

Effective Classification for Crater Detection: A Case Study on Mars

Jue Wang, Wei Ding

Barry Fradkin, Cuong H. Pham, Peter Sherman, Binh D. Tran,

Dawei Wang, Yun Yang

University of Massachusetts Boston

100 Morrissey Blvd

Boston, MA 02125, USA

E-mail: {jue.wang002, wei.ding, barry.fradkin001, cuong.pham001, peter.sherman001, binh.tran004, dawei.wang001, yun.yang001}@umb.edu

Tomasz F. Stepinski

Lunar and Planetary Institute

3600 Bay Area Blvd.

Houston, TX 77058

tom@lpi.usra.edu

Abstract—Craters are important geographical features caused by the impacts of other celestial bodies. Craters have been widely studied because they contain crucial information about the age and geologic formations of a remote planet. This paper discusses an automated crater-detection framework using knowledge discovery and data mining (KDD process) including sampling, feature selection and creation, and supervised learning methods. The framework is evaluated on a real world case study on Mars crater detection. Compared with the existing method, the detection rate of F-measure is improved from 0.613 to 0.772 on a studied Martian site of area 451,562,500 m².

Session—Data Mining

Index Terms—Pattern Classification, Data Sampling, Data Preprocessing, Crater Detection Algorithm, Mars

1. INTRODUCTION

Impact craters, a.k.a. craters, are formed by high velocity collisions of solid celestial bodies onto the planet's surface. Up to this point, scaling impact craters from remotely sensed images has been the only way to deduce the chronologic information of a remote planet. Furthermore, surfaces of planets which process more active ground motions contain less visible craters. Thus the ability of counting craters enables the effective study of the geology of a remote planet, and Mars has been at the center of Solar system exploration. Craters on Mars contain information about the past and present geological processes of this red planet. Mars craters larger than 5 km in diameter, about 42,283 craters in total, have already been manually counted and catalogued [1] from planetary images. However, the huge sheer number of small sub-kilometer craters makes it infeasible for manual detection. This paper focuses on automatic detection of sub-kilometer craters.

The unique challenges of crater detection from remote images are as follows. Because of the eroding, burying, overlaying, and transforming problems during a crater life cycle, detecting circular-shape craters is more difficult than simply detecting perfect circles from images. Instead of using edge detection for circular crater rims, we observe that a crater can be recognized

as a pair of highlight and dark crescent regions. First, image processing techniques, described in our early work in [2], are applied to remote sensing images, including removing irrelevant background, identifying crescent shapes, constructing crater candidates by combining highlight and dark regions. Second, we design and implement an automated crater-detection framework using knowledge discovery and data mining (KDD) process [3]. Training data is constructed using datum sampling method including SpreadSub sampling, Resample, and Cluster-based sampling. Mathematical morphology-based features are created for crater candidates. Then we use a greedy algorithm to search for the best features and the best classifiers. For the purpose of comparative study, we evaluate the performance of our framework at the same site centered on the Nanedi Valles on Mars used in [2]. The detection rate of F1 measure has been improved significantly from 0.613 to 0.772, where 0.613 is the reported detection rate in [2].

1.1 Related Work

Different methods on auto-detection of craters have been proposed in the following literature. Vinogradova et. al. developed a validation process applied to a feature-identification algorithm which is a typical template matching method [4]. Texture analysis is applied to crater detection in [5], which segments the images through the principal component analysis (PCA) of statistical texture measures. Honda et. al. developed a crater detection algorithm based on generalized Hough Transform to look for circular crater rims [6]. By computing the mapping between parameter space and image space, the algorithm can find the target objects within a certain class of shapes. The work most related to our framework is the crater-detection method proposed by Urbach and Stepinski in [2]. Similarly in [2], we identify a crater by two components: a bright region and a shadow region. Based on the preliminary work done in [2], we perform efficient KDD process [3] on feature creation and feature selection, training set sampling, and utilizing multiple classification methods to improve the detection accuracy.

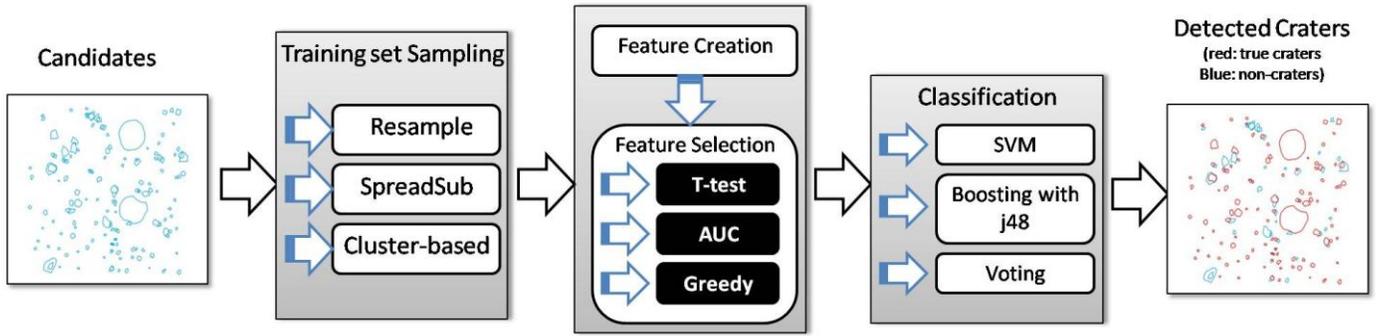


Fig. 1: Flow Chart of Main Procedure.

