Cost-Sensitive Feature Acquisition with Imbalanced Training Data

David R. Thompson,
Kiri L. Wagstaff, Walid A. Majid, Dayton L. Jones
Jet Propulsion Laboratory,
California Inst. of Technology
4800 Oak Grove Dr.
Pasadena, CA 91109 USA
{firstname.lastname}@jpl.nasa.gov

Sarah Burke-Spolaor
Swinburne University
Center for Astrophysics and Supercomputing
Melbourne, Australia

ABSTRACT

Many online classification tasks involve measuring hidden features through a sequence of costly tests prior to class assignment. Cost-sensitive decision trees (CSDTs) address this problem by applying the most effective diagnostic test at each branch node. We apply CSDTs to the problem of active searches for very rare or unique events. These tasks are characterized by training sets that are often not representative of the target features in question - in other words, they exhibit both within-class and cross-class imbalance. We present a new tree-learning algorithm to address this problem. A factorized model of data attributes permits a nonparametric representation reflecting the anticipated features of target events. We evaluate performance for the task of real-time analysis to identify interesting signals in radio astronomy data.

Categories and Subject Descriptors

I.5.0 [**Computing Methodologies**]: Pattern Recognition—*General*
; I.2.6 [**Computing Methodologies**]: Artificial Intelligence—*Learning*
; G.3 [**Mathematics of Computing**]: Probability and Statistics—*Distribution Functions*
; J.2 [**Computer Applications**]: Physical Sciences and Engineering—*Astronomy, Physics*

General Terms

Real Time Data Mining, Decision Trees, Radio Transients, Radio Astronomy

1. INTRODUCTION

This work treats the problem of cost-sensitive feature acquisition for classification with misrepresentative training data.

Cost-sensitive feature acquisition attempts to classify a candidate datapoint from incomplete information. In this task an agent acquires features of the datapoint using one or more costly diagnostic tests, and eventually ascribes a classification label. A cost function describes both the penalties for feature acquisition as well as misclassification errors. This problem is common in such areas as industrial production, troubleshooting in production processes or communications networks, and medical diagnosis.

A common solution is a Cost Sensitive Decision Tree (CSDT), a branching sequence of tests with features acquired at interior decision points and class assignment at the leaves. CSDTs can incorporate a wide range of diagnostic tests and reflect arbitrary cost structure. They are particularly useful for online applications due to their low computational overhead. A variety of tree learning algorithms exist; most, such as C4.5 [17], estimate classification risk by propagating a representative training set through the partially-constructed tree.

Here we apply CSDTs to cost sensitive feature acquisition where the goal is to recognize very rare or unique phenomena in real time. Example applications from this domain include:

- Stream processing, where one seeks unique events in a real time data stream that is too large to store. Here the primary cost is computation; processing must be allocated adaptively to analyze the most promising candidate events.
- Fault protection, where a system must adapt quickly to react to anticipated errors by triggering repair activities or followup diagnostics.
- Real time sensor networks, where one seeks to classify unique new events as they occur. Feature acquisition takes the form of communication with individual nodes, with costs representing the cost of communication or the time required for additional data collection activities.
- Observational sciences, where a new generation of instrumentation seeks unique events through on-line analysis of large observational datasets. Feature extraction costs could represent computational cost for on-

line processing or the real cost of followup observations on candidate events.

In each of these cases the target events in question may have been seen only rarely, or may only be anticipated based on physical models. A training distribution is likely to be misrepresentative, exhibiting imbalances both *between* the classes of interest and *within* the attributes of the target class. This precludes traditional approaches to handling biased data sets, such as simply resampling to correct for the desired class distribution.

This work presents an alternative solution that permits principled CSDT learning while exploiting any prior knowledge of the designer to correct *both* between-class and within-class imbalance. We adaptively reweight training examples using a graphical decomposition of the data attributes. The result is a new nonparametric representation that matches the anticipated attribute distribution for the target events. This facilitates a simple tree-learning that minimize expected total misclassification and feature acquisition cost. It exploits both expert knowledge and training data, using each to model the parts of the distribution for which it is best suited.

The paper begins by surveying previous work in cost-sensitive feature acquisition. We also survey previous work on correcting imbalanced training data. Then section 3 details our decision tree learning approach. Finally we present a case study involving cost-sensitive feature acquisition for real time data mining in petabyte radio astronomy data streams. Detection of anomalous radio transient sources is tantamount to a classification problem with feature acquisition costs reflecting constraints on time and computational resources. Our experiments compare traditional and parametric decision tree learning strategies, as well as "brute force" nonadaptive detection strategies in this domain. Cost-sensitive decision trees provide significant performance improvements over *status quo* feature extraction methods.

2. PREVIOUS WORK IN COST-SENSITIVE FEATURE ACQUISITION

Decision tree classification has a long history, but most studies deal exclusively with either feature acquisition or misclassification cost. Turney offers one of the first joint treatments [19]. Turney introduces the ICET fitness function, which is the average cost of classification including costs of feature acquisition and misclassification. This combined cost function was later formalized by Greiner et al. who perform a PAC analysis of active feature acquisition [9].

