Mining Citizen Science Data to Predict Prevalence of Wild Bird Species *

Rich Caruana, Mohamed Elhawary, Art Munson, Mirek Riedewald, Daria Sorokina
Department of Computer Science
Cornell University
{caruana, hawary, mmunson, mirek, daria}@cs.cornell.edu

Daniel Fink, Wesley M. Hochachka, Steve Kelling
Cornell Lab of Ornithology
{df36, wmh6, stk2}@cornell.edu

ABSTRACT

The Cornell Laboratory of Ornithology's mission is to interpret and conserve the earth's biological diversity through research, education, and citizen science focused on birds. Over the years, the Lab has accumulated one of the largest and longest-running collections of environmental data sets in existence. The data sets are not only large, but also have many attributes, contain many missing values, and potentially are very noisy. The ecologists are interested in identifying which features have the strongest effect on the distribution and abundance of bird species as well as describing the forms of these relationships. We show how data mining can be successfully applied, enabling the ecologists to discover unanticipated relationships. We compare a variety of methods for measuring attribute importance with respect to the probability of a bird being observed at a feeder and present initial results for the impact of important attributes on bird prevalence.

Categories and Subject Descriptors: H.2.8 [Database Management]: Database Applications—*data mining, scientific databases*

General Terms: Experimentation

Keywords: Attribute importance, bagging, decision trees, model inspection, partial dependence function, sensitivity analysis

1. INTRODUCTION

Ecology is fundamentally the science of understanding the distribution and abundance of organisms. Ecologists interested in efficient environmental manipulation for conservation and management of wild birds have two general needs: (1) to be able to accurately predict where a species is and is not found; and (2) to understand the causes of presence and absence of a species. Within ecology, the conventional paradigm for analyzing data and gaining insights has been the formulation and testing of a small set of statistical models that are assumed, based on expert opinion, to be the most likely descriptions of the biological processes at work.

This conventional paradigm is now becoming unworkable, overwhelmed by increasingly available large ornithological data sets with many potentially important features (e.g., geographic data sets based on satellite imagery). One example of this is the Avian Knowledge Network (AKN, `http://avianknowledge.net`), a group of university, governmental, and non-governmental ornithological organizations that are combining their existing databases of bird distribution information. Currently, over 25 million bird observation records exist in the AKN's data warehouse, each record associated with data on over 200 environmental features, not even counting the additional geographic data. This volume of data requires new scalable analytical tools that provide ecologists with initial insights (hypotheses) to be subsequently examined in greater detail.

For this, ecologists need to identify features that are strongly associated with interesting patterns of species' occurrence and visualize their effects. The two main challenges addressed in this paper are (1) to develop and evaluate practical strategies for automatically identifying subsets of important features, and (2) to visualize the effects of important features.

Our analytical tools need to deal with several data quality challenges, including missing data, observer-specific biases, biases inherent in each of the multiple data-collection protocols, and potentially high inter-correlation among features. The tools must not only be able to accurately predict birds' distributions, but also identify a potentially limited sub-set of features that are important in predicting distributions of birds. Empirical experience from other ecological studies suggests that only a small number of environmental features will likely have large impact on the distribution and abundance of each bird species.

In this paper, we report on our initial steps in using data mining to explore the AKN data. To eliminate protocol bias, we are limiting ourselves to data collected under a single collection protocol, examining only data on presence/absence of species. We make the following contributions:

- We describe the data set, illustrate the kind of results

*This work was supported by NSF ITR award EF-0427914.

KDD'06, August 20–23, 2006, Philadelphia, Pennsylvania, USA.

in which the ecologists are interested, and point out practical challenges (Section 2).
- We present several very fast heuristics for extracting the most important attributes from a bagged decision tree ensemble by analyzing the structure of the trees. The surprising result is that these different methods produce almost identical rankings of the top-20 attributes and also agree with a far more expensive algorithm for sensitivity analysis (Section 3).
- We present results that the ecologists found useful (Section 4).

Section 5 contains our conclusions and future directions.

2. PROBLEM DESCRIPTION AND CHALLENGES

2.1 Description of the Data

The data examined come from Project FeederWatch (PFW, `http://birds.cornell.edu/pfw`), a winter-long survey of North American birds observed at bird feeders. PFW has been running since the winter of 1987-88, and as of March 2005 had over 1 million submissions, which report a total of 11.7 million bird sightings. Participants report *all* observed species; therefore the data also imply the absence of all species that were not reported. For the 100 most interesting species, this adds about 90 million "bird absence" records to the data set.