1.2 Data Description

The training set and testing set are constructed from a high resolution (12.5 m/pixel) panchromatic image h0905_0000 centering on Nanedi Valles on Mars [7]. The image is taken by a High Resolution Stereo Camera (HRSC) onboard the European Space Agency (ESA) spacecraft Mars Express. The training and test sites each contains 1700×1700 pixels, which covers an area of 451,562,500 m², respectively. These two sites are connected and located between longitudes of 47°.82E and 48°.11E, and the latitudes of -9°.52N and -10°.03N on Mars. We apply the image processing method discussed in [2] to extract crater candidates from the images. According to this method, highlight shapes and dark shapes are identified separately from images. Then bright shapes and dark shapes are combined. Those combinations of highlight and dark regions are promising locations for possible craters, which is then called a crater candidate. Note that not all crater candidates are correspondent to true craters. Efficient knowledge discovery algorithms have to be applied for crater detection.

1.3 Paper Organization

The paper is organized as follows. Section 2 discusses the approaches on dataset sampling, data preprocessing and classification. Section 3 presents the results of the experiments. Section 4 gives a discussion about the results. Section 5 concludes the paper.

2. THE FRAMEWORK OF CRATER DETECTION

To discuss our framework of crater detection, we focus on three major components of the procedure: training set sampling, feature creation and selection, and classification. Fig. 1 shows the main procedure of the framework. Candidates of craters are used as the input data in this procedure. Three sampling methods are applied on the training set: Resample [8], SpreadSub sampling [9] and Cluster-based sampling [10], a set of applied training sets are then built separately using the three sampling methods with different parameter settings. Those data sets are used as the training sets to build classification models and classification results from those training sets are compared and analyzed to identify the most efficient sampling method. Before the step of classification, two more procedures should be

done on both training and test sets: feature creation and selection, in which feature selection uses the following three methods: T-test, area under the curve (AUC) and a greedy algorithm. T-test and AUC are traditional statistical methods, while the greedy algorithm is a new feature selection method. Multiple sampling methods act across on multiple feature selection methods and different classification methods of SVM, Boosting with decision tree, and an ensemble voting classification method.

2.1 Training Set Creation

Our framework is based on supervised learning; constructing a training set is the first step. A domain expert manually labels true craters on the image, which is used as the ground truth for our case study. In our framework, crater candidates are automatically identified from the image. Crater candidates are labeled using the ground truth with the calculation done by the tool ArcGIS [11]. A method is developed to match the candidate shapes identified in an image automatically by our algorithm to the ground truth marked manually by the domain expert, with the purpose of labeling candidate shapes as true craters or non-craters. We deem the match criteria to be that the candidate shape in question must cover greater than 50% of a ground truth, and the ground truth must cover greater than 50% of the corresponding candidate shape. If the area-threshold 50% is met, then the candidate shape is considered to be a true crater.

2.2 Training Set Sampling

Since improved machine detection on craters is largely dependent upon improved training set creation, the original training set is sampled using several different methods. This sampling helps to avoid model overfitting to the training set, and it produces a more universally representative sample to be used on the testing set.

An explanation of the various sampling methods is as follows:

Method 1) Resample: Resample is a statistical processes which randomly picks instances from the original data sets. By observing the data set, the imbalance between crater candidates and non-crater candidates is obviously presented. In the training set, there are 479 true craters and 934 non-craters. Non-crater candidates have almost twice the number as the crater candidates.

Method 2) SpreadSub sampling method: Compared with Resample, SpreadSub sampling is more related to stratified sampling. This method pre-specifies the maximum difference between the most common class and the rarest class. By applying SpreadSub sample method, the proportion of the two classes can be brought under control. In our method, a different distribution spread is adjusted as the most important parameter to get the optimal result on classification.

Method 3) Cluster-based sampling method: This method uses clustering techniques in data mining to divide training datasets into clusters of “similar” objects. The similarity of two objects can be calculated as the distance between them. For instance, if each object is represented by a vector of N attributes, the distance between two objects can be a Euclidean distance. The purpose of clustering is to select the best representatives of each cluster for new training datasets. Intuitively, if the training datasets are well-clustered, i.e. the diversity of datasets is distributed properly in groups, and the best candidates from clusters are chosen, then better represented training datasets will be generated.

We use the simple KMeans clustering technique. The Cluster-based sampling algorithm is explained in Table 1. First the algorithm splits the original training set into crater set and non-crater set. Then it applies the Cluster-based sampling method on each of them. To make the new training sets be representative, the algorithm picks a number of n instances from each cluster. These instances compose the new training sets. For each new training set, F-measure is calculated. The training sets get the best F-measure value, which is then used to build the model and classify the test set.

F-measure is the external evaluation of the classification. In our training set, for example, there are 479 true craters and 934 non-craters. Thus if the classifier labels all candidates as false craters, the accuracy of the classification is 66.10%. However, successful detection of true craters is considered more significant than detection of non-craters. F-measure is an appropriate evaluation metric for crater detection. F-measure is defined as $\frac{2 \times TP}{2 \times TP + FP + FN}$, where True Positive (TP) is those

true craters which are also classified as craters by the classifier; False Positive (FP) is those non-crater candidates misclassified as craters by the classifier; False Negative (FN) is those true craters that are misclassified as non-craters by the classifier.

