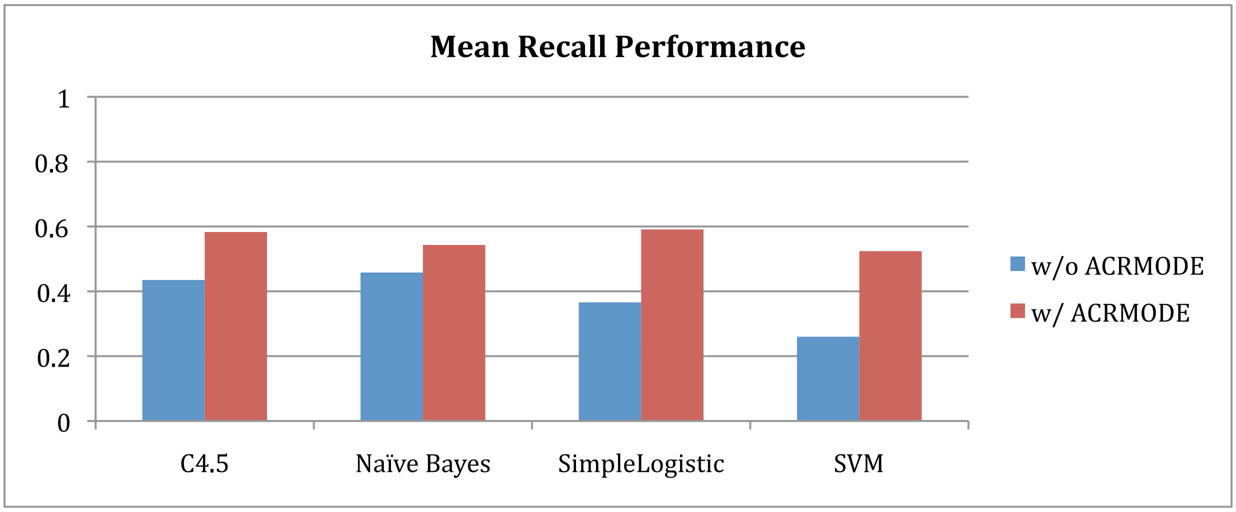

FIGURE 2 Mean recall performance with and without anchor mode attribute.

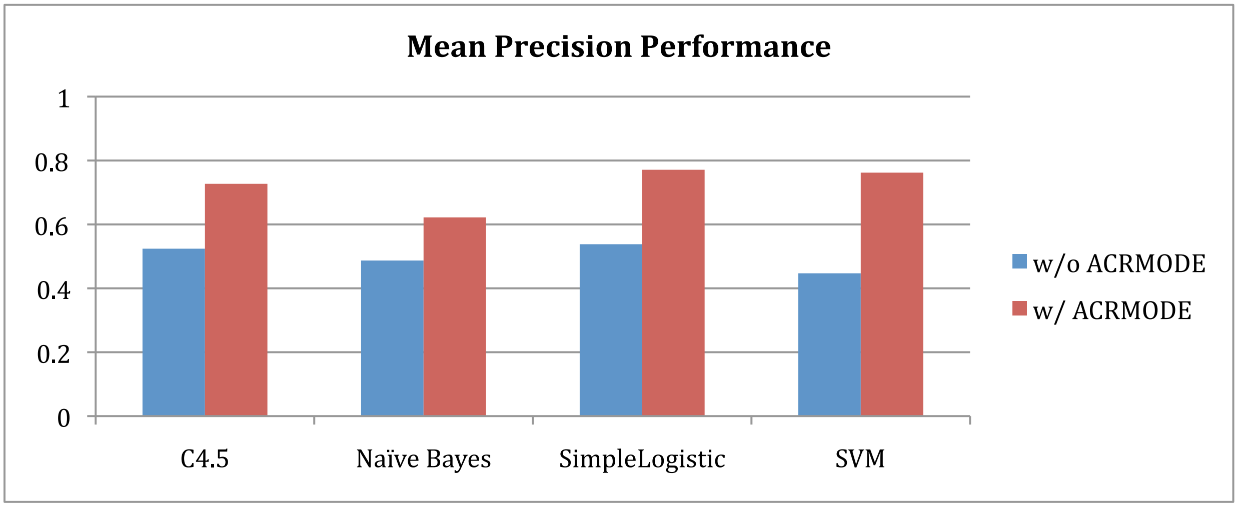

FIGURE 3 Mean precision performance with and without anchor mode attribute.

