Available online at www.sciencedirect.com
SCIENCE DIRECT®
Ecological Modelling 191 (2006) 47–57
ECOLOGICAL MODELLING
www.elsevier.com/locate/ecolmodel

Learning to predict channel stability using biogeomorphic features

Stephanie L. Moret [a], William T. Langford [b], Dragos D. Margineantu [c,*]

[a] *Louisiana State University, School of the Coast and Environment, Department of Environmental Studies, Baton Rouge, LA 70803, USA*
[b] *National Center for Ecological Analysis and Synthesis, University of California at Santa Barbara, 735 State Street, Santa Barbara, CA 93101-3351, USA*
[c] *The Boeing Company, Mathematics and Computing Technology, Adaptive Systems, P.O. Box 3707, M/S 7L-66, Seattle, WA 98124-2207, USA*

Available online 28 September 2005

Abstract

Current human land use activities are altering many components of the river landscape, resulting in unstable channels. Instability may have serious negative consequences for water quality, aquatic and riparian habitat, and for river-related human infrastructure such as bridges and roads. Resource management agencies have developed rapid bioassessment surveys to help assess stability in a fast and cost-effective way. While this assessment can be done for a single site fairly rapidly, it is still time-consuming to apply over large watersheds and assessment activities must be prioritized. We constructed a system that employs commonly available map data as inputs to cost-sensitive variants of decision tree algorithms to predict the relative channel stability of different sites. In particular, we use bagged lazy option trees (LOTs) and bagged probability estimation trees (PETs) to identify all unstable channels while making the smallest number of errors of classifying stable channels as unstable, thereby minimizing cost and maximizing safety. We measured the performance of the classifiers using ROC curves and found that the PETs performed better than the LOTs in situations where the number of instances of the stable and unstable classes were relatively balanced, but the LOTs did better where unstable examples were relatively rare compared to stable, perhaps due to the LOTs' ability to focus on individual examples.

Keywords: Channel stability prediction; Cost-sensitive learning; Class probability estimation; Bagging; Lazy option trees; Decision trees

1. Introduction

This research proposes methods for prioritizing and reducing the amount of fieldwork required to assess the status of various environmental conditions such as stream health through the use of cost-sensitive machine learning algorithms. The proposed methods involve learning the outcomes of Rapid [Bio]Assessment

* Corresponding author. Tel.: +1 425 957 5057; fax: +1 425 865 2964.
E-mail addresses: smoret@lsu.edu (S.L. Moret), langfob@nceas.ucsb.edu (W.T. Langford), dragos.d.margineantu@boeing.com (D.D. Margineantu).

0304-3800/$ – see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.ecolmodel.2005.08.011

Protocols (RAPs) from existing data that is commonly available (e.g., topographic maps). A specific RAP for stream channel stability is used as an example.

1.1. Problem background

RAPs are commonly used to collect, analyze, and interpret a variety of stream data to assist diverse management decisions. The focus of these RAPs may be to assess salmon habitat, riparian health, channel stability or any of a number of specific ecological functions for a given region. RAPs are also used in terrestrial settings for a variety of purposes such as habitat, vegetation, and species evaluations.

Many regions have kilometers of unassessed streams but limited resources for stream monitoring and surveying. Cooper et al. (1998) note that "Rapid [Bio]assessment Protocols were created to facilitate cost-effective stream surveys designed to rapidly collect, compile, analyze, and interpret environmental data to assist management decisions." Given this, the conceptual principles of Rapid [Bio]assessment Protocols (Barbour et al., 1997; Cooper et al., 1998) are: "cost-effective, yet scientifically valid procedures, provisions for multiple site investigations in a field season, quick turn-around of results for management decisions, easily translated to management and the public, and environmentally benign procedures". In an effort to meet the principles outlined above, RAPs are often subjective and do not incorporate detailed data collection. Some may question the accuracy of RAPs compared to the results of more rigorous scientific studies of the same sites, but the limited resources of land management agencies often cannot support the costs of more rigorous studies. As a result, RAPs are the method most likely to be used to inventory and assess natural resources by federal and state agencies in the United States and management decisions are often based upon these assessments. In the context of this paper, it does not matter whether we use the outcomes of RAPs or of more detailed assessment methods; the machine learning approach proposed would be the same and RAPs are simply chosen as one example of the approach.

Even aided by RAPs, it is still expensive for federal and state agencies to send investigative teams into the field; hence, many sites go unmonitored. Because of this, it would be useful to have an automated system to prioritize these investigations. Ideally, this system could use readily available office materials and make reliable predictions based on reducing costs (e.g., field costs, property and infrastructure loss) while increasing safety (e.g., prioritizing human lives).

In the late 1970s, the United States Department of Agriculture Forest Service (USFS) designed a RAP to evaluate stream channel stability using data collected from the Rocky Mountains, USA. The method is called the Stream Reach Inventory and Channel Stability Evaluation (SRICSE). It has been used in over 60% of the national forests in the United States (Parrott et al., 1989) and is used by the forest service and others today (Kaplan-Henry et al., 1994; Myers and Swanson, 1996; United States Forest Service, 1992).