Each PFW location and submission is described by multiple attributes, which are provided by project participants. These attributes can be roughly grouped into features related to observer effort, weather during the observation period, and attractiveness of the location and neighborhood area for birds. The observer-provided attributes were supplemented with several hundred additional descriptions of the environment that came from a variety of geographic data sets, e.g., the U.S. Census Bureau's 2000 census (human impact), the USGS National Elevation Dataset, the USGS National Landcover Dataset, and various descriptions of local climatic conditions (e.g., monthly snow depths, wind speed, temperature) from the National Climatic Data Center's Climate Atlas of the United States.

After data were screened to exclude those observation records that were improperly submitted, or did not have sufficient fields to be included in the analysis, a total of about 800,000 observation records with 197 interesting attributes were available for analysis. Recall that each record indicates the presence of a set of species and implies the absence of all other species. Hence for each bird species there are 800,000 records about its presence or absence, constituting a massive amount of information about bird occurrence. These records still included a considerable fraction of fields where the participant-reported attributes were missing.

Because of the ecologists' expectation that species' population trends and the most important influences on any species' distribution will vary across the continent, we subdivided the continent into ecologically-relevant units, using 37 existing Bird Conservation Regions (BCRs; see `http://www.nabci-us.org/map.html`). There are 600+ BCR-species pairs with sufficient data for a data mining analysis. In this paper we limit our attention to the simpler, but nevertheless very challenging, problem of analyzing data from nine BCR-species pairs; specifically we analyze the American Goldfinch, Dark-eyed Junco, and House Finch in BCR's 5, 22, and 30. Due to space constraints, however, we only present results for House Finch in BCR 30 (the U.S. Atlantic coastal plain region from southern-most Maine to northern-most Virginia); the high-level conclusions hold for the other 8 pairs. BCR 30 has 92,514 observations, during which the House Finch is present 55,860 times.

2.2 Desired Results and Challenges

In Section 1, we noted the ecologists' primary goals of predicting and understanding the causes of bird species' distributions and abundances. Motivated by this first ecological goal, one facet of our work has focused on providing precise predictions of changing presence of feeder birds both within a winter, and across years. Motivated by the ecologist's second goal, we have explored multiple techniques for identifying a small number of attributes, from the larger-dimensional attribute set, that together have the most important role in predicting bird's presence or absence.

Data mining tools are well suited for the exploratory analysis of species occurrence in space and time. For example, the `yearseason` graph in Figure 5 illustrates how data mining analysis can produce summaries of changing distribution, in this instance showing the inter-annual trend in presence of House Finches at feeders, after accounting for the effects of all other features. (See analysis details in Section 4.) A steep, disease-induced decline in House Finch occurrence clearly emerges from our analysis.

Conventional ecological analyses of species occurrence based on statistical techniques (e.g., Generalized Linear Models [13]) can detect changes in occurrence and identify important subsets of variables. However, these techniques require expert training and tuning, especially when there are complications like missing data, the need to fit non-linear effects, and potentially many interactions. These challenges quickly overwhelm standard techniques when faced with large sample sizes and large numbers of attributes.

Data mining tools offer better flexibility and near automatic application compared to standard statistical techniques. Many data mining methods scale to large sample sizes and large numbers of attributes. Decision trees are particularly well suited to dealing with missing data. Yet, there has been little emphasis to date on extracting information about variable importance from these predictive models.

Approaches that rely on comparisons of the predicted probability surface tend to be conceptually simple but computationally expensive. The essential problem is that comparing predicted probabilities as a function of any single focal attribute, such as in Figure 5, requires enough predictions at each value of the focal attribute to effectively marginalize over the effects of all other attributes (see also discussion in Section 4). The number of predictions necessary for this marginalization step increases dramatically with the number of attributes, quickly becoming infeasible. The problem is further compounded when interactions are considered for pairs and tuples of attributes.

We therefore need a fast heuristic approach to identifying sets of important attributes that enables ecologists to later refine and investigate the effect of important attributes. Our approach focuses on mining the structure inherent in the predictive models. Then interaction plots are produced to visualize the effects of important attributes. We describe our approaches and compare them in the following section.

3. BUILDING AND ANALYZING MODELS

Our general approach is to first build highly accurate non-parametric models that capture the relationships between attributes. Then we determine important attributes by analyzing the model. For the analysis we selected bagged decision trees as our model of choice [2, 1], because trees can capture non-linear relationships and they handle missing values gracefully. In recent work it has been shown that bagged trees are competitive with the best available learning methods [5, 14]. Decision trees provide the added benefit of being intelligible. The tree structure reveals valuable information about attribute importance based on which attributes the tree selects for splits.

