Bernoulli Trials Based Feature Selection for Crater Detection

Siyi Liu, Wei Ding, Joseph Paul Cohen, Dan Simovici
Department of Computer Science
The University of Massachusetts Boston
Boston, Massachusetts, 02125
{silu, ding, joecohen, dsim}@cs.umb.edu

Tomasz Stepinski
Department of Geography
University of Cincinnati
Cincinnati, Ohio, 45221
stepintz@uc.edu

ABSTRACT

Counting craters is a fundamental task of planetary science because it provides the only tool for measuring relative ages of planetary surfaces. However, advances in surveying craters present in data gathered by planetary probes have not kept up with advances in data collection. One challenge of auto-detecting craters in images is to identify an image's features that discriminate it between craters and other surface objects. The problem of optimal feature selection is known to be NP-hard and the search is computationally intractable. In this paper we propose a wrapper based randomized feature selection method to efficiently select relevant features for crater detection. We design and implement a dynamic programming algorithm to search for a relevant feature subset by removing irrelevant features and minimizing a cost objective function simultaneously. In order to only remove irrelevant features we use Bernoulli Trials to calculate the probability of such a case using the cost function. Our proposed algorithms are empirically evaluated on a large high-resolution Martian image exhibiting a heavily cratered Martian terrain characterized by heterogeneous surface morphology. The experimental results demonstrate that the proposed approach achieves a higher accuracy than other existing randomized approaches to a large extent with less runtime.

Categories and Subject Descriptors

I.5.2 [**Design Methodology**]: Classier design and evaluation; Feature evaluation and selection; Pattern analysis; I.5.4 [**Pattern Recognition**]: Applications – Astronomy

General Terms

Algorithms, Experimentation, Verification.

Keywords

feature selection, classification, planetary and space science, spatial data mining

1. INTRODUCTION

Impact craters are approximately circular depressions in the surface of a planet or other solid body in the Solar System, formed by the hyper-velocity impact of smaller bodies with the surface. Craters are among the most studied geomorphological features in the Solar System because they yield information about the past and present geological processes and provide the only tool for measuring relative ages of observed geologic formations [7]. It becomes extremely challenging to automatically count a very large number of small, sub-kilometer size craters in a deluge of high resolution planetary images. Identification of craters in remotely sensed images can be considered as a special case of object detection in images, which is an important task in computer vision exemplified by a popular task of face detection. However, craters have characteristics unlike most objects traditionally subjected to automated identification in images, because they are numerous, have large range of sizes, and they continuously merge into a background.

The problem of optimal feature selection is known to be NP-hard and the search is computationally intractable. We propose a wrapper based randomized feature selection method to efficiently select relevant features for crater detection. Wrapper approaches use the target induction algorithm to evaluate subset of features. We design and implement a dynamic programming algorithm to search for relevant feature subsets while removing irrelevant features and minimizing a cost objective function simultaneously.

2. RELATED WORK

In this paper, we use a wrapper method to select features. A wrapper method was coined by John, Kohavi, and Pfleger in 1994 [4], but the technique was used originally by Devijver and Kittler in 1982 [1]. A collection of subsets combined with a wrapped classifier creates a space that has been explored before with various techniques. Our wrapper based method fits into the supervised learning category [9, 5, 10].

3. BERNOULLI TRIALS BASED FEATURE SELECTION

We use a binary search approach to simplify the detection of relevant variables. Our proposed algorithm searches through the feature subset space in search of irrelevant features to discard, in order to achieve a more accurate classifier by maximizing relevant features in the final subset.

Our method first calculates the number of features that

Symbol	Description
n	Number of features in the feature space
r	Number of relevant features
k	Number of features will be removed in each iteration
$T(n,r,k)$	The total cost of feature selection algorithm based on n features, r relevant features, and will remove k features
$C(L,n-k)$	The cost of L learning algorithm with $n-k$ features
$p^+(n,r,k)$	The probability of successfully selecting k irrelevant features
$N^-(n,r,k)$	Number of trials before a success
X	i if the first success is on trial i
$\binom{1}{n}$	Number of combination select 1 from n
$E(X)$	Expectation of X
$p^-(n,r,k)$	The probability of successfully selecting k relevant features
$N^+(n,r,k)$	Number of trials before a failure

Table 1: Description of notation

should be removed in each iteration, and then randomly selects the calculated features for removing. Before removing the selected features, we individually evaluate the features and put the features that might be relevant back to the feature set. Our method then evaluates the new feature set and if the error rate is smaller, we remove the selected features. Otherwise, we randomly select and calculate features and do the individual evaluation again. At the same time, we count the number of trials. If the number of trials exceed the number of expected fail trials it is hard to remove more features. That is, the number of estimated relevant features is too low. If the number of trials exceeds the number of expected success trials it too easy to remove features. That is, the number of estimated relevant features is too high. We use this principle to seek the number of relevant features. The final subset is the set of relevant features. Table 1 summarizes the notation used by our method to be discussed in the following sections.