In an effort to reduce costs and increase public safety, this research explores how estimates of channel stability may be predicted from data on hydrological, biological, and geomorphological features derived from mapping data commonly available to resource managers. These hydrobiogeomorphic features include sinuosity, topographic gradient, elevation, land use and land cover, and geology. While this study is specific to detecting channel stability in this region, the methods described are intended to be applicable for predicting the outcomes of any Rapid [Bio]Assessment Protocol for any purpose in any region.

1.2. Channel stability

River channel stability was defined by Rosgen (1996) as "the ability of the stream, over time, to transport the flows and sediment of its watershed in such a manner that the dimension, pattern and profile of the river is maintained without either aggrading or degrading". Channel instability is important because it may have negative consequences for aquatic and riparian habitat, and for river-related human infrastructure.

Over the course of thousands of years, a river reaches equilibrium; a state of maximum efficiency in transporting sediment through its basin (Bonneau and Snow, 1992). Equilibrium is influenced by geologic and climatic history, which control the bed and bank material, sediment delivery, vegetation, basin relief, knickpoints, and hydrology (Morisawa, 1968; Mount, 1995). More specifically, rivers exhibit dynamic equilibrium because they must constantly scour and deposit sediment in order to maintain equilibrium despite changes in sediment supply, discharge, and river pattern. While

some instability is natural, excess instability can indicate river dysfunction.

Unstable channels are often the result of changing land use and land cover. Historical land use practices that have disrupted equilibria along the longitudinal profiles of rivers include farming, grazing, forestry, urban development, dams and mining (Collins and Dunne, 1990; Haible, 1980; Mount, 1995; Starkel, 1989). Complex stream channel patterns are simplified when wetlands and floodplains are filled and drained for development, riparian vegetation is cleared, and when banks are stabilized against erosion or leveed for flood protection. These activities can remove roughness by disconnecting a river from its floodplain, reduce water storage capacity and can shorten the river length (its flow path). These factors increase the flow velocity and reduce sediment storage causing channelization, channel simplification, riparian degradation and erosion which all contribute to channel instability.

Erosion caused by channel instability produces sediment, which is sometimes regarded as a non-point source pollutant. It is well documented that excess sediment has contributed to the decline of salmon and other aquatic organisms by causing bed siltation and decreasing water quality (Williamson et al., 1995). Excess material or the wrong size sediment can disable salmon spawning and smother redds. Deep streambed scour can reduce embryo survival in salmonids (Montgomery et al., 1996).

The riparian zone is a key element in both biological and physical stream 'health' and it too can be adversely affected by channel instability. The riparian zone is responsible for filtering recharge water, providing food and habitat for terrestrial and aquatic organisms, providing large woody debris, cooling stream temperature, adding roughness (velocity resistance), and stabilizing banks. Channel bed degradation can drop the water table and result in poor health or death of riparian vegetation. Channel erosion can also directly fell trees by undercutting the supporting banks (Barkhurst and Moret, 1996). Riparian protection should be a critical factor in watershed restoration efforts and maintaining a normative degree of channel stability is important to protecting the riparian zone.

In addition to compromising water quality and causing aquatic and riparian habitat damage, channel instability can be costly to human infrastructure. It can undermine bridge supports, expose pipelines or other structures buried within the riverbed (Collins and Dunne, 1990), and can cause road and trail damage. Channel instability often instigates or increases the intensity and frequency of mass wasting and flooding, resulting in large-scale events such as bank failures and landslides. Such events have destroyed many homes and even lives in the last decade.

It is important to note that channel instability is detrimental only when it is excessive and when and where it negatively impacts aquatic, riparian, and human well being. Channel instability is, geologically, a positive reaction to non-equilibrium and is the river's way of repairing itself (restoring equilibrium). A natural river in dynamic equilibrium has both stable and unstable areas, which vary in frequency and magnitude over space and time.

2. Methods

2.1. Data collection

The channel stability fieldwork was conducted on a select portion of third and fourth order drainage systems at high elevations in the Upper Colorado River Basin, Colorado, United States of America. These rivers were selected because they are longer, larger and more accessible than first and second order streams, and more diverse in their hydrobiogeomorphic features and land use.

The Upper Colorado River basin was selected for several reasons. The primary consideration is that the most recognized method for evaluating channel stability, the SRICSE, was developed in the Central Rocky Mountain region. This region also has large land tracts of relatively undisturbed forest, thus reducing confounding variables and exemplifying the problem of limited worker resources covering a vast area.

To evaluate channel stability with the SRICSE method, a series of inventory items must be completed using maps, field observations, and field measurements. The stream reach inventory is a short, simplified list of the location, basic hydraulic and geomorphic properties, and water quality components. Channel stability is evaluated by assigning scores from a rating sheet to 15 channel stability attributes related to stability of the channel bottom, lower bank and upper bank (seven discrete and four continuous attributes).

Attributes include landform slope, mass wasting, debris jam potential, vegetative bank protection, channel capacity, bank rock content, obstructions and flow deflectors, cutting, deposition, rock angularity, moss content, consolidation, stable materials, scouring and deposition, and aquatic vegetation. A small handbook (Pfankuch, 1978) is used to guide the rating choices. These attribute scores are added together to get a total reach score (TRS) for each stream. Higher TRS values indicate more unstable channels.