We have examined and compared several approaches to measuring variable importance. One of these methods is a "black box" method; i.e., it examines only the inputs and outputs of the model and does not depend on the structure of the model itself and therefore can be applied to any classifier. The other methods are "white box" techniques that analyze the structure of individual trees in an ensemble.

In the following sections we discuss these methods and compare their results. For all experiments the tree ensemble consists of 100 ID3 trees[1], built using the IND package [4]. The data sets are partitioned into roughly 2/3 training and 1/3 testing. Building the entire ensemble on a single processor modern PC (3.6 GHz, 1 GB RAM) takes about 2 hours.

3.1 Black Box Approach: Sensitivity Analysis

We use Breiman's sensitivity analysis technique [3] to assess the importance of a feature while treating the learned model as a black box (this method is related to randomization and permutation tests used in statistics [12]). The idea is to compare the performance of the model on a test set before and after noise is added to the target feature.

To measure the importance of feature A, all A-values are shuffled, essentially permuting the original vector of A-values (when viewing the data set as a matrix whose rows are the different observation records and columns correspond to the different features). If the attribute is important, performance should drop on the perturbed test data set compared to the real one, because the model relies on the spoiled values when making predictions.

There are many different measures of model performance. For our sensitivity analysis, we selected a diverse set of commonly used measures to avoid measure-related bias. In particular, we based the importance rankings on three different metrics: accuracy (ACC), root mean squared error (RMS), and ROC area [15] (ROC). The performance was measured on a separate test set, i.e., none of the test records was used for training the bagged trees.

Table 1 shows the feature sensitivity results for the House Finch in BCR 30, sorted by RMS. An entry in the table reports the relative loss in performance between the real and perturbed test data set. For example, let x be the model's accuracy on the real test data, and y be its accuracy after permuting the latitude values. The relative loss for latitude is $(x-y)/x$. Relative loss for RMS is computed as $(y-x)/x$ since lower RMS scores indicate better performance.

Sensitivity analysis is a relatively fast method for estimating variable importance. Once the model is trained, we only need to evaluate its performance for different perturbed test data sets, one for each attribute. This is much faster than the costly approach of re-training models for different sets of attributes, as required for feature selection methods [10, 11, 8]. Nevertheless, for large high-dimensional data sets like PFW, even sensitivity analysis requires considerable resources: evaluating the sensitivity of a *single* feature using the 32K test cases for BCR 30 takes about 4-5 minutes. Using this approach for all 197 features of interest (or even pairs or larger sets of features) and for all 600+ BCR-species combinations requires access to expensive high-performance computing resources. In the following section we propose efficient heuristics to address this issue.

attribute	ACC	RMS	ROC
latitude	0.079	0.15	0.070
longitude	0.0056	0.045	0.014
numfeeders_hanging	0.012	0.034	0.013
halfdays	0.013	0.034	0.015
yearseason	0.012	0.032	0.014
dayselapsed	0.016	0.030	0.014
numfeeders_thistle	0.0098	0.022	0.0095
ave_fam_sz	0.0016	0.011	0.0040
effort_hrs_atleast	0.0030	0.010	0.0045
asian	0.0014	0.0091	0.0030
elev_ned	0.00023	0.0067	0.0024
evgr_trees_atleast	0.00072	0.0050	0.0017
numfeeders_suet	0.00045	0.0048	0.0016
gcsnow2912	8.5E-05	0.0045	0.0015
pop00_sqmi	8.5E-05	0.0041	0.0012
vacant	0.00048	0.0037	0.0011
count_area_size	0.00016	0.0037	0.0012
other	-0.00063	0.0035	0.0011
elev_gt30	0.00012	0.0032	0.0010
ave_hh_sz	-0.00093	0.0030	9.8E-4

Table 1: Top-20 attributes for sensitivity analysis, sorted by RMS

3.2 White Box Approach: Looking At Trees

The methods discussed in this section leverage the fact that we are using ensembles of decision trees. We can inspect the learned trees to see which attributes have been selected. Because selected attributes separate positive and negative observations, they are clearly important predictors. If an attribute is "important" for many of the trees in the ensemble, then we have strong evidence of its overall importance. The main challenge is defining a good measure to quantify an attribute's importance in a tree and in an ensemble of bagged trees.

We have implemented several ranking methods that use only the information about the tree structure and how a training set is partitioned by the different trees. This information is available once the ensemble is built, so there is no need to generate new models or new predictions in order to calculate these rankings. This is a clear advantage over black box methods like sensitivity analysis or feature selection. We can compute the complete ranking of *all* features in less than 2 minutes (no matter which of the methods introduced below we are using), compared to 4-5 minutes *per feature* for sensitivity analysis—a factor of 500 speedup!