More recently, Ling et al. design cost-sensitive decision trees that minimize misclassification and feature acquisition costs during both training and testing [15]. They propagate training examples through the tree, and choose tests that minimize the total cost on the training set. This replaces the usual entropy criterion used by C4.5 [17]. Yang et al. extend this approach with naive Bayes variants to decide on which features to test. Notable variations include csNB, which uses a knapsack algorithm to select a set of "most valuable features" using dynamic programming based on a total cost budget.

Also notable is Ji and Carin's alternative formulation treating cost-sensitive feature acquisition and classification as a Partially-Observable Markov Decision Process [12]. This abandons the tree representation and instead views feature acquisition as a temporal sequence of actions. Each action results in some non-deterministic state transition, an observation, and a reward. The observations provide information about the true class (i.e. the hidden state). This approach is highly general, and accounts for complications like diagnostic tests that modify the underlying class. Unfortunately the true POMDP is intractable for most real-world problems. Ji and Carlin demonstrate significant benefits using a myopic approximation.

Sheng and Ling describe the Sequential Batch Test strategy for feature value acquisition at test time [18]. This assumes an existing cost-sensitive model has been learned from complete training data (they use their cost-sensitive decision tree). The main advance is the inclusion of delay time costs, which are incorporated by translating them into dollars using an assumed hourly rate. They focus on the issue of overlapping delay times, where one may want to test several features at a time, and use an A* search to find a batch of features to query with maximum expected cost reduction. Thus, their method incorporates both acquisition, misclassification, and delay costs.

These previous approaches all propagate instances from the training set through the decision tree. One can then compute the combined feature extraction and misclassification cost by counting the tests performed and the errors at each leaf. However, this approach presents several problems for the diagnosis of rare events. First, the target events may outnumber uninteresting ones by many orders of magnitude. This is the problem of between-class imbalance, which is generally remedied by resampling [14]. Unfortunately resampling is infeasible for finding very rare or unique events. For the case study we consider in this work, matching the anticipated distribution using just a single "positive" training example could require a training set as large as the entire observational history of the agent.

An alternative to correcting between-class imbalance alters the cost function [14]. One could extend this strategy for cost sensitive feature acquisition by scaling both classification and feature acquisition terms. However, this could not address an important related problem: *within-class* imbalance. The rare events in question may never have been seen before, or they may only be anticipated based on physical models. For example, a sensor network designed to detect earthquakes may wish to detect new events in an area where none have been observed previously. If the training data's attributes are themselves nonrepresentative, than neither resampling nor alterations to the cost function will produce an accurate answer.

In principle one can leverage training examples, along with domain knowledge about predicted features, to compute an optimal feature acquisition policy. Japkowicz [11] addresses the within-class imbalance problem in the case where the classes can be represented by a mixture of discrete clusters. This is tantamount to subdividing each class into several "subclasses," which can then be resampled independently

until the desired distributions match. Unfortunately, this method relies on resampling so it does not translate well to our domain of unique events. Moreover, it assumes that the imbalance can be described in terms of varying coefficients to a mixture model. In the more general case, we may wish to ascribe an arbitrary new distribution over some subset of continuous-valued features. To our knowledge no general solution to the within-class imbalance exists that would address these problems.

One other possibility might be to treat the tree structure itself as a model parameter and learn it through a combination of data and Bayesian priors. This could incorporate domain knowledge in the form of prior trees to seed a more informed structure search. This is the technique employed by Andronescu and Brodie [1], who ascribe probability distributions to each node in order to construct populations of decision trees. They use prior trees to improve performance for traditional classification tasks, with the likelihood of a given data/tree combination defined by both the data likelihood and the tree probability determined by the prior. Other notable examples of Bayesian tree-learning strategies include work by Denison et al. [5] and Chipman et al. [3] who use Markov Chain Monte Carlo to sample the distribution of tree structures in a nonparametric regression task.

The next section explores an alternative method which could complement or supplant Bayesian tree learning. We factorize the data's features into "properties" that can be modeled and "attributes" that result from diagnostic tests, and compute weighting factors for the training set in order to match the target events' distribution over classes and attributes. This nonparametric representation of target events can train a CSDT using one of the existing learning algorithms. We find that the correction can significantly alter the choice of diagnostic features and improve the overall efficiency and accuracy of the resulting CSDT. It preserves the properties of the learning tree algorithm for rare events without growing the size of the underlying data set.

3. CORRECTING IMBALANCED TRAINING DATA

Our approach relaxes many of the standard assumptions used in CSDT literature, such as the form of the test diagnostics (generally thresholds on a single feature) and the data representation (generally an unweighted training set). We revisit the basic CSDT problem here to establish our assumptions and notation conventions.

3.1 Method

Consider that the independent data points each have a single discrete class label $c \in \mathcal{C}$. We also define permanently hidden features $h \in \mathcal{H}$ corresponding to intrinsic properties of the event. These properties could include one or more discrete or continuous values, and influence the result of any diagnostic tests. Diagnostic tests applied to the data point reveal observed features $f \in \mathcal{F}$. For simplicity, we will only consider the case where the feature acquisition process is totally deterministic.