Fifty-five streams were surveyed at randomly selected sites during the summer of 1997. Each survey had one replicate survey taken, for a total of 110 surveys. The survey and replicate were averaged for each stream. The replicates were taken one-eighth of a mile upstream from the randomly selected survey site. Each survey length was 8–12 times the channel width. Over the 55 streams, the stability measures ranged from 55 to 117.5.

Site-specific information that is not directly measured by the SRICSE was taken from both paper maps and a Geographic Information System (ESRI ArcView combined with EPA Basins). This information includes hydrobiogeomorphic features thought to influence channel stability and be commonly available without requiring any fieldwork. The continuous variables measured were topographic stream gradient, sinuosity, elevation, and precipitation. Categorical variables measured were geology, and land use and land cover.

2.2. Defining classes by thresholding

We are interested in making a binary decision in this problem and the original target data (stability measures) is in the form of real numbers instead of discrete classes. Consequently, we have transformed the problem from its original regression format into several two-class classification problems in which the classes are stable and unstable. These classes correspond to whether investigators will need to go in the field to assess the channel or not. Unstable channels are assumed to have the potential for incurring large losses and therefore, require further field investigation. Stable channels are assumed to have low risk and require no field investigation. Membership in the class is defined by thresholding the total reach scores. For example, if a threshold of 80 is chosen, then any reach whose reach score is at least 80 will be assigned to the unstable class, and any reach having a score below 80 will be classed stable.

Note that there is no "correct" threshold in our study. The threshold would be chosen by the individual resource expert according to their specific geographical area and problem. We have shown a range of thresholds so that we could examine the behavior of the algorithms in a variety of conditions. We have defined different classification problems for each of the possible different values of the stable-unstable threshold θ in the interval [80, 95]. The values 80 and 95 were chosen because values less than 80 are unlikely to ever be considered unstable and values greater than 95 are rare and therefore, provide too small a sample of unstables for reliable learning. While we could have defined more than two classes (e.g., good, fair, and poor), we chose to use only two classes because we do not have cost information for more than two classes and we are investigating this problem in the context of decision costs.

2.3. Classification costs

The difficulty in modeling the problem as a classification task is that when making a decision, one class of errors (misclassifying unstable channels) is much more expensive than the opposite error. Standard statistical and machine learning techniques attempt to minimize the total number of mistakes (in the case of classification) and mean squared error (in the case of regression), without regard to their relative costs. For example, an error that may result in destruction (predict stable when it is really unstable) is treated as no more or less important than an error that results in inconvenience (predict unstable when it is stable). Such treatment is unsuitable for this application because our primary objectives are to reduce cost (i.e., damages) and increase safety. Costs related to channel instability might include: agricultural decline due to a lower water table or bank failure, riparian decline caused by a lower water table, fish habitat loss, property loss and loss of lives, as well as infrastructure loss if bridges or pipelines are undermined.

While we do not have precise values for the costs of the different decisions this problem, we do know the approximate range of values for the cost matrix. The ranges of the costs are represented in Table 1

Table 1
The cost matrix for the river channel stability task, representing the ranges of costs associated with each decision

Actual	Predicted	Result	Cost (US$)
Stable	Unstable	One day labor	$f_p \approx 10^3$
Unstable	Stable	Possible loss of infrastructure, property, lives	$f_n \geq 10^7$
Unstable	Unstable	One day plus full cost of remediation	$t_p \approx 10^3$ to 10^4
Stable	Stable	No effect, no action required	$t_n \approx 0$

(in US$). The largest cost is the cost of the unstable channels that are classified as stable, a value in the order of tens of millions (representing the losses of human lives and losses of property that can be caused by an unstable channel that was not remediated). An incorrect classification of a stable channel has associated with it the cost of 1 day's work of an expert sent to evaluate the channel. A correct classification of an unstable channel will incur the cost of 1 day's expert work plus the cost of the remediation. The dominating cost however, corresponds to the risk of misclassifying an unstable river channel. Therefore the objectives of the learned classifiers are to classify all unstables correctly while having as few misclassified stables as possible.

2.4. Decision trees

There are a number of machine learning techniques that could be used for this research. We chose decision trees for the analysis because they have good performance, few control parameters, are fast to train, yield relatively comprehensible models, and the software is readily available (Breiman et al., 1984; Quinlan, 1986, 1993). These algorithms produce tree-structured models for classification consisting of internal nodes and leaves where the internal nodes specify tests on attribute values while the leaves specify class labels.

To classify an unlabeled example, the decision tree will perform the attribute test specified by the root node and follow the branch corresponding to the outcome of the test. In the case of reaching another internal node, the instance will traverse the tree through the branches corresponding to the outcomes of the tests, all the way to a leaf node. When a leaf node is reached, the example will be assigned the class label specified by the leaf.