Given the performance gains shown by these tests, a second stage of testing was performed using the (un-) conditional models. As previously described, in this method, two classifiers are used: one (un-conditional) for estimating the mode on the first trip and another (conditional) for estimating the mode on subsequent trips. In the initial tests, both the conditional and unconditional classifiers are of the same type. All types of classifiers used in the first stage of testing were used for these tests, along with the AdaBoost classifier. Two versions of the AdaBoost classifier were used: one was built using C4.5 decision trees, the other employed naïve Bayes classifiers. The results of the tests are shown in Figures 4 through 6.

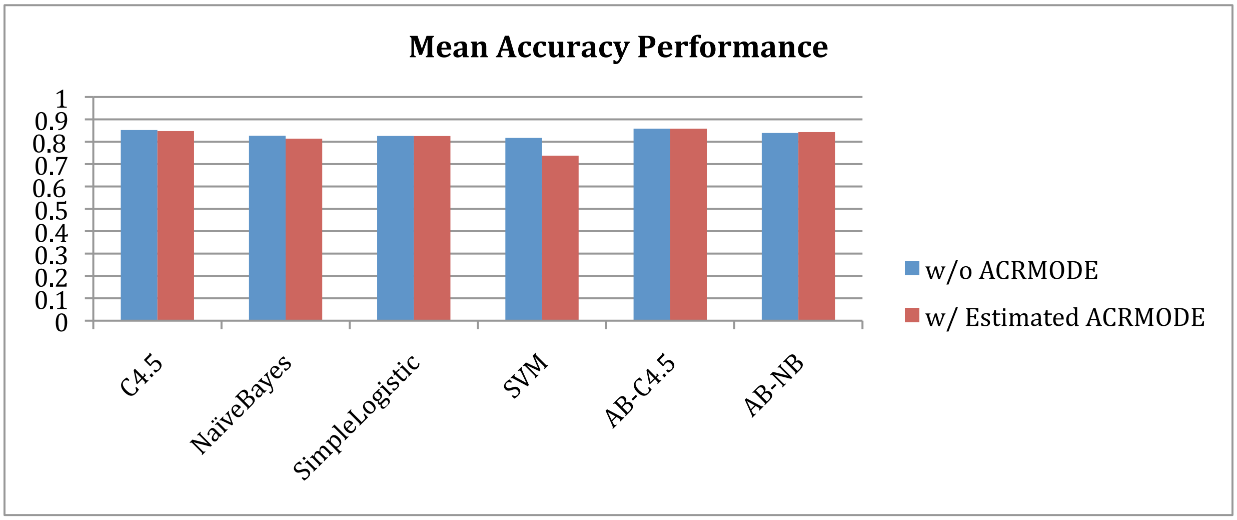

FIGURE 4 Mean accuracy performance with and without estimated anchor mode.

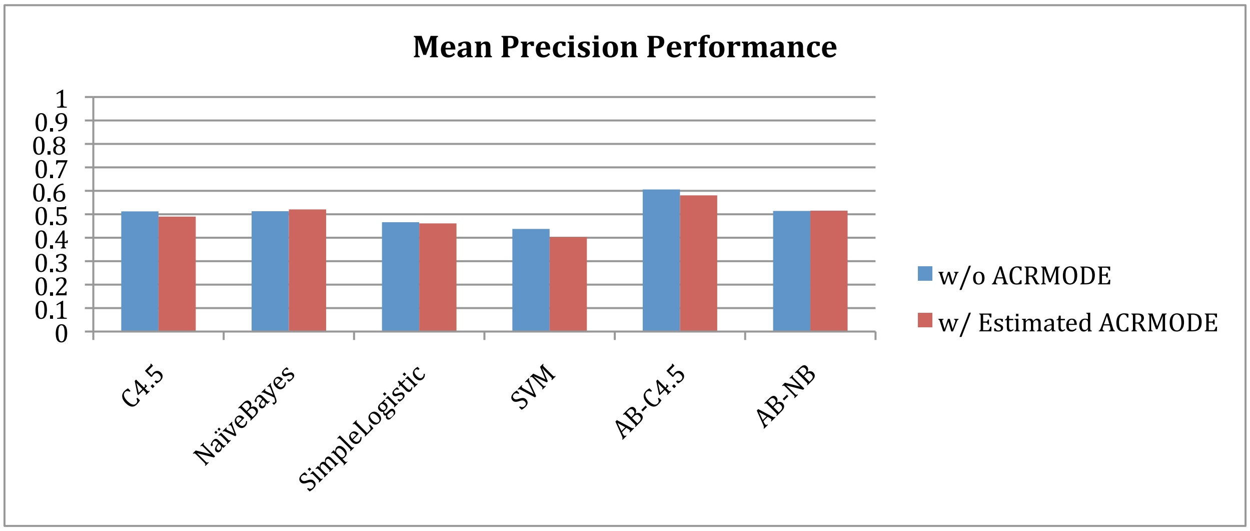

FIGURE 5 Mean precision performance with and without estimated anchor mode.