2.3 Feature Creation and Selection

In [2], 15 attributes are used to describe features. Table 3 discusses the meaning of all attributes. To further improve the training set we attempt to create 4 new attributes which would exhibit a strong correlation with the class attribute. Additionally, we seek to select a subset of existing old and new attributes which would exhibit a strong correlation with the class distribution.

2.3.1 Feature Creation

In order to help improve our classification accuracy, we use the Zonal Geometry function in ArcMap [11] to generate new

attributes for the training and test sets. This function calculates several geometric measurements for polygons, including area, perimeter, thickness, centroid coordinates, major and minor axes, and orientation. After generating these new attributes, they are tabulated and matched to the index of the features and Class fields from the training set and test set candidate files. Multiple combinations of the new attributes are created by dividing one attribute by another to produce a ratio, which is not specific to the size of the crater. Finally, all new attributes are evaluated by generating Receiver Operating Characteristic (ROC). The attributes with the highest Area Under the Curve (AUC) are selected and added to the training sets and candidate sets.

TABLE 1
PSUEDO CODE FOR CLUSTER-BASED SAMPLING ALGORITHM

Input: *TS: original training set*
n: number of training examples to be generated

Variable: *CS: crater set*
NS: non-crater set
c: clusters of objects
r: representatives instances
d_{CS}, d_{NS}: temporary dataset for craters and non-crater,
respectively
N_p: new dataset with p objects
f_p: F-measure value
k: numbers of clusters
p: numbers of objects picked from each clusters.

Output: *N: new training set*

Method:
1) *(CS, NS) = Split (TS)*
2) **FOR** *CS or NS do:*
 c = SimpleKMeans (CS or NS, k)
 FOR *each cluster in c do:*
 r_p = select a number of n instances in each cluster
 d_{CS or NS} = d_{CS or NS} ∪ r_p
 END FOR
 END FOR
3) *N_p = d_{CS} ∪ d_{NS}*
4) **FOR** *each N_p:*
 f_p = Calculate F-measure (N_p)
 END FOR
5) *N = N_p where f_p is the highest.*
6) *Output N*
END

2.3.2 Feature Selection

Among the 19 features used, some of them are not representative enough to differentiate the craters from the non-craters. Thus features are selected with 3 different methods in order to get the best F-measure on classification.

Method 1) Area Under the Curve. All attributes are evaluated by generating Receiver Operating Characteristic (ROC) curves. The area under the ROC curve of a feature, denoted as AUC (area under the curve), is a good indicator of the discriminating power of the feature. The features with the high area under curve (AUC) are selected. ROC curve areas are calculated for the combined list of original and newly-generated features, and then ranked from highest to lowest AUC. Features with larger values in AUC should provide better classification results; this has been confirmed by empirical results in [12]. Setting a series of minimum AUC thresholds, a different set of features are selected. For each test, the features with AUC below the

specified threshold are deleted and the modified training set is run on the test set.

Method 2) T-test. Independent sample T-tests are conducted to determine whether there is a significant difference in mean attribute values between the classes. Mean attribute values with the greatest difference are considered to be most useful in distinguishing between the two classes. Attributes that are not significant at a user-defined confidence levels are eliminated.

Base on AUC and T-test values, all attributes are sorted by the AUC and T-test from best to worst. Best 1 till n attributes will construct corresponding data sets which contains 1 to n attributes only. After applying i-attribute training set on different classifiers, we compare the F-measure of them and then get the best training set which contains all the attributes we want to select. We name this method Stepping Increasing method.

TABLE 2

PSUEDO CODE FOR GREEDY FEATURE SELECTION ALGORITHM

Input: $\{A\}$: whole attribute set
 n : total number of attribute

Variable: a : attribute
 TS : temporary data set, initially is \emptyset which contains no attribute.
 f : F-measure
 DS_i : i -attribute data set, which contains i attributes.

Output: FDS : final data set
 ff : final F-measure

Method:
1) **FOR** i in $[1, n]$:
 FOR each attribute $a_j \in \{A\}$ which has not been selected yet:
 $DS_j = \text{add } a_j \text{ to } DS_i$
 $f_j = \text{calculate F-measure } (DS_j)$
 END FOR
 IF DS_j has the best f_j :
 $DS_i = DS_j$
 $f_i = f_j$
 Classifier parameters setting
 IF $f_{i+1} < f_i$:
 $FDS = DS_i$
 $ff = f_i$
 BREAK
 END FOR
2) Output FDS, ff
END

Method 3) Greedy algorithm. A greedy approach is used for feature selection. When facing multiple choices, a greedy algorithm always follows the method which will bring the best result calculated within the current situation. In our greedy method, different features are treated as the different choices. Among all the features which have not been chosen so far, the one which brings with the best F-measure value is added to describe the candidate crater shapes using 10-fold cross validation on the training set. By applying this method, no more calculation needs to be done for measuring the quality of features. One feature will be chosen if and only if it shows the implication that implies the best result. The pseudo-code of the algorithm is shown in Table 2. During feature selection, since there are 19 attributes in the training set, one training set is split into 19 one-attribute attribute training set. Then, these 19 training sets are used to classify the test sets and compare the F-measure of them. Keep the attribute which gets the best

F-measure in the training set, and add one more attributes among the left 18 to the training set. Thus we get 17 2-attribute training sets. Use these 2-attribute training sets do the classifying and keep the new attribute which get the best F-measure with the first one. Based on this 2-attribute training set, and expand the attribute list until we cannot improve F-measure anymore.