To train the classifier, all possible attribute tests are considered and assessed (on the training data) at each node based on some evaluation function (usually a heuristic). The test with the highest score is chosen, and the training data is split using the test. The induction procedure is called recursively for each resulting partition of the data. The splitting process halts when any one of the following conditions is satisfied: (1) all training instances reaching the node belong to the same class, (2) all training instances have the same attribute values, (3) the number of instances is smaller than the minimum number allowed (a parameter given by the user), or (4) the assessment heuristic indicates that no further improvement of the model can be achieved. Most of the heuristic evaluation functions for choosing a test that are used in practice, make use of some measure of the purity of the data (i.e., seek tests that lead to nodes in which the number of instances in one class is much larger than the instances from other classes, ideally having only instances from a single class). When the tree model is constructed, a pruning procedure is often employed to avoid overfitting the input data.

2.5. Learning good class probability estimates

Given the fact that we need classifiers that are cost-sensitive, the most flexible approach to handle the costs is to employ classifiers as class probability estimators (or, conditional density estimators) and compute the optimal decision based on the estimated probabilities and the decision costs.

In general, in the case of a classification task, an instance x should be labeled with the class γ that minimizes the conditional risk (or, expected loss) for that instance

$$R(\gamma|x) = \sum_{j=1}^{K} P(j|x)C(\gamma, j) \quad (1)$$

where K is the number of classes (in our case, 2: *stable* and *unstable*) and C is the $K \times K$ cost (or loss)

matrix ($C(i, j)$ is the cost associated with classifying an instance that is in class j as being in class i). This equation gives us the optimal labels if the probabilities $P(j|x)$, for all $j = 1, \ldots, K$, are accurately computed.

All classification algorithms can be converted into class probability estimators. However, because most of these algorithms were designed to learn models that try to minimize the misclassification error (and not for probability estimation), the computed probabilities are inaccurate in most cases.

In the case of decision trees, the class probability estimates $P(y|\boldsymbol{x})$ for an unseen instance $\boldsymbol{x}$ that reaches a leaf l, are approximated using the class counts of the training instances that reach l. For example, if a leaf is reached by eight instances from class *stable* and 0 instances from the class *unstable*, the probability estimated using the tree, for an instance $\boldsymbol{x}$ that reaches the same leaf is: $P(\text{stable}|\boldsymbol{x}) = 8/8 = 1.0$ and $P(\text{unstable}|\boldsymbol{x}) = 0/8 = 0.0$.

As noted by different studies (Bradley, 1997; Provost and Domingos, 2003; Provost et al., 1998; Smyth et al., 1995), the class probability estimates of the decision trees are poor. There are three major factors that cause this deficiency. First, the greedy induction mechanism that splits the data into smaller and smaller sets leads to probability estimates that are computed based on very small samples, and this leads to inaccurate estimates. Second, most of the existing decision-tree induction algorithms focus on minimizing the number of misclassifications (through the purity-based heuristics) and on minimizing the size of the model (through the pruning procedure). This causes the learned models to compute class probabilities that are too extreme (i.e., close to 0.0 and 1.0), as in the example above, and therefore incorrect. The third factor is the shape of the decision tree hypotheses (piecewise linear decision boundaries). This kind of decision space assigns uniform probability values to points that are in the same region and will not differentiate between points that are closer to the boundary of the region and points that are farther from the boundary.

The following sections will present two approaches for learning more accurate probability estimates for river channel stability and will show how these methods have been employed for the task of classifying river channels. The first method, bagged probability estimation trees (B-PETs) addresses the problem of having extreme probability values at the leaves. The second method, bagged lazy option trees (B-LOTs) addresses the problem of differentiating between values at different distances from decision boundaries.

2.6. Probability estimation trees (PETs)

Provost and Domingos (2003) show that decision tree class probability estimates can be improved by skipping the pruning phase and smoothing the distributions by applying a Laplace correction (or Dirichlet prior) as follows:

$$P(y_j|x) = \frac{N_j + \lambda_j}{N + \sum_{i=1}^{K} \lambda_i} \tag{2}$$

where N is the total number of training examples that reach the leaf, N_j the number of examples from class y_j reaching the leaf, K the number of classes, and λ_j is the prior for class y_j (assumed to be uniform $\lambda_i = 1.0$ for all $i = 1, \ldots, K$ in this case, and in all other applications in which there is no prior knowledge about the distribution of the instances). The Laplace correction (Bradford et al., 1998; Cestnik, 1990; Good, 1965) will smooth probability estimates that are too extreme because of the small size of the sample that reaches the leaf. This smoothing permits probability estimation trees to reduce the effects of the second of the causes for inaccurate estimates (extreme probabilities), described in the previous section.

To handle the other two sources of inaccuracy of tree-based probability estimates, Provost and Domingos apply Bagging (Breiman, 1996). Bagging averages the probabilities computed by multiple models. Each of the models is trained using a bootstrap replicate (Efron and Tibshirani, 1993) of the training data. The resulting models are called bagged probability estimation trees (or B-PETs).