Each datum presented to the decision tree follows a path through branch nodes n. A branch applies a processing operation to acquire one or more features and incurs a *feature acquisition cost* $R_{\text{acq}}(n)$. It then performs a test to determine to which of its child subtrees the datum will propagate. This test could be a complicated classifier in itself, or as simple as a comparison of one or more features against a threshold. Note that the test has access to all the features extracted both by the node itself and also by its parent nodes.

On reaching a leaf node $\ell \in \mathcal{L}$ the datum receives a class label. The detector incurs a misclassification cost described by the matrix $R_{\text{class}}(c, \ell)$. We define $R_{\text{tot}}(\ell, c)$ to be the total cost of a particular path which is the sum of misclassification costs at the leaf and the combined feature acquisition cost of the leaf node's ancestors $Anc(\ell)$.

$$R_{tot}(\ell, c) = R_{\text{class}}(\ell, c) + \sum_{n \in Anc(\ell)} R_{\text{acq}}(n) \quad (1)$$

$P(\ell|c, h)$ represents the probability of a datum of a given class c and hidden properties h arriving at node ℓ. The total risk of the tree configuration is the expected cost over all possible classes and properties.

In this work we use a nonparametric representation of $P(f, h, p)$. Specifically we use a set of weighted point masses $i \in M$ such that $P(f_i, h_i, p_i) = \alpha_i, \quad \sum_i \alpha_i = 1$. The destination leaf of each point mass is ℓ_i which is a deterministic result of observable features. We can compute the expected utility of a given tree by propagating the point masses through the sequence of tests. For the set M_ℓ of all point masses arriving at node ℓ:

$$\begin{aligned} U &= \sum_{\ell \in \mathcal{L}} P(\ell) R_{\text{tot}}(\ell, c) \\ &= \sum_{\ell \in \mathcal{L}} \sum_{i \in M_\ell} \alpha_i \left[R_{\text{class}}(\ell, c) + \sum_{n \in Anc(\ell)} R_{\text{acq}}(n) \right] \end{aligned} \quad (2)$$

Figure 1 shows a simple tree learning rule, based on modifications to the algorithm by Ling et al [15].This myopic strategy begins with a single branch at the root node. It chooses an optimal diagnostic test assuming its children are leaf nodes with classification labels that best reduce the total cost. The total utility of a diagnostic test is based on the expected cost per datapoint (Equation 1). We maintain a list $\mathcal{Q}$ of unexpanded leaf nodes, and continue adding descendants in similar fashion. If no feature extraction would decrease the total expected cost, we keep that node as a leaf and remove it from the list of unexpanded nodes. Note that in the case where each point mass is weighted equally, the algorithm is equivalent to the Ling algorithm which propagates elements of the training set.

The set of point masses derived from training data represents $P(f, h, c)$ for the training environment. However, in many cases we would like to use this same training data approximate some new distribution $P'(f, h, c)$. Perhaps domain knowledge supplies more specific prior information about target class and property distributions. For example, a seismic network might have some advance knowledge about the location and likelihood of new earthquakes, but have never observed any in this region previously. This presents

$\mathcal{Q} = \{root\}$
while $|\mathcal{Q}| > 0$
$\ell \leftarrow$ next unexpanded leaf from $\mathcal{Q}$
compute the M_ℓ, the subset of point masses arriving at ℓ
Best utility $U^\star$ initialized to the cost of not branching
  for each i in M_ℓ
    $U^\star \leftarrow \alpha_i R_{class}(\ell, c_i)$
for each potential diagnostic test t
  add child leaf nodes $\mathcal{N} = \{n_j\}_{j=1}^n$
  for each child node n
    compute $M_n \subset M_\ell$, the point masses arriving at n
    ascribe a class label to minimize $\sum_{i \in M_n} \alpha_i R_{class}(n, c_i)$
  $U_{branch} = R_{acq}(\ell) + \sum_{n \in \mathcal{N}} \sum_{i \in M_n} \alpha_i R_{class}(n, c_i)$
  if $U_{branch} < U^\star$
    $U_{branch} \leftarrow U^\star$
    $t^\star \leftarrow t$, $\mathcal{N}^\star \leftarrow \mathcal{N}$
if any branch outperforms the non-branching cost
  branch with best diagnostic $t^\star$ and children $\mathcal{N}^\star$

Figure 1: Myopic tree learning algorithm for cost-sensitive feature acquisition a weighted nonparametric distribution

a case of nonrepresentative training data with within-class and between-class imbalance.

Fortunately we can still approximate $P'(f, h, c)$ using the old training set. We posit a factorized representation for the data features with a conditional distribution $P(f|h, c)$ that is common to both the imbalanced training set and the test environment, i.e. $P(f|h, c) = P'(f|h, c) \ \ \forall \ f, h, c$.