The importance score of an attribute for the tree ensemble is computed by summing the importance scores on the individual trees. To illustrate the differences between the methods, we will use the simple tree shown in Figure 1. It splits on three attributes: A, B, and C. The training set has 100 records; numbers in parentheses indicate the number of

[1]Experiments with other tree types indicated that the choice of tree had fairly little effect on the ensemble performance.

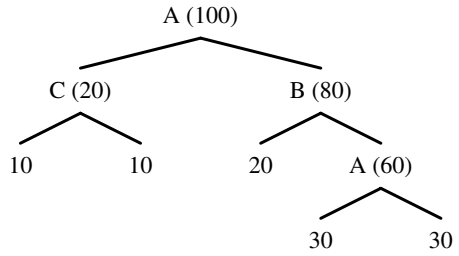

Figure 1: Sample decision tree

records affected by the corresponding split (i.e., the number of records in the corresponding subtree). We consider the following methods for computing attribute importance scores on a single tree.

Number of nodes (#nodes). An attribute's score is the number of nodes in the tree that selected the attribute for the split. In our example attribute A gets importance score 2, while B and C receive importance scores of 1 each. This method will give too much weight to continuous attributes, because the tree can split on them more often. The other methods address this issue.

Weighting by height (height). Greedy tree growing algorithms usually choose the most important attributes early, so they appear higher in the tree structure. This method weights each node inversely proportionally to the length of the path from it to the root. The root itself is considered to have importance 1, so in the example attribute A receives importance $1 + 1/3$ (importance of root + importance of the rightmost subtree split), while attributes B and C each have importance $1/2$. The example in Figure 1 illustrates a problem with height-based weighting. Attributes B and C receive the same weight, whereas splitting on them affects different numbers of cases in the data set. To correct for this, the following methods take into consideration the number of training cases affected by the split.

Weighting by size of training set — multiple counting (multiple). This method weights a node by the number of training cases in its subtree, i.e., the cases affected by the split at this node. In the example, attributes A, B, and C receive scores of 160, 80 and 20, respectively.

Weighting by size of training set — single counting (single). As with *#node* ranking, there is a risk that continuous attributes will be over-weighted when using the *multiple* counting of training points. In the example, the 60 records in the lower-right subtree with parent node A are counted twice towards A's score. To fix this problem, *single* counting assigns weight zero to all nodes that have an ancestor with the same split attribute. In the example A receives an importance score of 100 instead of 160, while the scores for B and C do not change.

Weighting by size of training set — giving weight to the path (path). This method compromises between *single* and *multiple* counting. Intuitively, training records from every leaf are distributed evenly between the splits on the path from the root to the leaf. Each split is still counted, even if there is another split on the same attribute in an ancestor node. In our example, the 30 records from the rightmost leaf are distributed between the two splits on A and the one split on B, i.e., 20 points go to A and 10 to B. Similarly, the 10 points from the leftmost leaf are given to A and C, in this case 5 points to each. Counting from left to right, A receives an importance score of $5 + 5 + 10 + 20 + 20 = 60$, B gets $0 + 0 + 10 + 10 + 10 = 30$ and C gets $5 + 5 + 0 + 0 + 0 = 10$. It is worth mentioning that importance scores for all attributes sum to the size of the training set used to build the tree. A similar method was used by Friedman [7] for estimating attribute importance in an ensemble of rules.

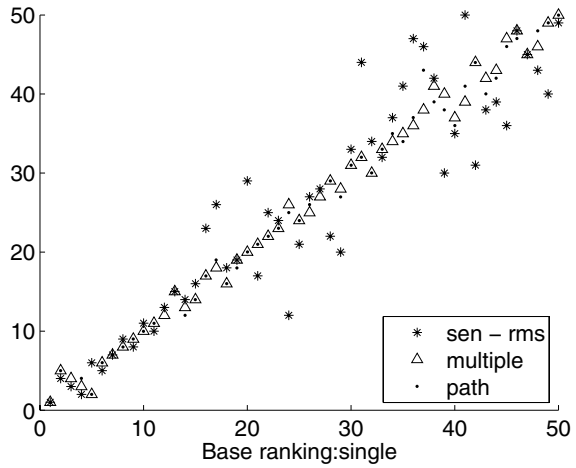

Figure 2: Comparison of different rankings (first 50 features shown). X-axis represents attributes in the order induced by ranking *single*, y-axis measures their position in other rankings.