2.7. Lazy learning

If a point to be classified lies near a decision boundary, then points within the same decision region that will be used to compute the class probabilities are likely to be farther away from it on average (i.e., less similar) than they would be if the test point was in the center of the decision region. Standard supervised learning (such as decision tree induction) does not make use of any knowledge about the points to be classified (such

as their attribute values), so it is unable to guarantee that the decision boundaries will be far from the test point. Lazy learning is a framework in which a model is built only when the attribute values of the instance to be classified are known. Using this knowledge, decision boundaries can be chosen to guarantee that they do not fall near the test point. Commonly, under this framework a model is built for each individual unlabeled test instance. The most popular example of a lazy learning algorithm is the nearest neighbor algorithm (Dasarathy, 1990; Wettschereck, 1994), which classifies an example based on the labels of the instances that are most similar to it (according to some distance measure).

The vast majority of the efforts in lazy learning have focused on accurate 0/1-loss classification (Aha, 1997). We have used a new lazy algorithm for accurate estimation of class probabilities. Our decision to employ the lazy learning framework is based on the intuition that lazy models can more accurately capture the specific characteristics of previously unseen instances and the neighborhood around them and therefore may be able to compute better probabilities—especially for tasks like our river channel stability problem, in which limited training data is available. The disadvantage of lazy learning is that classification time can be significantly larger for lazy algorithms, and this can affect their utility in some practical applications. For our channel stability classification task, we employed a method based on the lazy tree learning algorithms proposed by Margineantu and Dietterich (2002).

2.7.1. Lazy trees

A combination of the lazy learning idea and decision tree algorithms is an algorithm called lazy decision trees introduced by Friedman et al. (1996). The lazy decision tree algorithm builds a separate decision tree for each test instance (the query point). Because only one instance has to be classified by the learned model, each internal node will have only one outgoing branch—the one that tests positive on the query point. For each internal node, the selected test will be the test that maximally decreases the entropy for the node to which the test instance would branch. The information gain is defined to be the difference between the two entropy values. Class probabilities are estimated based on the counts of the classes of the instances reaching the leaf node as described in Eq. (2).

2.7.2. Multiple options at nodes: lazy option trees (LOTs)

Although lazy trees attempt to help avoid inaccurate estimates of points too close to the decision boundary, the small sample size at the leaves and the greediness of the induction method still influence the quality of the probabilities. The greedy selection of tests used for splits means that at a given split, there may be more than one test with a similar information gain, but only the single split with the highest gain will be chosen. To correct for this, instead of having a single test in the internal nodes, we allow multiple tests (options) in each node to grow lazy option trees (or, LOTs). This idea extends Buntine's (1990) and Kohavi and Kunz's (1997) ideas of option decision trees for classification into the lazy tree learning framework.

Given a specified number t of options to allow, in each node the lazy option tree algorithm selects the t-tests with the highest information gain. To compute the class probability $P(y|\boldsymbol{x})$, the algorithm will calculate the proportion of training examples from class y from each of the tree leaves and will average the values. This corresponds to taking all the paths from the root node to the leaf node. Because the tree was built just for $\boldsymbol{x}$, all the tests on these paths will be satisfied by the attribute values of $\boldsymbol{x}$.

The primary advantage of the options mechanism is that tests having an information gain almost as high as the best test will be performed, while they might never be performed in decision trees, lazy trees, or even bagged trees. This way, diversity is increased, and this might lead to better probability estimates. In addition, lazy option trees (LOTs) offer a single compact model that is comprehensible because the rules extracted are very similar to the rules extracted from regular decision trees and the options can be interpreted as logical disjunctions.

The options represent an alternative to the voting mechanism of bagging for smoothing and improving the probability estimates of the tree models. Nonetheless, to improve the class probability estimates of the LOTs we propose applying bagging on top of the algorithm, resulting in bagged lazy option trees (B-LOTs). Our intuition is that the different nature of the two mechanisms, options and bagging, will help improving the computed estimates.

In the case of LOTs and B-LOTs, the user will have to set two additional parameters: max T is the maximum

number of tests that are allowed in a node, and min G is the minimum gain proportion ($0.0 < \min G < 1.0$), a number indicating the minimum gain for a test in order to be selected.

For all experiments that involved LOTs and B-LOTs we have chosen the value for the maximum number of tests allowed in a node to be $\max T = 3$ and the gain proportion $\min G = 0.2$. Preliminary experiments show that although the complexity of the trees grows a lot with the value of $\max T$, the results do not show major improvement of the LOTs or B-LOTs as $\max T$ is assigned larger values (we tested for $\max T = 5$, $\max T = 7$, and $\max T = 10$). In the meantime, larger values of $\max G$ have proven to hurt the overall performance of the algorithm.

3. Experimental results

3.1. ROC curves

A popular method for the evaluation of classifiers is the receiver operating characteristic, or ROC (Egan, 1975; Provost and Fawcett, 1997). ROC curves assume that the classification algorithm to be evaluated can output some instance class rankings. By changing the thresholds for making the decisions (based on the ranking), the learned classifier will output the whole spectrum of hypotheses that are computable by that classifier.

An ROC curve shows the different cost tradeoffs available for a given algorithm for different decision threshold values. The origin of the graph corresponds to the classifier that always predicts the negative class. The (1, 1) point of the graph corresponds to the classifier that always predicts the positive class. The best possible performance would occur at the point (0, 1) (corresponding to a false positive rate of 0 and a true positive rate of 1), where all examples would be classified correctly. All classifiers above the diagonal from (0, 0) to (1, 1) have predictions that are better than random, while the ones below have an accuracy worse than random. There have been several attempts at inferring metrics from the ROC curve that would robustly evaluate the performance of learning algorithms. The most commonly used of these metrics is the area under the ROC curve or, AUC (Bradley, 1997).