3.1 Cost Function

Given n features, some of the features are relevant, and some are not, but we do not know how many relevant features exist in n features. Our method tries to find the number of relevant features as well as relevant features, in the same time the algorithm reduces the total cost as small as possible. By using the Bernoulli trials we get the number of times of the learning algorithm will execute based on the number of features removed from the feature set. The cost of the learning algorithm is the time complexity of the learning algorithm which increases with the number of input features. Thus our method searches the best number of features to remove in each iteration [6].

$$T(n,r,k) = N^-(n,r,k) * C(L,n-k) + C(L,n-k) \quad (1)$$

$$p^+(n,r,k) = \frac{\binom{1}{n-r}}{\binom{1}{n}} * \frac{\binom{1}{n-r-1}}{\binom{1}{n-1}} * \ldots * \frac{\binom{1}{n-r-k+1}}{\binom{1}{n-k+1}} \quad (2)$$

$$p^+(n,r,k) = \prod_{i=0}^{k-1}(\frac{n-r-i}{n-i}) \quad (3)$$

$$P(S) = P^+, P(FS) = (1-p^+)p^+, \ldots, \ldots \quad (4)$$

$$E(X) = \sum_{i=0}^{\infty}(1-p^+)^i p^+ = \frac{1}{p^+} \quad (5)$$

$$N^-(n,r,k) = E(X) - 1 = \frac{1}{p^+} - 1 \quad (6)$$

In many datasets, especially most real-world datasets, we don't know the number of relevant features, r, so our algorithm must be able to search for the number of relevant features. In our search process, if the number of trials exceeds the expected number of consecutive failures, this indicates the estimated number of relevant features r has been chosen too low and should be increased. Similarly, if the number of trials exceeds the expected number of consecutive success, this indicates the estimated number of relevant features r has been chosen too high and should be decreased.

$$p^-(n,r,k) = \frac{\binom{1}{r}}{\binom{1}{n}} * \frac{\binom{1}{r-1}}{\binom{1}{n-1}} * \ldots * \frac{\binom{1}{r-k+1}}{\binom{1}{n-k+1}} \quad (7)$$

$$= \prod_{i=0}^{k-1}(\frac{r-i}{n-i}) \quad (8)$$

$$N^+(n,r,k) = \frac{1}{p^-} - 1 \quad (9)$$

3.2 Dynamic Programming for Irrelevant Feature Removal

Due to the nature of randomized search, we must consider the situation that it is very likely not all irrelevant features can be found in one iteration. We need to calculate the k $(0 < k \leq n-r)$ at each iteration step to minimize the cost. We get $n-k$ features. Then we need to calculate another k to remove from $n-k$ before we eventually approach r relevant features. Thus, we have the following total cost of function

$$T_{sum}(n,r) = min_k(T(n,r,k) + T_{sum}(n-k,r)) \quad (10)$$

The optimal value of k, given n and r, can be computed as

$$k_{opt}(n,r) = argmin_k(T(n,r,k) + T_{sum}(n-k,r)) \quad (11)$$

We use dynamic programming to solve this recursion. Algorithm 1 describes how to compute the k sequence and the cost sequence. The size of k sequence is $N-r$. The algorithm guarantees that the minimum cost can be achieved if k irrelevant features are removed at the correspondent state.

3.3 The Algorithm

The Bernoulli Trails based algorithm begins by computing tables for $k_{opt}(n,r)$ and $T_{sum}(n,r)$ for values of r between r_{max} and r_{min}, where r_{max} and r_{min} is the upper bound and the lower bound of number of relevant features, respectively. There are $N-r+1$ rows and $r_{max}-r_{min}+1$ columns in these tables.