In general, the only difference between the greedy algorithm with T-test/AUC and pure greedy algorithm is the criteria of feature selection. The first one considers the statistic value of each attribute when the later one only takes the classification results (with these attributes) into consideration.

2.3 Classification on Crater Detection

This section discusses the supervised learning algorithms which are tested and selected for crater detection. Various training sets are used as model input for supervised classification of both training and testing sites. During the sampling steps, around 150 training sets are created with 3 sampling methods and the best 3 training sets are applied where each of them represents the best training set of its sampling method. Around 500 experiments have been done to find the best F-measure of Greedy algorithm. For AUC and T-test feature selection method, each of them experiments 760 times for seeking the best F-measure. About another 500 experiments are done to seek the best classifier and paparameter. All experiments including classification, parameterization, model induction, and performance metrics are implemented using Java inside the 64 bit Microsoft Windows environment with an 8 GB RAM workstation. Python 2.6 serves as the primary tool for a test driver on running those experiments. Previous experiments and domain research direct our focus surrounding four major classification algorithms: SMO, LibSVM, AdaBoostM1/MultiBoostAB with J48 and Vote.

To get the best parameter setting of each particular classifier, a greedy search method is applied. Usually there are a couple of parameters for one classifier. To tune one parameter, the other parameters are kept unchanged. The tuned parameters will not be changed again once its local optimal value is identified.

2.3.1 Support Vector Machine: SMO & LibSVM

Support Vector Machine (SVM) [13] is a supervised classification method, constructing high-dimension hyperplane to divide data into two classes. Kernel mapping functions are applied to project and transform data into different dimensional spaces which are easier to separate data into classes. Kernel mapping functions are powerful to use to resolve the grouping problem with complicated boundaries. Different types of Kernel, however, have different dominance on different types of data, which directly decide the performance of the corresponding SVM algorithm. There are 4 Kernel Functions implemented by Weka 3.6 [9]: Linear: $u \cdot v$, Polynomial: $(\text{gamma} \cdot u \cdot v + \text{coef0})^{\text{degree}}$, Radial basis function (RBF): $\exp(-\text{gamma} \cdot |u-v|^2)$ and Sigmoid (feed-forward neural network): $\tanh(\text{gamma} \cdot u \cdot v + \text{coef0})$. In our paper, tests are conducted for all these 4 Kernel functions & using SMO [9] and

LibSVM [14].

2.3.2 Boosting Algorithm: AdaBoostM1/MultiBoostAB with j48

J48, an implementation of the C4.5 decision tree, applies a greedy method to induce the classification model. A decision tree is a simple and quick classification technique. .

AdaBoost [15] is short for the adaptive boosting machine learning algorithm. By repeating certain iterations, AdaBoost will construct the weight distribution data base for the weak classifier. By absorbing the weight knowledge from AdaBoost method, classifiers like decision trees will dramatically improve classification result. AdaBoostM1 is AdaBoosting with the M1 [16] method, which is designed to handle multi-class data sets. The method requires the basic classifier to be comparatively strong with an error rate of less than 50%.

MultiBoostAB [9] has been proved to significantly reduce errors when assisting the classification. This decision committee learning method combines AdaBoost with wagging (Wagging trains the entire training set but assign a stochastic weight to each instance.) [17] and the output of which is usually performed by majority voting. MultiboostAB shows the strengths of both AdaBoost and wagging. This combination allows MultiBoostAB to develop those two decision committees and keep both classified members. It offers great advantages and more efficiently works with decision trees.

2.3.3 Ensemble Method: Voting

Voting [9] is a meta-algorithm which decides the class label by combining the results of different classifiers. The combination rules are crucial. In our framework, we apply average of probabilities, majority voting, and maximum probability combination rules.

3. EXPERIMENTAL RESULTS

In this section we discuss the evaluation results of applying our framework to a real work case study on Mars crater detection. Evaluations are performed to gauge the comparative performance within and between, the sampling methods, feature creation and selection, classification and parameterization.

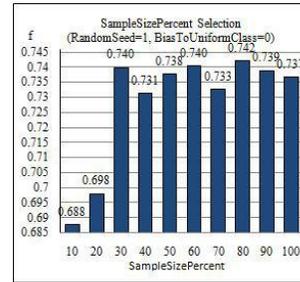
3.1 Sampling Methods

Method 1) Resample: The Resample routine is run in a series of experiments which used various adjustments in the aforementioned parameters to generate 32 different training sets. Fig. 2 shows the performance of the Resample method using different parameter settings. Based on these performance results, several “best” parameter combinations are also adjusted, and then tested. The best performance is achieved when sample size is 80% of the original training set.