We generated an ROC curve for each classifier at each value of the stable/unstable threshold. There are too many curves to show them all here, but Fig. 1 gives one example of two ROC curves generated when θ is set to 86, one for the B-LOT classifier and one for the B-PET. We can compare the performance of the two classifiers across a range of parameter settings without regard to cost by comparing the area under each of their ROC curves (AUC), with a larger area implying a better classifier. We can also compare the performance of the two classifiers with respect to cost. Using this measure, the best performance for each of these classifiers corresponds to the point where they can recognize

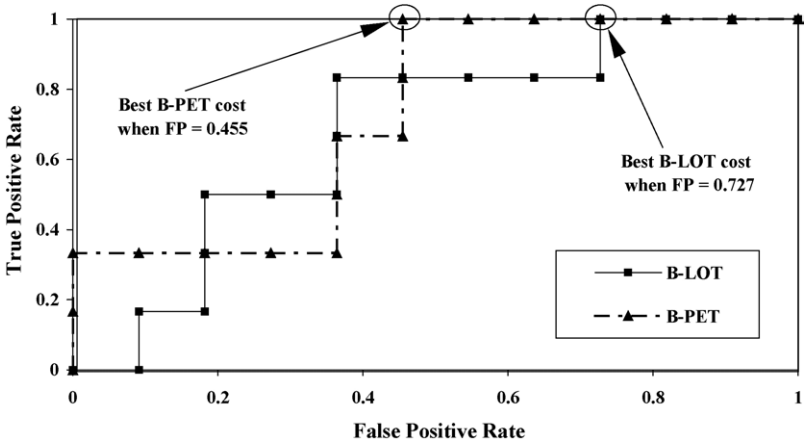

Fig. 1. ROC curves for B-LOT and B-PET classifiers for stable/unstable threshold $\theta = 86$.

100% of the true positives with the smallest number of false positives. The next two sections give the details for our tests using each type of the two types of measure.

3.2. Overall performance without using cost measures (AUC)

We first evaluated the algorithms in the absence of any cost matrix using the area under the ROC curve (AUC) as the performance measure. To do this, we employed leave-one-out cross-validation (LOOCV) (Fukunaga, 1972; Kearns and Ron, 1999), because of the small size of the available data sample. Table 2 shows the values of the AUC for the two algorithms, for different values of the θ parameter. By this measure, the B-LOTs had better overall performance than the B-PETs for all values of θ except for the [90, 93] range.

3.3. Overall performance using cost measures

For each threshold value θ, we have applied the bagged LOTs (B-LOTs) and the bagged PETs (B-PETs) to learn the ranking of the river channels to make the decision that produces the smallest number of false positives (stables classified as unstable) while finding all of the true positives (correctly labeled unstables). In the case shown in Fig. 1, that point for the B-LOT classifier is at the point where it gets 72.7% false positives. For the B-PET classifier, the best value is at 45.5% false positives, so it is the better of the two classifiers for the threshold $\theta = 86$ when errors are expressed in terms of cost. The results for all tested values of θ are summarized in Table 3. The values in the table correspond to the circled points shown in Fig. 1 for each value of θ. By this measure, the performance was nearly reversed from the AUC measure. The B-LOTs were better for θ in the [90, 95] range and the B-PETs were better for θ less than 90.

Table 2
Overall performance without using cost measures (AUC)

θ	B-LOTs	B-PETs
80	**0.813**	0.743
81	0.839	0.814
84	0.818	0.795
86	**0.789**	0.743
88	**0.78**	0.652
90	0.673	0.684
93	0.61	**0.656**
95	0.671	0.652

The table shows the values of the area under the ROC curve (AUC) for the two algorithms (B-LOTs and B-PETs) for different values of θ (the value that sets the threshold between stable and unstable river channels). These values were computed using leave-one-out cross-validation. The results printed with bold face indicate the values corresponding to the algorithm that was better for a particular setting of θ at a significance level of $p < 0.05$.

Table 3
Overall performance using cost measures

θ	B-LOTs	B-PETs	Fraction unstable sites in dataset
81	0.33	**0.22**	0.49
84	0.60	**0.40**	0.45
86	0.72	**0.45**	0.42
88	0.73	**0.55**	0.36
90	**0.47**	0.58	0.29
93	**0.50**	0.75	0.25
95	**0.28**	0.50	0.20

The values represent the proportion of stable channels that were misclassified (out of the total number of stable channels) when all unstable channels were classified correctly. θ is the value of the stability factor that sets the threshold between stable and unstable river channels. The table reports the smallest false positive rate corresponding to a true positive rate of 1.0. The values in bold face indicate the classifier with the better score. The rightmost column shows the fraction of sites in the dataset that are unstable at the given threshold θ. This reflects the relative balance of the two classes stable and unstable.