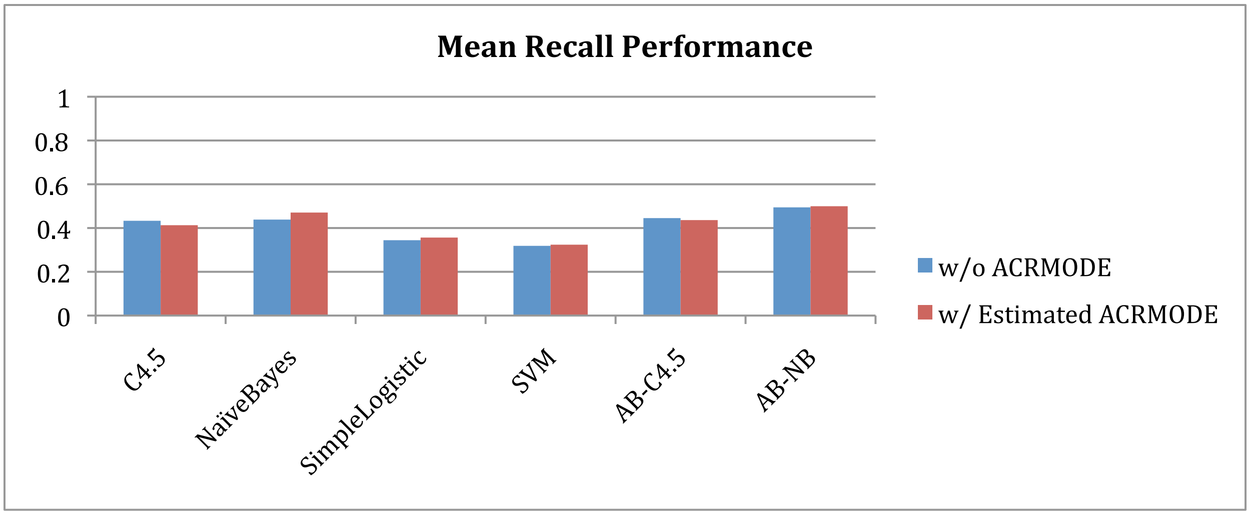

FIGURE 6 Mean recall performance with and without estimated anchor mode.

The results, as can be seen in the figures, show that using the (un-) conditional models did not provide any major improvements in performance. In fact, the performance measures of most classifiers are actually lower. Despite this, the results do show that using AdaBoost with C4.5 decision trees provides the best accuracy (85.85%), and AdaBoost with Naïve Bayes provides the best mean recall (49.96%). This observation suggested the idea of using different techniques for the conditional and un-conditional classifiers in order to combine the high accuracy of AdaBoost-C4.5 with the high recall value of AdaBoost-Naïve Bayes. To evaluate this method, a number of combinations of classifiers were used. The best performance was achieved when using AdaBoost-NaiveBayes as the unconditional classifier and AdaBoost-C4.5 as the conditional classifier (AB-NB/AB-C4.5). Other combinations were able to achieve similar performance in terms of accuracy, but could not match the mean recall of AB-NB/AB-C4.5, which was over 4% higher when compared to the next highest performing model. Table 1 summarizes the performance of these tests and compares it to the highest performing classifiers from the previous tests.

Classifier	Description	Accuracy	Mean Recall	Mean Precision
AdaBoost-C4.5 (w/ o ACRMODE)	Best accuracy without using ACRMODE attribute	85.85%	0.4453	0.6058
AdaBoost-NaiveBayes (w/ o ACRMODE)	Best recall without using ACRMODE attribute	83.92%	0.4945	0.5142
AdaBoost-C4.5 (w/ est. ACRMODE)	Best accuracy with estimating ACRMODE by an unconditional classifier	85.84%	0.4363	0.5807
AdaBoost-NaiveBayes (w/ est. ACRMODE)	Best mean recall with estimating ACRMODE by an unconditional classifier	84.31%	0.4996	0.5154
C4.5 uncond / SimpleLogistic cond	Best accuracy using a combination of different classifiers	84.85%	0.4202	0.5050
AdaBoost-NaiveBayes uncond / AdaBoost-C4.5 cond	Best mean recall using a combination of different classifiers	84.79%	0.4903	0.5665
NaiveBayes uncond / C4.5 cond	2nd best mean recall using a combination of different classifiers	83.28%	0.4444	0.5134

TABLE 1 Data-mining algorithm performance comparison.