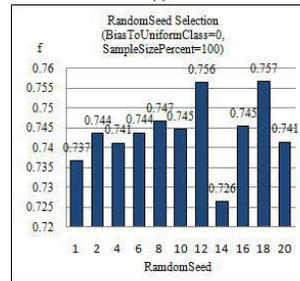
Method 2) SpreadSub sampling: Fig. 3 shows the performance of methods with different values on parameters DistributionSpread and MaxCount, which have a major effect on the result. The best subsample is found when the filter parameters are set as AdjustWeights=False,

DistributionSpread=1.5, MaxCount=0 and RandomSeed=5.

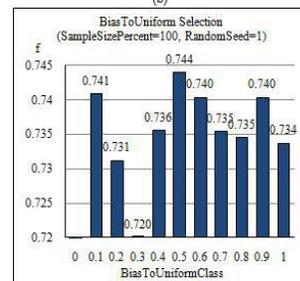
Method 3) Cluster-based sampling: There are two testing routines done. In the first routine, the P (numbers of objects) is adjusted while the K (numbers of clusters) is fixed. In second routine, the number of clusters is adjusted while P is fixed. P is tested in a range from 65% to 95%. To summarize, the best F-measure obtained by the Cluster-based sampling, specifically the K-Means sampling, is when 0.763 cluster numbers of craters is equal to 9, and when clusters number of non-craters are equals to 12, and 80% of the objects in the original training datasets are selected. The result is illustrated in Fig. 4.



In fig. (a), RandomSeed and BiasToUniformClass are settled. SampleSizePercent is enumerated and the F-measures are compared. With SampleSizePercent=80, the F-measure is 0.742.



In fig. (b), RandomSeed varies from 1 to 20 while the other two parameters are 0 and 100 respectively. The best F-measure is 0.757 when RandomSeed is 18. Besides, when RandomSeed equals to 12, F-measure is also good.



In fig. (c), BiasToUniformClass is selected by not changing SampleSizePercent and RandomSeed. The highest F-measure is 0.744, getting from the 0.5 BiasToUniformClass.

Fig. 2: Resample Method

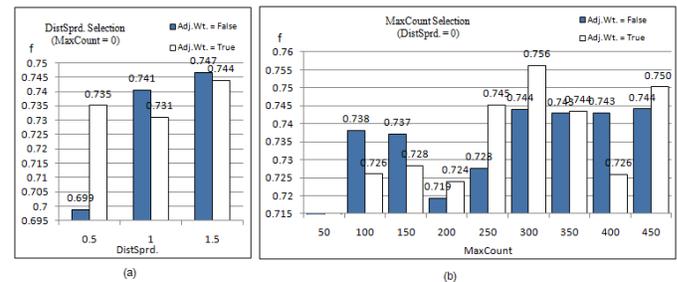


Fig. 3: SpreadSub Sampling Method (a) the parameterizing of distributionSpread. (b) MaxCount selection.

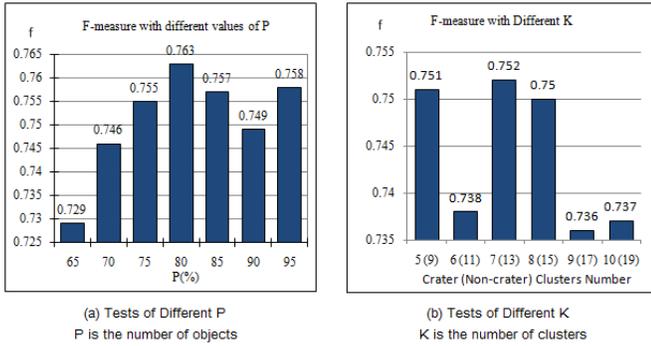


Fig. 4: Cluster-based Sample Parameter Setting. (a) shows the F-measure changing with different objects number. (b) F-measure with different cluster number.

3.2 Feature Creation and Selection

The newly created features are Ellipticity, PeriMinor, RecipOrient and MajorThick. The equations to create these new attributes are Major Axis/ Minor Axis, Perimeter/Minor Axis, $1/(\text{Orientation}+1)$ and Major Axis/Thickness separately. Thickness is defined as the radius of the largest circle that can be drawn within the polygon. This should be close to the diameter when a crater is nearly circular. For an elliptical polygon, the major axis is the longest diameter, and the minor axis is the shortest diameter. Orientation is the direction of the major axis in degrees counterclockwise from 0, up to a maximum of 180. ROC curve areas of the newly created attributes are the criteria used to justify the newly created attributes which are 0.678, 0.707, 0.620 and 0.594. Table 3 shows the description of all attributes.

Orientation is one of the attributes that initially fell below the 0.5 threshold, but the AUC is improved by calculating the reciprocal. The value is increased by 1 because some craters have an orientation of 0, which results in a null calculation when the value is inverted. PeriMinor has the highest AUC of the new attributes, ranking 6th overall when compared to the original attributes, followed by Ellipticity and RecipOrient. MajorThick is 10th in the list of all 19 attributes evaluated in the AUC feature selection algorithm.

Using the T-test attribute selection method, Hu5, Hu7, Hu6 and BrightLength shows weak evidence of a difference in mean values between the crater and non-crater groups. Table 4 shows the T-test values for all attributes. For greedy algorithm feature selection based on T-test value, the best result comes from the test set which does not contain Hu7.