4. Discussion and conclusions

The fact that the AUC for one algorithm A_1 is better than the AUC for another algorithm A_2 while the cost or error rate of A_2 is lower than the cost (or error rate) for A_1 shows how important is the assessment method that is employed. AUC (Table 2) assesses the performance of an algorithm over all possible costs/decisions, whereas the values in Table 3 essentially give the loss incurred by the algorithms for a given decision threshold.

Our results show that both the B-LOTs and the B-PETs can learn to recognize every example of an unstable channel in the dataset without labeling all of the stables as unstable. However, the misclassification rate for calling stables unstable is often in the neighborhood of 50% even if we are allowed to always choose

the better of the two classifiers. This is not surprising given the complexity of the problem and the small sample size for training, particularly at higher thresholds. Nevertheless, there is still a substantial cost savings for the user since it means that there are many stable sites that do not have to be visited while it is highly probable that every unstable site does get visited. Moreover, these results have further practical value in that they did not require knowledge of exact costs (only order of magnitude) and the system can be interfaced to a geographic information system (GIS) to provide assessments along the full length of all channels in a large region.

Our results also show that if the classifier's internal decision threshold (probability above which an example is labeled unstable) is set optimally, the B-PETs give lower costs when $\theta < 90$ and the B-LOTs are better for larger values of θ. It is important to observe that larger values of θ correspond to fewer unstable examples in the data, creating a class imbalance. This suggests that lazy learning was able to handle larger class imbalance better than the B-PETs. The intuition on why the lazy trees tended to perform better on imbalanced data is that the lazy trees focus separately on each previously unseen instance and build "more carefully" the decision boundaries around the "rare" or "minority" points, and thus, compute better estimates for these points.

We should also note that while we have only shown results here for B-PETs and B-LOTs, we have evaluated several other methods for this application and found them unsuccessful. This was part of the motivation for investigating the B-LOTs and B-PETs. In similar earlier work (Moret, 2001), we tried C4.5 and linear regression. There we found that the only threshold where C4.5 could only find all of the unstables was at $\theta = 81$ because of its poor probability estimates. There was no threshold where linear regression could identify all of the unstables correctly. In the current study, we also tried logistic regression and support vector machines and got similar poor results. While support vector machines have been successful in many applications, they are not appropriate for probability estimation because of their regularizer which tries to maximize the margin of the separating hyperplane. This margin maximization leads to extreme class probability estimates (that are close to 1.0 or 0) (Provost and Domingos, 2003; Platt, 1999). In spite of these results, other methods need to be considered in future work, for example, random forests (Breiman, 2001). This paper has compared the performance of two types of classifiers based on probability estimation trees for the purpose of classifying stream channels into categories of stable and unstable with the goal of insuring the recognition of all unstable channels and minimizing the number of misclassified stable channels. We found that B-PETs performed better at lower thresholds corresponding to more balanced distribution of training examples and B-LOTs performed better in the more unbalanced case, perhaps due to their ability to focus on individual examples. Regardless of whether these are the best methods for probability estimation, we have shown that a cost-based learning approach can increase safety and reduce costs in a difficult practical application where cost-free methods may fail.

References

Aha, D., 1997. Special issue on lazy learning. Artif. Intell. Rev..

Barbour, M.T., Gerritsen, J., Snyder, B.D., Stribling, J.B., 1997. Revision to rapid bioassessment protocols for use in streams and rivers. Off Water. EPA-841-D-97-002.

Barkhurst, M., Moret, S.L., 1996. Post-flood assessment of woody debris in oak creek. In: Proceedings of the Second Annual Pac NW Water Issues Conference. American Institute of Hydrologists.

Bonneau, P.R., Snow, R.S., 1992. Character of headwaters adjustment to base level drop, investigated by digital modeling. Geomorphology 5, 475–487.

Bradford, J.P., Kunz, C., Kohavi, R., Brunk, C., Brodley, C.E., 1998. Pruning decision trees with misclassification costs. In: Proceedings of the European Conference on Machine Learning, Lecture Notes in Artificial Intelligence, vol. 1398. Springer, pp. 131–136.

Bradley, A.P., 1997. The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern Recog. 30, 1145–1159.

Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J., 1984. Classification and Regression Trees. Wadsworth International Group.

Breiman, L., 1996. Bagging predictors. Machine Learning 24, 123–140.

Breiman, L., 2001. Random Forests. Technical Report. Department of Statistics, University of California, Berkeley.

Buntine, W, 1990. A theory of learning classification rules. Technical Report. School of Computing Science, University of Technology, Sydney, Australia.

Cestnik, B., 1990. Estimating probabilities: a crucial task in machine learning. In: Aiello, L.C. (Ed.), Proceedings of the Ninth European Conference on Artificial Intelligence. Pitman, pp. 147–149.

Collins, B.D., Dunne, T., 1990. Fluvial Geomorphology and River-Gravel Mining: A Guide for Planners; Case Studies Included, Special Pub. California Division of Mines and Geology.

Cooper, J., Rediske, R., Northup, M., Thogerson, M., Van Denend, J., Annis, R., 1998. History Of The Rapid Bioassessment Protocols: The Agricultural Water Quality Index, WRI Publication #MR-98-1. Michigan Water Resources Institute.