Looking at Table 1, one can see that while the AB-NB/AB-C4.5 combination did not achieve the highest performance across all of the measures, its values were relatively close to the highest. Its recall value of 0.4903 was only marginally lower than the best recall value of 0.4996 achieved by the AdaBoost-NaïveBayes (w/ est. ACRMODE) classifier, while at the same time having a much higher precision and somewhat better accuracy. Similarly, the accuracy was not the highest at 84.79%, yet not much lower than the best accuracy value of 85.85%. This accuracy was also achieved with much higher recall than the AdaBoost-C4.5 (w/ o ACRMODE) classifier which achieved the 85.85% accuracy. The AB-NB/AB-C4.5 combination was thus able to combine high accuracy with high recall simultaneously, making this the best overall classifier that was tested.

In the last stage of experimentation, the AB-NB/AB-C4.5 model was tested against a multinomial logit (MNL) model. In a manner equivalent to the data mining algorithms, both conditional and un-conditional models were built and evaluated. Due to word-count limitations the MNL model results table is not presented, but its parameters and goodness-of-fit measures are briefly discussed. The un-conditional model used the following attributes: arrival time, activity duration, primary trip purpose, total number of people traveling, travel time, out-of-pocket cost, gender, age, disability status, driver's license status, student status, number of vehicles in the household, number of people in the household and household income. For the conditional model the anchor mode attribute was used along with travel time and the total number of people traveling. Attribute selection was based on t-test significance statistics ($t > 1.96$) and intuition. The adjusted rho-squared (ρ^2) goodness-of-fit values were 0.684 and 0.691 for the un-conditional and conditional models respectively. The results, compared to the AB-NB/AB-C4.5, are listed in Table 2.

Classifier	Description	Accuracy	Mean Recall	Mean Precision
MNL-w/o ACRMODE	MNL model that does not use the anchor mode attribute	79.25%	0.2838	0.4557
MNL-w/ est. ACRMODE	Combination of 2 MNL models (conditional and unconditional)	78.08%	0.2592	0.4224
AdaBoost-NaiveBayes uncond / AdaBoost-	Best overall data mining technique	84.79%	0.4903	0.5665

TABLE 2 Best data-mining algorithm vs. MNL model performance comparison.

As can be seen in Table 2, the data-mining algorithm significantly outperforms the traditional MNL model in every performance category – despite the use of (un-) conditional MNL models (which actually lowered performance slightly). Most importantly, the largest performance margin is in the recall value, in which AB-NB/AB-C4.5 achieved a higher mean recall by almost 0.21. This, along with higher accuracy and precision, indicates that the data mining model has the potential to be much more generalizable in practice, and should be seen as a viable method for complementing the existing MNL modeling technique.

While there still exist questions regarding the interpretability and usefulness of many data mining models for use in transportation forecasting, the outstanding prediction capabilities and large reduction in human computation spent crafting model-structures, should be seen as a great advantage of the AB-NB/AB-C4.5 data mining technique.

CONCLUSIONS

The purpose of this study was to extend upon the existing body of work that applies data-mining algorithms to the problem of mode choice modeling. By incorporating the concept of "anchor mode" as a new attribute upon which data-mining algorithms can learn, a novel solution was devised: the idea of using an ensemble of conditional and un-conditional classifiers. Under this framework, this ensemble of classifiers is trained with and without knowledge of the anchor mode respectively, allowing the un-conditional model to make mode predictions without pre-condition for the first trip on a tour, followed by the conditional model which then makes mode predictions for the subordinate trips on a tour. This method was tested on the Chicago Travel Tracker Survey dataset, and prediction performance was evaluated across four different data-mining algorithms: C4.5 decision-trees, Naïve Bayes, Simple Logistic and Support Vector Machines. Ultimately, the solution with the best predictive performance was arrived at, using a combination of Naïve Bayes for the un-conditional classifier and C4.5 for the conditional classifier.

The results of this study have shown that data mining approaches are a viable option for the problem of mode choice estimation. The AB-NB/AB-C4.5 combination of classifiers achieved a high level of accuracy, precision, and recall, clearly outperforming the MNL model that is traditionally used. Most importantly, the recall performance, which in this paper is argued to be the best measure of performance, is higher by a large margin. AB-NB/AB-C4.5 also outperformed other data mining algorithms by leveraging the anchor mode attribute, resulting in a combination of excellent recall performance, along with high levels of accuracy and precision.