This conditional distribution describes observed features for specific hidden properties. It relates to instrument response or noise parameters that may be difficult to describe analytically. One can still use the training data to model this distribution while simultaneously exploiting domain knowledge to model the anticipated hidden class and properties $P'(h, c)$. We approximate a balanced training set by computing a new weighting factor α_i' for each point mass. Consider the following decomposition:

$$\begin{aligned}
\alpha_i' &= P'(f_i, h_i, c_i) \\
&= P(f_i|h_i, c_i)P'(h_i|c_i)P'(c_i) \\
&= [P(f_i|h_i, c_i)P(h_i|c_i)P(c_i)]\frac{P'(h_i|c_i)}{P(h_i|c_i)}\frac{P'(c_i)}{P(c_i)} \\
&= P(f_i, h_i, c_i)\frac{P'(h_i|c_i)}{P(h_i|c_i)}\frac{P'(c_i)}{P(c_i)} \\
&= \frac{1}{Z}\frac{P'(h_i|c_i)}{P(h_i|c_i)}\frac{P'(c_i)}{P(c_i)} \qquad (3)
\end{aligned}$$

Here Z is a normalizing constant. The result is that we weight the point mass by two terms correcting for the between-class and within-class imbalance of the training set.

The numerators $P'(h_i|c_i)$ and $P'(c_i)$ are given by domain knowledge. The denominator terms $P(h_i|c_i)$ and $P(h_i|c_i)$ reflect the "incidental" distribution of the training set and can be estimated directly from the data. We estimate the conditional distribution over properties $\hat{P}(h_i|c_i)$ using a Parzen kernel density estimator. The following expression provides the estimate at h_i using the subset M_{c_i} of training points from class c_i:

$$\hat{f}_{c_i}(h) = \frac{1}{wn}\sum_{j \in M_{c_i}} K(\frac{h - h_j}{w}) \qquad (4)$$

Where n is the number of training points of class c, and K is a Gaussian kernel of bandwidth w. This yields:

$$\hat{P}(h_i|c_i) = \frac{\hat{f}_{c_i}(h_j)}{\sum_{j \in M_{c_i}} \hat{f}_{c_i}(h_j)} \qquad (5)$$

Correcting for the training set's distribution and applying the new prior produces an appropriate weighting factor to best approximate the target class/property distribution.

3.2 Synthetic Example

For a simple example, consider the task of detecting earthquakes among seismic events in a 2D map. Here the available event classes are "earthquake" or "none", and each event has hidden properties corresponding to its latitude and longitude position. A diagnostic for an earthquake might involve querying one or more seismic sensors about the strength of a tremor it received. The result of this test is conditionally dependent on both the class and the event's position. Figure 2 shows an example training set in which earthquake and non-earthquake events are uniformly distributed throughout the domain according to prior class probabilities of 1.0 and 0.9 respectively. Sensors are also distributed randomly, and provide a signal of 1.0 at the epicenter of an earthquake or 0.4 for non-earthquake events. Their signal response attenuates with the squared exponential (Gaussian) of distance to the epicenter, so it is advantageous to test sensors close to the actual earthquake.

We constrain the tree to have binary decisions and the test at each node to be a simple threshold on the signal response of a sensor. The cost structure is quite simple, with a cost of 1 for each sensor polled, 10 for a false positive event, and 100 for a false negative event. However, this is sufficient to produce interesting behavior and sensing policies that intelligently cover the domain. In figure 2, the myopic learning algorithm has chosen several sensors indicated by large "x" shapes. The uniformly-distributed training earthquake events are shown by blue circles.

Suppose prior knowledge suggests that future earthquakes may occur near a known location. This new distribution is a Gaussian centered on the red circle in Figure 2. We assume the overall rate of earthquakes remains constant, but applying the correction for within-class imbalance reweights the training data set to produce the result in Figure 3. The width of each circle indicates the magnitude of the weighting factor α for that datapoint. Note that the new decision tree now polls only two sensors near the new epicenter, reflecting a more cost efficient feature acquisition policy. The thresholds of the tests also change to accommodate the new training set.

We simulated several hundred trials with randomly distributed sensors, earthquakes, and target means. Our tests compared a decision tree trained using the traditional algorithm which ignores the new domain knowledge, a tree trained using our correction for within-class imbalance, and a third

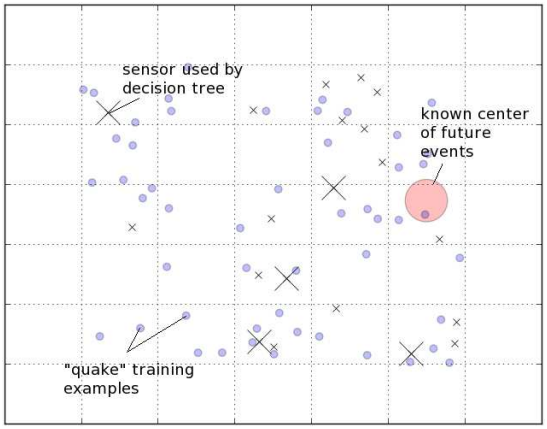

Figure 2: Earthquake example. Training earthquake events are shown in blue; training "no earthquake" events are uniformly distributed and more numerous; we omit these for clarity. Sensors are marked by dark "x" shapes; the larger ones were used in the final decision tree trained on this data. The red circle indicates the known center of future earthquake activity, which is not represented in the training set.