Dasarathy, B.V., 1990. Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques. IEEE Computer Society Press, Los Alamitos, CA.

Efron, B., Tibshirani, R.J., 1993. An Introduction to the Bootstrap. Chapman & Hall.

Egan, J.P., 1975. Signal Detection Theory and ROC-analysis. Academic Press.

Friedman, J.H., Kohavi, R., Yun, Y., 1996. Lazy decision trees. In: Proceedings of the 13th National Conference on Artificial Intelligence. AAAI Press/MIT Press, pp. 717–724.

Fukunaga, K., 1972. Introduction to Statistical Pattern Recognition. Press Academic.

Good, I.J., 1965. The Estimation of Probabilities: An Essay on Modern Bayesian Methods. MIT Press.

Haible, W.W., 1980. Holocene profile changes along a California coastal stream. Earth Surf. Process. 5, 249–264.

Kaplan-Henry, T.A., Henry, W.T., Eddinger, H., 1994. Sequoia National Forest South Creek Riparian ecosystem analysis. Watershed Manage. Council 6 (2), 5–6.

Kearns, M., Ron, D., 1999. Algorithmic stability and sanity-check bounds for leave-one-out cross-validation. Neural Comput. 11 (6), 1427–1453.

Kohavi, R., Kunz, C., 1997. Option decision trees with majority votes. In: Proceedings of the 14th International Conference on Machine Learning. Morgan Kaufmann, pp. 161–169.

Margineantu, D.D., Dietterich, T.G., 2002. Improved class probability estimates from decision tree models. In: Denison, D.D., et al. (Eds.), Nonlinear Estimation and Classification, Lecture Notes in Statistics, vol. 171. Springer, pp. 169–184.

Montgomery, D.R., Buffington, J.M., Peterson, N.P., Schuett-Hames, D., Quinn, T.P., 1996. Can. J. Fish. Aquat. Sci. 53, 1061–1070.

Moret, S.L., 2001. Predicting channel stability in Colorado mountain streams using hydrobiogeomorphic and land use data: a cost-sensitive machine learning approach to modeling rapid assessment protocols. Ph.D. Dissertation. Env. Sciences, Oregon State University, 2001.

Morisawa, M., 1968. Streams—Their Dynamics and Morphology. McGraw-Hill, Inc., New York.

Mount, J.F., 1995. California Rivers and Streams—The Conflict between Fluvial Process and Land Use. University of California Press, Berkeley, CA.

Myers, T.J., Swanson, S., 1996. Temporal and geomorphic variations of stream stability and morphology: mahogany creek, Nevada. Water Res. Bull. 32 (2), 253–263.

Parrott, H., Marion, D.A., Perkinson, R.D., 1989. A four-level hierarchy for organizing wildland stream resources information. In: Woessner, W.W., Potts, D.F. (Eds.), Proceedings of the Symposium on Headwaters Hydrology. American Water Research Association, pp. 41–54.

Pfankuch, D.J., 1978. Stream Reach Inventory and Channel Stability Evaluation—A Watershed Management Procedure. US Department of Ag. Forest Service, Northern Region 10, Intermountain Forest and Range Experiment Station, Ogden, UT, USA.

Platt, J., 1999. Probabilistic outputs for support vector machines and comparisons to regularize likelihood methods. In: Smola, A., Bartlett, P., Schoelkopf, B. (Eds.), Advances in Large Margin Classifiers. MIT Press.

Provost, F., Domingos, P., 2003. Tree induction for probability-based ranking. Machine Learning 52 (3), 199–215.

Provost, F., Fawcett, T., 1997. Analysis and visualization of classifier performance: comparison under imprecise class and cost distributions. In: Proceedings of the Third International Conference on Knowledge Discovery and Data Mining. AAAI Press, pp. 43–48.

Provost, F., Fawcett, T., Kohavi, R., 1998. The case against accuracy estimation for comparing classifiers. In: Proceedings of the International Conference on Machine Learning, pp. 445–453.

Quinlan, J.R., 1986. Induction of decision trees. Machine Learning 1 (1).

Quinlan, J.R., 1993. C4.5: Programs for Machine Learning. Morgan Kaufmann.

Rosgen, D., 1996. Applied River Morphology. Wildland Hydrology. P.Springs, Colorado.

Smyth, P., Gray, A., Fayyad, U., 1995. Retrofitting decision tree classifiers using kernel density estimation. In: Proceedings of the 12th International Conference on Machine Learning, pp. 506–514.

Starkel, L., 1989. Valley floor evolution in the marginal areas of the Himalaya Mountains and the Khasi-Jaintia plateau. Zeits fur Geomorphologie Supplentband 76, 1–8.

United States Forest Service (USFS), 1992. Integrated Riparian Evaluation Guide. Intermountain Region, Ogden, UT.

Wettschereck, D., 1994. A study of distance-based machine learning algorithms. Technical Report. Oregon State University, Corvallis, OR.

Williamson, K.J., et al., 1995. Gravel Disturbance Impacts on Salmon Habitat and Stream Health, vol. II: Technical Background Report. Oregon Water Resources Research Institute.