Greedy algorithm is different from the AUC and T-test where it cannot give any particular standard to evaluate attributes. However, in greedy algorithm, F-measure is the sufficient criterion of feature selection. Using 10-fold cross validation, all three sampling method is applied to select the best one attribute and SMO with SpreadSub sampling method gives the best 1-attribute test set. The other best attributes selected are DarkElongation (F-measure is 0.579), Hu1 (F-measure is 0.620) and Hu3 (F-measure is 0.533) with Resample, SpreadSub sampling and Cluster-based sampling method separately. The best attribute with its sample method is thus kept using to search the best 2 to N attribute until the optimal

TABLE 3
ATTRIBUTES DESCRIPTION

Attribute Names	Description
BrightLength (BrtLen.)	$\sqrt{\text{size of the bright region}}$
Distance	$\sqrt{(x_a - x_b)^2 + (y_a - y_b)^2}$, where (x_b, y_b) is the center of the bright region, and (x_s, y_s) is the center of the shadow region.
AreaRatio	$\frac{\text{size of the bright region}}{\text{size of the shadow region}}$, if the first value < 1, then $\frac{\text{size of the shadow region}}{\text{size of the bright region}}$
BrightElongation (BrtElg.)	elongation of the bright region (seems to be the first of Hu's 7 moments invariants with some adjustment)
DarkElongation (DarkElg.)	elongation of the shadow region (seems to be the first of Hu's 7 moments invariants with some adjustment)
BothElongation (BothElg.)	elongation of the combined region (seems to be the first of Hu's 7 moments invariants with some adjustment), which is the crater candidate constructed by pairing bright and shadow regions
Dissimilarity (Dissmlrt.)	Euclidean distance between bright and shadow regions using Hu's 7 moments as the attribute vector
Circularity (Circrlrt.)	Compare the size of the bright and shadow regions with a perfect circle. This value is a ratio = (the area of the bright and shadow regions)/(the size of a perfect circle in a compatible size)
Hu1	Hu's Moment 1
Hu2	Hu's Moment 2
Hu3	Hu's Moment 3
Hu4	Hu's Moment 4
Hu5	Hu's Moment 5
Hu6	Hu's Moment 6
Hu7	Hu's Moment 7
Ellipticity	Major Axis/ Minor Axis
PeriMinor	Perimeter/Minor Axis
RecipOrient (RcpOrt.)	$1/(\text{Orientation}+1)$
MajorThick (MjorTh.)	Major Axis/Thickness

TABLE 4
ATTRIBUTES RANKED BY THREE FEATURE SELECTION METHODS

Ranked Attribute AUCs	T-test for Equality of Means		Attribute Step Adding		
	Area	Attributes	Equal Variances Assumed	Attributes	F-measure with attribute 1 to current
Hu1	0.813	Hu1	2.86E-81	Hu1	0.620
DarkElg.	0.760	DarkElg.	1.71E-59	Dissmlrt.	0.616
Hu4	0.759	Hu2	5.47E-41	Hu3	0.625
Hu3	0.752	Hu3	1.50E-36	Hu4	0.621
Hu2	0.740	PeriMinor	4.28E-34	Hu5	0.625
PeriMinor	0.702	Ellipticity	9.71E-24	Hu6	0.626
Ellipticity	0.674	Circrlrt.	1.98E-13	Hu7	0.626
RcpOrt.	0.623	Hu4	1.30E-12	Hu2	0.629
Circrlrt.	0.620	BothElg.	1.07E-09	DarkElg.	0.703
MjorTh.	0.596	AreaRatio	3.15E-06	BothElg.	0.738
Hu5	0.584	Distance	5.59E-06	BrtElg.	0.752
Hu6	0.575	MjorTh.	2.07E-05	AreaRatio	0.753
AreaRatio	0.556	BrtElg.	0.005	BrtLen.	0.754
BrtElg.	0.549	Dissmlrt.	0.015	Distance	0.763
Dissmlrt.	0.539	RcpOrt.	0.020	RcpOrt.	0.754
BrtLen.	0.538	BrtLen.	0.046	Ellipticity	0.755
Hu7	0.492	Hu6	0.068	MjorTh.	0.755
BothElg.	0.410	Hu5	0.208	PeriMinor	0.738
Distance	0.389	Hu7	0.907	Circrlrt.	0.739

F-measure is found. As shown in Table 4, the method can provide the best F-measure 0.763 when applying 14 attributes, Hu1, Dissimilarity, Hu3, Hu4, Hu5, Hu6, Hu7, Hu2, DarkElongation, BothElongation, BrightElongation, AreaRatio, BrightLength and Distance.

3.3 Classification and Parameterization

Both SMO and libSVM classifiers are used to test the Stepping Increasing AUC/T-test method with different sampling methods. According to the F-measure, SMO gives a better result for the most time. To avoid the impact from different numbers of attributes on the performance of kernels, all the Kernels are used to work on different n-attribute set. The best results are all given by the SMO Normalized Polynomial kernel. The comparison of different sampling method is given as the Table 5. Table 5 also contains the result of the greedy algorithm with the SMO and libSVM classifiers.