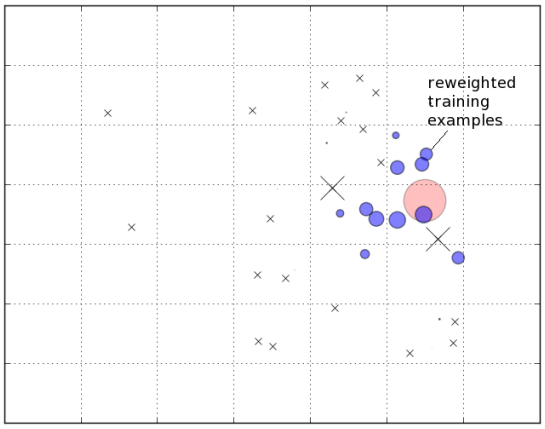

Figure 3: Result of tree learning with correction for within-class imbalance. The width of each blue circle corresponds to the magnitude of the point mass weight α for that earthquake event. The CSDT learning algorithm now chooses only two sensors in the decision tree, representing a more efficient classification strategy. All previous earthquake examples are used as point masses in the new distribution, but most are weighted so lightly that they are invisible in this plot.

tree trained on training data that was drawn from the same Gaussian distribution as the test set. 1000 training points were used in each test.

The performance results on a test set appear in Figure 4. The horizontal axis shows the standard deviation of the width of the test earthquake distribution; in other words, it represents the uniformity of the new prior. The imbalance-corrected tree learning algorithm shows significant cost improvement over the traditional tree, with best performance near an earthquake distribution width of 0.07. Here the future earthquakes are well-localized which permits easy predictive accuracy, but the distribution is still large enough to be represented by the sparse and imbalanced training data.

4. CORRECTING IMBALANCE FOR DETECTION OF RADIO TRANSIENTS

This section provides a more comprehensive case study from the domain of radio astronomy. Our scenario involves real time detection of extremely rare transient signals in radio array data. Transients are short pulses of radio energy, often just a few milliseconds in length, emitted by a variety of exotic astronomical phenomena [4, 8, 13, 7]. Because of their inherent scientific interest, a single such millisecond event may justify an entire weeks' observation. Unfortunately detection of these signals is computationally demanding and computational resources for on-line processing are quite limited. Future arrays will collect far more data than can ever be stored, meaning that most data could be discarded within seconds without having been analyzed. Efficient, adaptive allocation of computing resources is essential. This suggests the domain is amenable to cost-sensitive feature acquisition.

Transients manifest as signal pulses in a time series of radio data. We will consider a simple formulation with two classes: a "pulse" class and a "nopulse" class. Every pulse has three hidden physical properties: an amplitude, a duration, and a *dispersion measure*, or DM, representing the frequency dependence of the signal propagation velocity through the interstellar plasma [2]. A single pulse at frequency ν experiences a time delay according to the equation:

$$\delta t \quad = \quad 4150s \frac{DM}{\mathrm{cm}^{-3}\mathrm{parsec}} \left(\frac{\nu}{1\mathrm{MHz}}\right)^{-2} \qquad (6)$$

In order to detect a broad-band signal that crosses many frequencies, one must first reverse the dispersion with a corrective transformation of the time series. The appropriate DM is not known in advance, so pulse detection requires correcting for many hypothetical DMs. After dedispersion, feature extraction observes the maximum responses of a matched filter convolved across the dedispersed time series. Thus, for n different trial DMs and m different potential filter widths we have a choice of $m \times n$ candidate features f_{jk}, each of which is a single scalar value. Our complete class/property/feature space is:

$$\begin{aligned} \mathcal{C} &= \{pulse, nopulse\} \\ \mathcal{H} &= \{amplitude, duration, DM\} \\ \mathcal{F} &= \{\{f_{jk}\}_{j=1}^{n}\}_{k=1}^{m} \end{aligned}$$

This yields thousands of potential features for each datum. Current transient detection searches use a "brute force" search that uses DMs spaced regularly across the window of interest. Time segments where any filter response exceeds a threshold are excised and stored for more thorough examination. The exhaustive DM search makes detecting pulses computationally expensive, and precludes on-line or real time detection for many installations. Therefore a large quantity of observed data is never analyzed.

We hypothesize that if computational resources are overburdened one could improve detection rates by training a

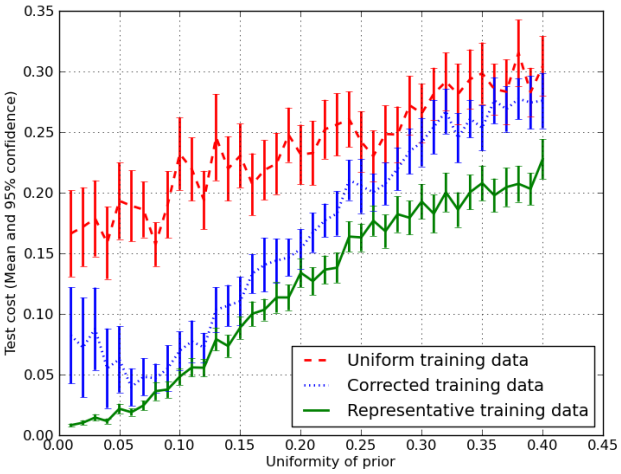

Figure 4: Performance results for standard CSDT learning and CSDT learning with correction for within-class imbalance. The horizontal axis shows the standard deviation of the Gaussian determining locations for the test distribution. As this value increases it becomes more difficult to localize earthquakes. The lower line shows the theoretical optimum in which the training and test sets are drawn from the same distribution.