3.3 Comparison of Rankings

All three measures based on the size of training set in splitting nodes are very similar (Fig. 2). This result is surprising, because different ways of handling continuous attributes could in theory have significant influence on the resulting rankings. In practice we observed only minor differences.

#nodes and *height* produced rankings that are very similar to each other, but differ from the previous group. Fig. 3 shows that *height* and *#nodes* almost always agree, but are very different from the diagonal where they would be if they were correlated with *single*. Subsequent tests showed that results of these methods are less reliable than those of *single*, *multiple* and *path* (see Section 3.4).

One of the sensitivity analysis rankings — *sensitivity-rms* — shows a lot of similarity with the three most reliable methods from the "white-box" group (Fig. 2). *sensitivity-acc* tends to agree with them only for the top ranked features and then shows a significant amount of discrepancy (Fig. 3). Because accuracy is known to be a high variance measure, while RMS is very stable, we have more confidence in the results of *sensitivity-rms*. (*sensitivity-roc* produced results similar to *sensitivity-acc* and therefore is omitted from the plot.)

The most important result here is the fact that our very fast white-box methods essentially identify the same top 20 features as the much more expensive black-box method. This result is also true for the other 8 BCR-species pairs we analyzed. It means that we can take advantage of the faster methods without sacrificing result quality.

3.4 Sanity Check

There is no guarantee that taking the top-ranked features from any of these importance measures will yield an ensemble with good predictive power. While prediction accuracy

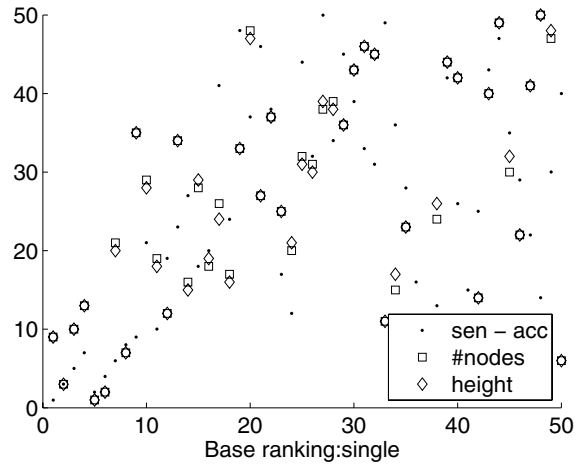

Figure 3: Rankings that do not agree well with *single*. The farther from the diagonal each point is, the larger is the disagreement.

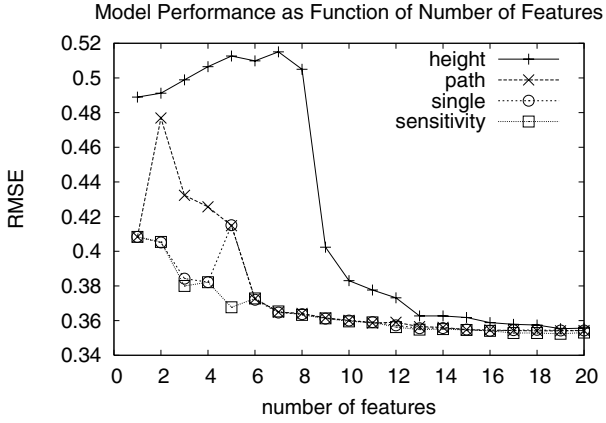

Figure 4: Performance as a function of the number of features used for training. Each line represents a different method for ordering features by importance—yielding slightly different sets of features.

is not the only goal of this study, it is a necessary precondition. Clearly we cannot hope to learn something about this domain by studying inaccurate models. Also, ecologists are interested in comparing the important attributes of a species occurrence in different BCRs. This can be achieved by comparing rankings, but only after checking that some minimum predictive performance is met in all analyses.

As a sanity check, we compared the performance of bagged trees trained using all features with bagged trees trained using only the top 20 features from the different importance rankings. With all features, the bagged trees achieve a RMS of 0.3469, accuracy of 0.8336, and area under the ROC curve of 0.9012.

Figure 4 plots the ensemble's RMS performance when only the top N features from each ranking are used, for different values of N. Because the rankings differ from each other, different features are included at each point for the different lines (Table 2 and the RMS column from Table 1). The overall pattern is similar for accuracy and ROC area, so we omit those graphs.