TABLE 5
BEST F-MEASURE OF DIFFERENT SAMPLING METHODS AND CLASSIFIERS

			Best F-measure	# of attribute used
SpreadSub	AUC	SMO	0.7498	18
		libSVM	0.6688	18
	T-test	SMO	0.7476	18
		libSVM	0.6650	16
	Greedy	SMO	0.7630	14
		libSVM	0.7359	14
Cluster-based	AUC	SMO	0.7434	18
		libSVM	0.6430	18
	T-test	SMO	0.7416	18
		libSVM	0.6288	10
Resample	AUC	SMO	0.7430	19
		libSVM	0.6424	18
	T-test	SMO	0.7473	18
		libSVM	0.6395	10

From Table 5, it is obvious that the greedy algorithm gives the highest F-measure using SMO. Comparing SMO and libSVM, SMO is generally better than libSVM. The difference of F-measure between these two classifiers is about 0.1. In greedy algorithm however, it is less than 0.03. It shows that greedy feature selection algorithm is more consistent with different classifiers. Since the second highest F-measure is given by SpreadSub sampling method (highlighted in bold red in

Table 5), the same training and test sets are also used to compare different classifiers.

After carefully adjusted the classifier parameters of SMO, the best F-measure among AUC and T-test data set is 0.7649. Using the same sampling method but with the first 18 AUC attributes on MultiBoostAB with j48, the F-measure is 0.7720, which is also the highest F-measure has ever gotten. The highest F-measure for the greedy algorithm is 0.763. If applying MultiBoostAB with j48 on best greedy algorithm data set, the F-measure is 0.764, which is also very close to the F-measure gotten from the SMO. It confirms that the different classifier does not impact the classification result on greedy algorithm data set. In the experiment of voting classifier, MultiBoostAB,

SMO Normalized Polynomial Kernel, LibSVM with linear, RBF, Sigmoid Kernels respectively are applied. The F-measure is 0.750 (sampling method: SpreadSub Sample, feature selection method: Stepping Increasing of AUC, 18 attributes).

4. DISCUSSION

Table 4 shows three feature selection methods. In all these three method, the best feature selected is Hu1. Hu moments are a set of image weighted average values which described rotation, scaling, and translation invariant. Hu1, like the moment of inertia among the centroid in physics, implies how much the distribution of the images gather in the horizontal and vertical axes. On the other hand, it implies if the image is easy to rotate or not. In our case, a crater is considered as a pair of bright and dark regions, thus the ability of rotation is important to decide if a candidate is a true crater or not. This means that the dark distribution of a true crater has a certain pattern. So it is not surprising that all three feature selection methods select Hu1 as the best feature. Also in Table 4, the ranks of some features in greedy feature selection algorithms are different from those in traditional static feature selection methods (AUC/T-test). DarkElongation, for example, in both AUC and T-test is at the second order in the table. In the greedy feature selection algorithm, DarkElongation is ranked in the middle part. It is because that, that DarkElongation describes the similar character of a crater with Hu1 moment. Consequently, though it can differentiate craters and non-craters well, it cannot enhance the performance of the classifiers further when Hu1 is already applied in the data sets.

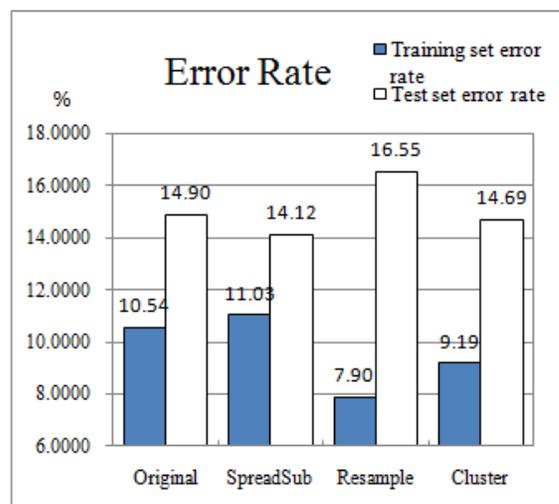


Fig. 5: Error Rate

In Fig. 5, error rate is compared among four sampling methods. The error rate is calculated by the following equation: $4 \times (FN+FP)/(TP+TN+FP+FN)$. It is quite interesting that SpreadSub sampling method gives the highest error rate in training set but lowest error rate in test set. Resample gives the totally reverse result. It shows that the model built according to the Resample training set is overfitted to a particular training set. If there is any noise in the training set, it will lead to an error

classified for the test set. According to Fig. 5, this problem does exist.

5. CONCLUSION

In the paper, we discuss a framework to improve the accuracy of Mars crater detection using KDD process. One 1700×1700 pixel tile is manually labeled and used as a training set while the same size of an adjacent image tile is used as a test tile. We made improvements on the data sets' sampling, feature creation and selection, and classification. The F-measure, as the main evaluation, is enhanced from 0.613 to 0.772. This result is obtained from the SpreadSub Sampling training set, with Stepping Increasing AUC feature selection method and the classifier is MultiBoostAB with j48. Fig. 6 shows the classified test set which got the highest F-measure.