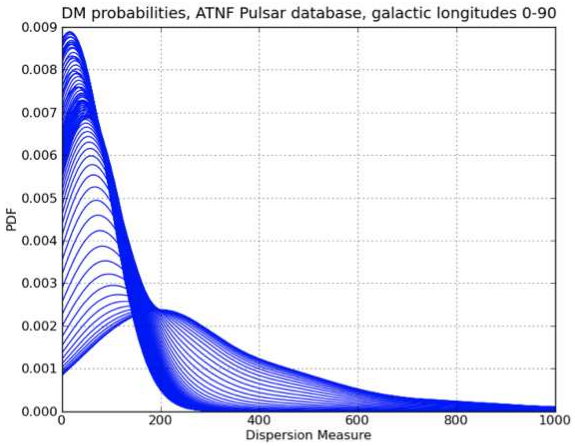

Figure 5: Dispersion measure distributions at different galactic longitude for pulsars in the ATNF pulsar database. Each blue line corresponds to a single pointing angle.

CSDT to recognize pulses efficiently. There are several potential reasons that a CSDT approach could improve performance. One benefit comes from exploiting prior knowledge about source properties. In particular the dispersion measure depends on the distance to the source and the density of ionized plasma along the signal path. This varies throughout the galaxy, and reasonable dispersion measures for our observations range from 0 to several thousand depending on the antenna pointing direction. Figure 5 demonstrates the relationship between pointing angle and the DM property. It shows Parzen density estimation of a set of over 1500 pulsars in the ATNF pulsar database [16], with each blue line representing the probability of a DM at a specific pointing angle. Near the galactic disk and the galactic center, the ionized gas is denser and DMs are usually larger than for sources that are located in directions orthogonal to the galactic disk. As a sample of sources within our own galaxy, the database suggests the possible DMs of future sources that might be discovered with different pointing positions. In the experiments that follow, we will use the ATNF database as a predictive model for future DM measures of galactic sources.

CSDTs offer another possible benefit over status quo pulse detectors. Specifically, they could exploit a hierarchical search strategy to reduce the computational burden without sacrificing accuracy. Figure **??** shows the relationship between signal responses and the difference between the searched and actual DM. A pulse from the Crab Giant Pulsar, with a true DM of 56.75, is dedispersed to various trial DMs and the filter response recorded. Note that the source is still "visible" for a range of DMs around the true value. This suggests that a subset of DMs (red crosses) could provide enough information to direct a more exhaustive local DM search (green diamonds). We hypothesize that a hierarchical scheme, which uses a coarse search and a lenient threshold followed by a fine local search with a higher threshold, could improve computational efficiency without sacrificing detection sensitivity or introducing additional false positives.

A final area where learning strategies might improve performance is in rejection of Radio Frequency Interference, or RFI. These signals often masquerade as pulses, but have distinctive filter signatures that could be exploited by a CSDT that is trained to discriminate pulses.

4.1 Experimental Method

We tested traditional and CSDT architectures with a training set of over 10,000 time segments from the CSIRO Parkes observatory Intermediate and High Galactic Latitude Surveys [10, 6]. This data offers a typical picture of the local radio interference environment around the antenna; spurious weak signals occasionally appear at different DMs. The majority of radio frequency interference (RFI) manifests as stronger signals at low DM. For example, a high response at DM=0 corresponds to zero interstellar dispersion so it is almost certainly a "nonpulse" event associated with local RFI. Each time series contains 96 frequency channels sampled at $125\mu s$ resolution. The data is captured at single-bit quantization, and segmented into time series segments of 1.0s each.

Positive examples are generated using a standard exponential decay pulse profile that is generated at varying signal-to-noise ratios, dispersed to simulate interstellar propagation, and added to one of the empty time segments. The result is a set of training data consisting of pulses generated uniformly from signal-to-noise ratios of $5 - 10$ and DMs $0 - 500$.

The test environment assumes a pointing at galactic latitude 90, longitude 90. Here DMs are *not* drawn uniformly but rather sampled from a galaxy model consistant with the

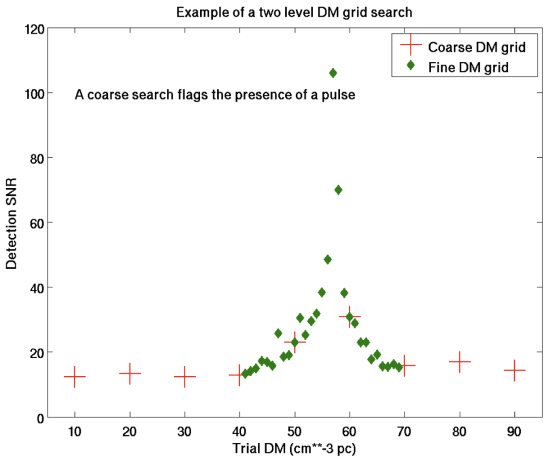

Figure 6: The relationship between the maximum filter response and the estimated DM used during dedispersion. Here we see detection results for the Crab Giant pulsar with a true DM of 56.75. The optimal signal response is achieved when the trial DM is exactly equal to the real DM, but the pulsar is still detectable for a range of DMs around the true value. The image illustrates how a promising response from a coarse DM search (red crosses) could trigger an exhaustive local DM search that achieves a higher SNR (green diamonds).