We make several observations from Figure 4. First, the

	height	path	single
1	dayselapsed	latitude	latitude
2	yearseason	dayselapsed	halfdays
3	halfdays	nf_hanging	nf_hanging
4	temp_lo_atleast	longitude	longitude
5	temp_hi_atleast	halfdays	dayselapsed
6	precip_len_atleast	yearseason	yearseason
7	effort_hrs_atleast	nf_thistle	nf_thistle
8	snow_dep_atleast	effort_hrs_atleast	effort_hrs_atleast
9	latitude	ave_fam_sz	ave_fam_sz
10	nf_hanging	elev_ned	elev_ned
11	nf_ground	asian	asian
12	nf_suet	pop00_sqmi	nf_suet
13	longitude	nf_suet	count_area_size
14	snow_cov_atleast	vacant	pop00_sqmi
15	nf_platfrm	count_area_size	vacant
16	pop00_sqmi	elev_gt30	black
17	elev_gt30	black	age_65_up
18	nf_water	ave_hh_sz	elev_gt30
19	asian	age_65_up	ave_hh_sz
20	black	houden	houden

Table 2: Top-20 attribute rankings; 'numfeeders_' is abbreviated as 'nf_'.

ensembles built using only 20 features perform quite well, although not quite as well as ensembles using all the features. The top 20 features do seem to catch most of the predictive power found in the full feature set. This gives us some confidence in relying on these measures as indicators of which features are important for modeling the PFW domain.

Second, while the rankings from single counting, path counting, and sensitivity-rms analysis show similar behavior, the height-based ranking behaves very differently. This agrees with the finding above that the height importance measure is not as highly correlated with the other measures.

One surprising aspect of this graph is that all the lines go up at least once: *path* at feature 2, *single* at feature 5, *sensitivity* at feature 6, and *height* for the first half of the graph. This phenomenon is partly caused by the feature `dayselapsed`; whenever it is added, performance gets worse in this graph. Given that all the measures rank this feature highly, and the ecologists believe it to be an important predictor, this is rather surprising.

Overall, we have identified methods for analyzing the ensemble model that produce very similar rankings of attribute importance. We have also shown that the resulting rankings are reasonable: models generated using only the top 20 features show good performance.

3.5 Discussion

While the importance heuristics can be used to choose reasonable small sets of features, the heuristics will not find many other sets of important features that perform as well or even better. We have performed additional experiments where we trained the model using the top 16 features from Table 2's *single* column. Retraining the model without access to the latitude and longitude features (i.e., using only 14 features) results in equivalent performance. In our environmental data, attributes tend to be correlated and thus contain varying degrees of redundant information.

In this sense, the importance heuristics necessarily are inferior to feature selection at determining important features. The heuristics only find features that are important relative to the learned model, and not necessarily features that are important to many or all models.

4. SELECTED RESULTS

As described in Section 1, one of the main goals for this project is identifying features that are important in predicting the abundance of bird species. Section 3 presented several heuristic methods for finding potentially important features. In order to decide if a certain feature requires closer examination, ecologists need information about how the feature affects the probability of observing the bird.

To provide such information we estimate and plot the probability of spotting the bird given different values of the feature in question. Figure 5 contains several examples of this kind of graph; for convenience we will refer to these graphs as *trend plots*. The rest of this section describes how we generate trend plots and discusses some sample plots.

4.1 Generating Trend Plots

We explore two methods for plotting trends: 1) computing conditional probabilities directly from the data; and 2) computing Friedman's *partial dependence function* [6] for the feature of interest, using the previously learned model to estimate probabilities. We will refer to these methods as *data*, and *partial*, respectively.

Data: Given each value of the feature of interest, we compute the probability of seeing a bird. This is just the mean of *all* the points in our data set that have the given value of that feature. Points lacking a value for the feature (i.e., missing value) are not used. Continuous features are discretized into 5% quantiles to yield twenty distinct values for plotting, with each data point summarizing roughly the same number of data records. The top of each bin (quantile) is plotted on the x-axis. Note that continuous features are discretized identically for both methods.

Partial: For each value v to plot for feature X, create an artificial data set D_v by setting $X = v$ for *all* the points in the test set.[2] Each artificial data set is labeled by the (previously learned) bagged tree model. The probability of observation when $X = v$ is computed by averaging the predictions for the set D_v. Missing values are a non-issue with this method.

The motivation behind partial dependence functions is that the target feature X may have high correlation with another feature Y for *some* values of X. If X is not an important influence but Y is, marginalizing to find X's influence on seeing a bird (using the data method above) can make X look like an important indicator for values where it correlates well with Y (the truly important attribute). As a result, perceived observation trends as a function of X may be exaggerated or may not exist at all.

Substituting $X = v$ for all points breaks up potential covariances and forces the model to focus more on the impact of X having value v. The only thing that changes between plot points is the value of X — holding all other features constant in some sense, while still maintaining the natural distribution of their values.