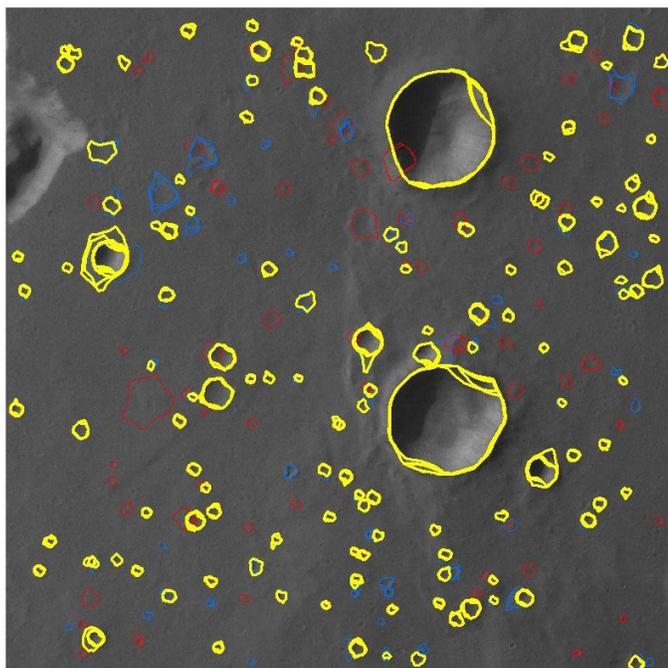


Fig. 6: Classification Result on the test set. Yellow features are TP, red features are FP, and blue features are FN.

The newly created feature selection method, greedy Stepping Increasing feature selection algorithm, shows its high accuracy and consistency.

ACKNOWLEDGMENT

The work is supported by UMass Boston Proposal Development Grant. The authors would like to thank informative discussion with Helenmary Hotz and Chirstopher A. Stillman of the department of Environmental, Earth and Ocean Sciences at UMass Boston.

REFERENCES

- [1] N.G. Barlow, "Crater size-frequency distributions and a revised Martian relative chronology," *Icarus*, vol. 75, p. 285-305, Aug. 1988.
- [2] E. R. Urbach, T. F. Stepinski, "Automatic Detection of Sub-Kilometer Craters in High Resolution Planetary Images," *Planetary and Space Science*, vol. 57, issue 7, pp. 880-887, Jun. 2009.
- [3] U. Fayyad, G. Piatetsky-Shapiro, P. Smyth "The KDD process for extracting useful knowledge from volumes of data," *Communications of the ACM*, vol. 39, issue 11, pp. 27-34, 1996.
- [4] T. Vinogradova, M. Burl, E. Mjolsness, "Training of a Crater Detection Algorithm for Mars Crater Imagery," *Aerospace Conference Proceedings*, 2002. IEEE, vol. 7, pp. 7-3201-7-3211.
- [5] T. Barata, E. I. Alves, J. Saraiva, and P. Pina, "Automatic recognition of impact craters on the surface of Mars," *Image Analysis and Recognition*, vol. 3212, *Lecture Notes in Computer Science*. New York: pringer-Verlag, 2004, pp. 489-496.
- [6] R. Honda, O. Konishi, R. Azuma, H. Yokogawa, S. Yamanaka and Y. Iijima "Data Mining System for Planetary Images - Crater Detection and Categorization", *Proceedings of the International Workshop on Machine Learning of Spatial Knowledge in conjunction with ICML*, 2000, pp. 103-108.
- [7] HRSC image viewer. (2008, March). Retrieved from Freie Universitat Berlin Planetology and Remote Sensing Insitute of Geosciences: http://hrscview.fu-berlin.de/cgi-bin/ion-p?page=entry_ion
- [8] P. Good, *Resampling Methods*. 3rd Ed. Boston: Birkhauser 2006.
- [9] "Weka 3 - Data Mining with Open Source Machine Learning Software in Java." (n.d.). Retrieved February 27, 2010, from University of Waikato, Computer Science Department Web site.: <http://www.cs.waikato.ac.nz/ml/weka/>
- [10] J. Kang, R.R. Kwang & H. -C. Kwon, "Using Cluster-Based Sampling to Select Initial Training Set for Active Learning in Text Classification," *PAKDD*, 2004, pp. 384-388.
- [11] ArcGISWebPage, 2008. <http://www.esri.com>.
- [12] S. Zhang, M. Maruf Hossain, Md. Rafiul Hassan, J. Bailey, K. Ramamohanarao, "Feature Weighted SVMs Using Receiver Operating Characteristics," *SDM*, 2009, pp 497-508.
- [13] D. Meyer, F. Leisch, K. Hornik, "The support vector machine under test," *Neurocomputing*, vol. 55, issue 1-2, pp. 169-186, Sep. 2003.
- [14] Weka LibSVM (WLSVM): Intergrating LibSVM into Weka Environment. Retrieved from Iowa State University, Department of Computer Science: <http://www.cs.iastate.edu/~yasser/wlsvm/>.
- [15] J. H. Friedman, "On Bias, Variance, 0/1-Loss, and the Curse-of-Dimensionality," *Data Mining Knowledge Discovery*, vol. 1, Number 1, pp. 55-77, Mar. 1997.
- [16] Y. Freund, R. E. Schapire, "Experiments with a New Boosting Algorithm," *ICML*, pp. 148-156, 1996.
- [17] L. Rokach, O. Z. Maimon, "Data Mining with Decision Trees: Theory and Applications," in *Series in Machine Perception and Artificial Intelligence*, vol. 69. World Scientific Publishing Company, 2008, pp. 107.