Although data mining algorithms have been applied to the mode choice problem in the past, this study shows that the margin of performance over MNL is higher than might have been previously thought. As such, it may be advantageous to consider using both techniques as complementary tools, allowing the MNL model and proposed AB-NB/AB-C4.5 approach to support one another in the task of mode choice prediction. Doing so may allow for building better, more accurate, travel forecasting models. Since questions have been raised about the interpretability of data-mining models, future work on this topic will seek to provide answers on this issue.

ACKNOWLEDGMENTS

This research was supported in part by the National Science Foundation IGERT program under Grant DGE-0549489. The authors would like to thank the Chicago Metropolitan Agency for Planning (CMAP) for the data used in this study, and Joshua Auld and Sara Niakian at the University of Illinois at Chicago, for their assistance in data pre-processing.

REFERENCES

1. Ben-Akiva, M., and S.R. Lerman. *Discrete Choice Analysis: Theory and Application to Travel Demand.* MIT Press, Cambridge, MA, 1985.

2. Wets, G., Vanhoof, K., Arentze, T.A. and H.J.P. Timmermans. Identifying Decision Structures Underlying Activity Patterns: An Exploration of Data Mining Algorithms. In *Transportation Research Record 1718,* TRB, National Research Council, 2000, pp. 1-9.

3. Arentze, T.A., and H.J.P. Timmermans. A learning-based transportation oriented simulation system. *Transportation Research Part B,* Vol. 38, 2004, pp. 613-633.

4. Hensher, D.A., and T.T. Ton. A comparison of the predictive potential of artificial neural networks and nested logit models for commuter mode choice. *Transportation Research Part E,* Vol. 36, No. 3, 2000, pp. 155-172.

5. Xie, C., J. Lu, and E. Parkany. Work Travel Mode Choice Modeling using Data Mining: Decision Trees and Neural Networks. In *Transportation Research Board: Journal of the Transportation Research Board, No. 1854,* Transportation Research Board of the National Academies, Washington, D.C., 2003, pp. 50-60.

6. Zhang, Y., and Y. Xie. Travel Mode Choice Modeling Using Support Vector Machines. CD-ROM. Presented at 87th Annual Meeting of the Transportation Research Board, Washington, D.C., 2008.

7. Moons, E., Wets, G., and M. Aerts. Nonlinear Models for Determining Mode Choice: Accuracy Is Not Always the Optimal Goal. In *Progress in Artificial Intelligence,* Lecture Notes in Computer Science, Vol. 4874, Springer Berlin/Heidelberg, 2007, pp. 183-194.

8. Cirillo, C., and K.W. Axhausen. Mode choice of complex tours: A panel analysis.

Arbeitsberichte Verkehrs- und Raumplanung, 142. Institut für Verkehrsplanung und Transportsysteme, ETH Zürich, Zürich, 2002.

9. Miller, E.J., Roorda, M.J., and J. Antonio. A tour-based model of travel mode choice. In *Transportation,* Vol. 32, No. 4, 2005.

10. Liu, B. *Web Data Mining: Exploring Hyperlinks, Contents and Usage Data.* Springer-Verlag Berlin Heidelberg, New York, NY, 2007.

11. Friedman, J., Hastie, T., and R. Tibshirani. *Additive Logistic Regression: A Statistical View of Boosting*. The Annals of Statistics, Vol. 28, No. 2, pp. 337-407, 2000.

12. Platt, J., *Machines using Sequential Minimal Optimization*. In: Schoelkopf, B., Burges, C., and A. Smola, editors, Advances in Kernel Methods – Support Vector Learning, 1998.

13. Hastie, T., and R. Tibshirani. *Classification by Pairwise Coupling*. In: Advances in Neural Information Processing Systems, 1998.

14. Freund, Y., and R.E. Schapire. *Experiments with a new boosting algorithm*. In: Thirteenth International Conference on Machine Learning, San Francisco, pp. 148-156, 1996.

15. Witten, I.H., and E. Frank. *Data Mining: Practical machine learning tools and techniques*. 2nd Edition, Morgan Kaufmann, San Francisco, CA, 2005.

16. Transport and Mobility Laboratory. *Biogeme*. Ecole Polytechnique Fédérale de Lausanne, Switzerland. http://biogeme.epfl.ch/. Accessed July, 2008.

17. Yu, C.T., and W. Meng. *Principles of Database Query Processing for Advanced Applications*. Morgan Kaufmann, San Francisco, CA, 1998.

18. Measuring Search Effectiveness. *Recall and Precision*. Creighton University Health Sciences Library and Learning Resources Center, Creighton University, Omaha, NE. http://newadonis.creighton.edu/HSL/searching/Recall-Precision.html. Accessed July, 2008.