In theory, partial dependence functions can produce misleading plots in cases where we generate many new points in regions of the feature space unsupported by our data. The model, which was not trained on the data from those regions, can produce unpredictable results that will harm our trend plots. The detailed description of a similar problem can be found in [9]. In our analysis, we have not discovered this problem yet.

[2]The mean value of each quantile is used as the substitution value for continuous features.

4.2 Sample Trend Plots

Because computing partial dependence functions for all 197 features is too computationally expensive, we examined the top 20 features from the single counting ranking (see Section 3.2).[3] Figure 5 shows six trend plots. Brief descriptions of the features plotted are given below.

Each graph shows the probability of observing the House Finch in BCR 30 as a function of a given feature. The *data* and *partial* lines are marked with `x`'s and `o`'s, respectively. As a general rule, the *partial* plots are much smoother than the *data* plots, which exhibit much more local variance. In most cases, however, both methods show the same general trends. Most of the comments below will focus on the *partial* plots, because they are easier to read and interpret.

`yearseason`: The observed decline in occurrence is consistent with ecologists' background knowledge that a novel bacterial pathogen, first appearing in 1994, has caused declines in abundance of House Finches across Northeastern North America.

`latitude/longitude`: As we know from Section 3 these two features both have high importance ranking. Other experiments (omitted due to space constraints) also showed that they are highly associated both with each other and with many other attributes. We believe that these attributes describe spatial gradients and possibly act as proxies for other attributes that also exhibit spatial variation. The greater range of variation in the latitude effect may be due to the large North-South orientation of BCR 30.

`numfeeders_hanging`: This feature counts the number of hanging bird feeders in the observation area. As the number of feeders increases from 0 to 5, we see an increasing probability of observation. The plateau effect past 6 feeders suggests that once there are sufficient feeders, adding more does not increase the chances of seeing a bird.

`dayselapsed`: This variable counts the number of days elapsed since the beginning of the PFW season. Since the season begins on November 1, day 31 is the beginning of December (for example). The observed pattern of probability of occurrence is consistent with the known partial-winter migratory behavior of House Finch populations in the Eastern United States, where a proportion of the winter population migrates.

`pop00_sqmi`: This is the *human* population per square mile, as measured during the 2000 census. This is a good example of the *partial* plot differing from the *data* plot. The former suggests that the influence of population density on House Finch occurrence is relatively small, despite the fact that the model considers it important. The latter, however, would indicate that the probability of seeing a House Finch increases dramatically as population density goes up. Taken together, it seems more likely that population density correlates with other important indicators (especially given the large peaks and valleys in the data line).

The example of `pop00_sqmi` also shows that an attribute can be important for model prediction even though its *partial* line is close to a flat line. A flat trend line does not prove that a feature is unimportant. Rather, it just shows

[3]Given the high correlation between the ranking methods for the top 20 features, the choice of ranking method is basically arbitrary.