ATNF pulsar database. This yields two data sets - a nonrepresentative uniform training set and a test set that simulates an actual pointing. Classes are ascribed probability weighting factors to represent the fact that transient pulse events are only 10^{-6} as likely as nonpulse events. We apply a cost structure that reflects the extreme importance of transient events. It ascribes a missclassification cost of 10^6 for a false positive detection and 10^{12} for a false negative.

A fully-unconstrained binary tree, thresholding on a single feature at each node, is infeasibly deep since any analysis will need to test a significant portion of the DMs in the domain. Here we consider three different parameterized tree architectures shown in Figure 7. They are given here in order of increasing flexibility. Each is trained with Algorithm of Figure 1 above.

- A "linear search" approach tests dispersion measures across the domain 0-500, free parameters are the detection threshold and DM search spacing. We will use this as our baseline performance estimate for existing detection systems.
- A "two-tier" tree adds some flexibility to the basic architecture. It searches dispersion measures at a predefined spacing up to a maximum value. A null child node collects time series whose maximal values do not exceed a threshold. It is understood that any time series with a maximum response at DM 0 is probably RFI; these are also routed to the null node. Any surviving candidates are passed to second tier nodes that perform a fine-grained search, exhaustively searching

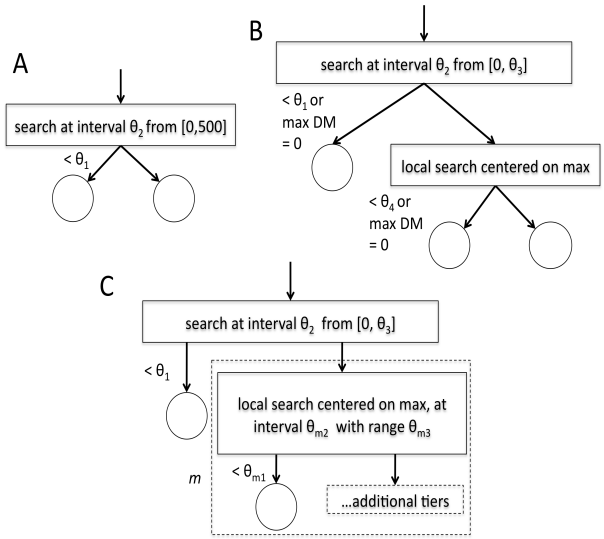

Figure 7: Tree architectures considered in the case study. Free parameters θ are indicated for each tree type, with classification occurring at the leaves (circles). A) Linear search: DM search spacing and threshold are learned. B) Two-tier search, with a high-resolution search centered on the maximally-responding DM from the top level. C) Multi-tier search: each level has a reject leaf as well as a child for each DM included in its search. The dashed box is repeated once for each DM in the search, and a data point is routed to the child associated with the maximal response.

all the intermediate DMs between the maximally-responding DM and its neighbors at the top level. Free parameters are the maximum DM, the spacing of the coarse DM search, the top level threshold, and the shared threshold for the fine DM search. The DM search interval is fixed but the same algorithm could accommodate a nonuniform search with (for example) logarithmic spacings.

- A "multi-tier" method is the most flexible. It assigns a single child node for each searched DM, and propagates the time series to the child associated with the largest filter response. After the initial evenly spaced DM search, the multi-tier adaptive method can grow many additional levels of fine-grained local DM searches at arbitrary new spacings, each centered on the maximally-responding DM from the previous test. Free parameters are the top level spacing, maximum DM, top level threshold, and the spacings, extent, and thresholds of lower-level nodes.

4.2 Results

Figure 8 shows test set performance for each architecture after training on standard and corrected training sets. Naturally overall cost grows with the cost of computation, as tests become more expensive and the CSDTs begin to sacrifice classification accuracy for efficiency. In the low-cost regime, exhaustive searches are still cost-effective but even here the linear search underperforms because it lacks the RFI excision capabilities of the more flexible architectures. "Kinks" in the performance curves represent important discrete changes such as the range of the DM search or the search resolution. For example, the obvious jump in the two-tiered architecture's green performance curve represents an increase in the top-level search interval. The more adaptive methods learn to favor a much higher signal response at low DMs, which eliminates many of these RFI-related false positives.

The two- and multi-tier architectures glean a significant performance benefit by correcting within-class imbalance. DMs above about 250 are less relevant for the test pointing; trees trained on the corrected dataset rarely spend any significant computational resources at high DMs. This improves cost-efficiency for all methods, except in the extreme low-cost regime where the cost of computation is essentially negligible relative to the cost of a single false negative.

In general the two-tier method outperforms the more flexibule multi-tier approach. We attribute this to superior regularization. Despite consisting of several thousand training points, our dataset did not contain enough examples of either "nonpulse" or "pulse" events to fully populate the relevant feature space. This is not a problem for the two-tier method that forces the same threshold across all fine DMs searches. In contrast, the adaptive method often overfits the training data and generates dedicated lower-tier nodes to fit outlier training data. Future work could identify new intermediate parameterizations that offer some increased flexibility without introducing these overfitting problems.