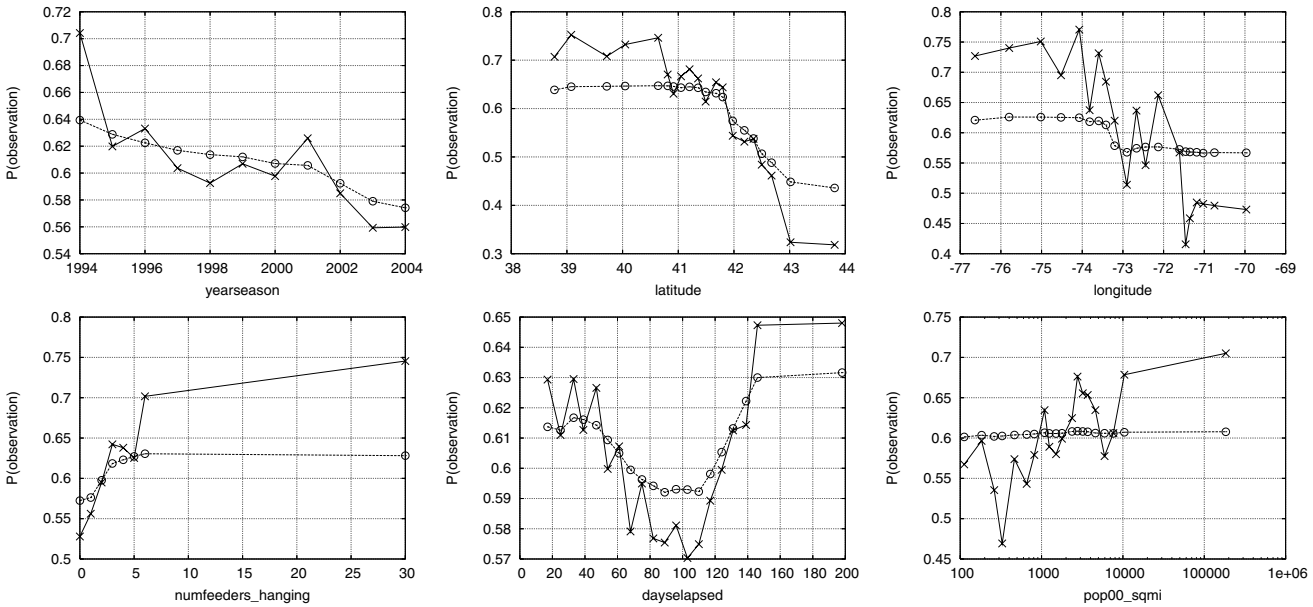

Figure 5: House Finch observation trends in BCR 30. Each graph shows the *data* (x line) and *partial* (o line) plots for a different feature. Note that the y-axes cover different ranges.

that in this marginalized setting the feature does not carry much predictive weight. Combined with other features, however, it may be very important for making good predictions. Therefore, examining trend plots is not a viable way to identify important feature sets by itself.

5. CONCLUSIONS AND FUTURE WORK

Finding important features for predicting the presence or absence of species is one of the major goals of ecology. The large data size, the large number of features, and the inherent quality issues of data collected by citizen science projects make this a truly challenging problem. In this paper we analyzed techniques where the importance of a feature is determined by how heavily an accurate data mining model relies on the feature for its predictions. More expensive approaches like feature selection did not scale, resulting in poor response times even for this limited study.

We presented very fast heuristics for measuring attribute importance that are based on analyzing the structure of decision trees. An interesting outcome of this study is that all heuristics that measure importance by the number of training cases affected by a node split produce almost identical feature rankings. Furthermore, the top 20 of these rankings are also highly correlated with those computed by much more expensive sensitivity analysis.

Once a small set of interesting features is identified, expensive trend plots can be generated to gain a better understanding of how certain features affect the observation probability for a species.

The analysis presented in this paper was applied to 9 BCR-species pairs, for a single project (PFW). As pointed out earlier, ecologists ultimately want to compare and contrast such results for all of the roughly 600+ pairs containing sufficient data. This, together with the rapid growth of data poses further demands for even faster techniques. For example, sensitivity analysis will not be a practical option at this scale.

Major directions of our future work include further analysis of resulting models, important features and relationships between them. We plan to extend our analysis from identifying single important features to detecting groups of interacting attributes.

6. REFERENCES

[1] E. Bauer and R. Kohavi. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine Learning*, 36(1–2), 1999.

[2] L. Breiman. Bagging predictors. *Machine Learning*, 24:123–140, 1996.

[3] L. Breiman. Random forests. Technical Report 567, University of California Berkeley, Statistics Department, 2001.

[4] W. Buntine. *Artificial Intelligence Frontiers in Statistics*, chapter Learning Classification Trees. Chapman and Hall, 1993.

[5] R. Caruana, A. Niculescu, B. Rao, and C. Simms. Evaluating the C-section rate of different physician practices: Using machine learning to model standard practice. In *The American Medical Informatics Conference (AMIA)*, 2003.

[6] J. Friedman. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29(5):1189–1232, 2001.

[7] J. Friedman and B. Popescu. Predictive learning via rule ensembles. Technical report, Stanford University, 2005.

[8] I. Guyon and A.Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

[9] G. Hooker. Generalized functional ANOVA diagnostics for high dimensional functions of dependent variables. Available at http://ego.psych.mcgill.ca/perpg/pstdc/giles, 2005.

[10] K. Kira and L. Rendell. The feature selection problem: Traditional methods and a new algorithm. In *Proc. Int. Conf. on Artificial Intelligence (AAAI)*, 1992.

[11] R. Kohavi and G. John. The wrapper approach. *Artificial Intelligence*, 97(1–2), 1997.

[12] E. L. Lehmann. *Nonparametrics: Statistical Methods Based on Ranks*. Chapman and Hall/CRC, 1989.

[13] P. McCullagh and J. A. Nelder. *Generalized Linear Models*. Mcgraw-Hill, 1989.

[14] A. Niculescu-Mizil and R. Caruana. Predicting good probabilities with supervised learning. In *Proc. Int. Conf. on Machine Learning (ICML)*, 2005.

[15] F. Provost and T. Fawcett. Robust classification for imprecise environments. *Machine Learning*, 42(3):203–231, 2001.