We conclude from these experiments that imbalance correction is feasible for the radio astronomy domain, and offers a

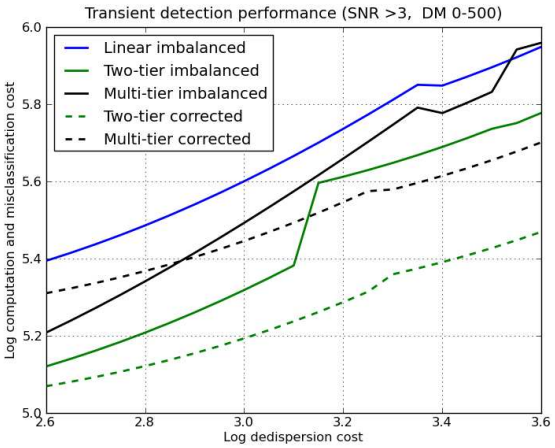

Figure 8: Performance on the Parkes data set for each of the three architectures. Please note the log scales.

principled method for building cost-sensitive decision trees. CSDTs could offer improvement in transient detection accuracy over status quo methods when computational resources are highly limited and an exhaustive DM search is infeasible. Future work could introduce additional parameters, such as the width or shape of the filter applied to the dedispersed time series. One could also introduce additional features of the raw, dispersed time series. CSDTs offer a principled approach for optimizing cost/accuracy tradeoffs for detection architectures that are too complex or heterogeneous for purely analytical performance studies.

5. ACKNOWLEDGMENTS

The work described in this paper was performed at the Jet Propulsion Laboratory, California Institute of Technology, under a JPL Research and Technology Development Grant. We benefited from the counsel and expertise of Radio Astronomers including Larry D'Adrio and Robert Navarro of the Jet Propulsion Laboratory, and the members of the CRAFT collaboration for radio transient research.

6. REFERENCES

[1] M. Andronescu and M. Brodie. Decision tree learning using a bayesian approach at each node. *22nd Canadian Conference on Artificial Intelligence*, 2009.

[2] D. Bhattacharya. Detection of radio emission from pulsars. *NATO ASIC Proc. 515: The Many Faces of Neutron Stars*, 1998.

[3] H. A. Chipman, E. I. George, and R. E. McCulloch. Bayesian cart model search. *JASA*, 93, 1998.

[4] J. Cordes and M.A.McLaughlin. Searches for fast radio transients. *The Astrophysical Journal*, 596:1142–1154, October 2003.

[5] D. G. T. Denison. Simulation based bayesian nonparametric regression methods. *Ph.D Dissertation*, 1997.

[6] R. Edwards, M. Bailes, W. van Straten, and

M. Britton. The swinburne intermediate latitude pulsar survey. *MNRAS*, 326, 2001.

[7] J. L. et al. The dynamic radio sky: An opportunity for discovery. *Astro2010: The Astronomy and Astrophysics Decadal Survey. Arxiv preprint arXiv:0904.0633*, 2009.

[8] J. M. C. et al. The dynamic radio sky. *New Astronomy Reviews*, 48:1459–1472, 2004.

[9] Greiner, Grove, and Roth. Learning cost-sensitive active classifiers. *Artificial Intelligence*, 139(2):137–âĂŞ174, 2002.

[10] B. Jacoby, M. Bailes, S. Ord, R. Edwards, and S. Kulkarni. A large-area survey for radio pulsars at high galactic latitudes. *Astropysical Journal*, 699, 2009.

[11] N. Japkowicz. Concept-learning in the presence of between-class and within-class imbalances. *Proceedings of the Fourteenth Conference of the Canadian Society for Computational Studies of Intelligence*, pages 67–77, 2001.

[12] Ji and Carin. Cost-sensitive feature acquisition and classification. *Pattern Recognition*, 40:1474–âĂŞ1485, 2007.

[13] E. Keane. The search for nearby rrats and other transient radio bursts. *Presentation at the Third Estrela Workshop, Sept. 2008, Bonn, Germany*, 2008.

[14] S. Kotsiantis, D. Kanellopoulos, and P. Pintelas. Handling imbalanced datasets: A review. *GESTS International Transactions on Cmputer Science and Engineering*, 30, 2008.

[15] Ling, Yang, Wang, and Zhang. Decision trees with minimal costs. *In Proceedings of the Twenty-first International Conference on Machine Learning*, page 544âĂŞ551, 2004.

[16] R. N. Manchester, G. B. Hobbs, A. Teoh, and M. Hobbs. *Astron. J.*, 129: astro-ph/0412641, 2005.

[17] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.

[18] Sheng and Ling. Feature value acquisition in testing: A sequential batch test algorithm. *Proceedings of the Twenty-Third International Conference on Machine learning*, pages 809–âĂŞ816, 2006.

[19] P. D. Turney. Cost-sensitive classification: Empirical evaluation of a hybrid genetic decision tree induction algorithm. *Journal of Artificial Intelligence Research*, 2:369–409, 1995.