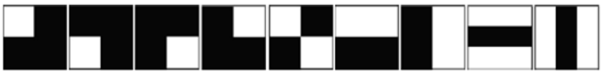

Figure 1: Nine types of masks used for Haar-like feature extraction on crater detection

The algorithm performs a binary search and begins with $r = \frac{r_{max}+r_{min}}{2}$. The algorithm chooses the number of features to remove at each step based on the current value of n. At each iteration, the algorithm selects $k_{opt}(n, r)$ input variables at random. Then the algorithm gets the error of each individual feature. If the error is smaller than a threshold, this indicates the feature is relevant and then we should put the feature back to the feature space and another feature is selected at random. After that the algorithm gets the new hypothesis from the $n-k$ features. If the error $e(h'')$ of new hypothesis h'' is less than the error $e(h)$ of the hypothesis h, then the selected k inputs are regarded as irrelevant and removed from the feature inputs. At the same time the success increases by 1 and the variable fail is set to 0. Otherwise, a new set of features are selected and the fail increases by 1 and success is set to 0.

The bound on the complexity of our algorithm is based on the complexity of the learning algorithm being used. If the given learning algorithm executes in time $O(n^2)$, then removing the $n-r$ irrelevant variables via randomized variable elimination also executes in time $O(n^2)$. This is a substantial improvement compared to the factor n of exhaustive search or more increase performance in removing inputs one at a time (we remove k features at a time).

4. EXPERIMENTAL RESULTS

We have selected a portion of the High Resolution Stereo Camera (HRSC) nadir panchromatic image h0905_0000 [3] to serve as the case study site for crater detection. As illustrated in Figure 2 the selected image has the resolution of 12.5 meters/pixel and the size of 13,500,000 (3,000 by 4,500) pixels. A domain expert manually marked 3,500 craters for this image to be used as the ground truth to which the results of auto-detection are compared. The image represents a significant challenge to automatic crater detection algorithms. We identify 12,542 crater candidates in the image using the pipeline explained by [8]. We extract 1,089 Haar-like features using masks similar to those in Figure 1. These masks are explained in [2].

We compare accuracy (accuracy $= \frac{TP+TN}{TP+TN+FP+FN}$) between the Bernoulli Trials based feature selection and other three algorithms of Randomized Variable Elimination (RVE) [6], exhaustive search (time limited) and random search (time limited). RVE [6] is a state-of-art randomized variable elimination method.

From table 2, we can see the runtime of Bernoulli Trials based is less than RVE and exhaustive search and random search, also the Iteration times (Iters) of Bernoulli Trials based method is the smallest one. This is reasonable, because we reduce the probability of relevant features to be removed. This will make the probability of $e(h'') - e(h) \leq 0$ high (Algorithm 2 Line 18), the n reduces more quickly,

Algorithm 1: Computing the k and Cost Sequence

```
Input: $L$, $N$, $r$
Output: $k$ sequence and the cost sequence
$T_{sum}[r+1..N] = 0$
$K_{opt}[r+1..N] = 0$
for $n = r+1 : N$ do
    $bestCost = \infty$
    for $k = 1 : i - r$ do
        $tempCost = T(n, r, k) +$
        if $tempCost < bestCost$ then
            $bestCost = tempCost$
            $bestK = k$
$T_{sum}[n] = bestCost$
$K_{opt}[n] = bestK$
```

Algorithm 2: Bernoulli Trials Based Feature Selection

```
   Input: $L$, $n$, $r_{max}$, $r_{min}$
   Output: Relevant feature subset and number of relevant
           features $r$
 1 Compute tables $T_{sum}(n, r)$ and $k_{opt}(n, r)$ for
   $rmin \leq r < rmax$
 2 $r = \frac{r_{max}+r_{min}}{2}$
 3 Success, fail = 0
 4 $h$=hypothesis produced by $L$ on $n$ inputs
 5 for $i$=1:$n$ do
 6     $h$=hypothesis produced by $L$ on $ith$ inputs
 7     $error(i) = e(h)$
 8 threshold = sum(error(i))/n
 9 while $rmin < rmax$ do
10     $k = K_{opt}(n, r)$
11     select $k$ features at random
12     for $j = 1 : k$ do
13         $h$ = hypothesis produced by $L$ on $jth$ inputs
14         while $e(h) < threshold$ do
15             Replace the $jth$ feature
16     Remove the selected $k$ features
17     $h''$=hypothesis produced by $L$ on $n-k$ features
18     if $e(h'') - e(h) \leq 0$ then
19         $n = n - k$
20         $h = h''$
21         success = success + 1
22         fail = 0
23     else
24         replace the selected $k$ features
25         fail = fail + 1
26         success = 0
27     if $fail \geq N^{-}(n, r, k)$ then
28         $r_{min} = r$
29         $r = \frac{r_{max}+r_{min}}{2}$
30         Success, fail = 0
31     else if $success \geq N^{+}(n, r, k)$ then
32         $r_{max} = r$
33         $r = \frac{r_{max}+r_{min}}{2}$
34         Success, fail = 0
```

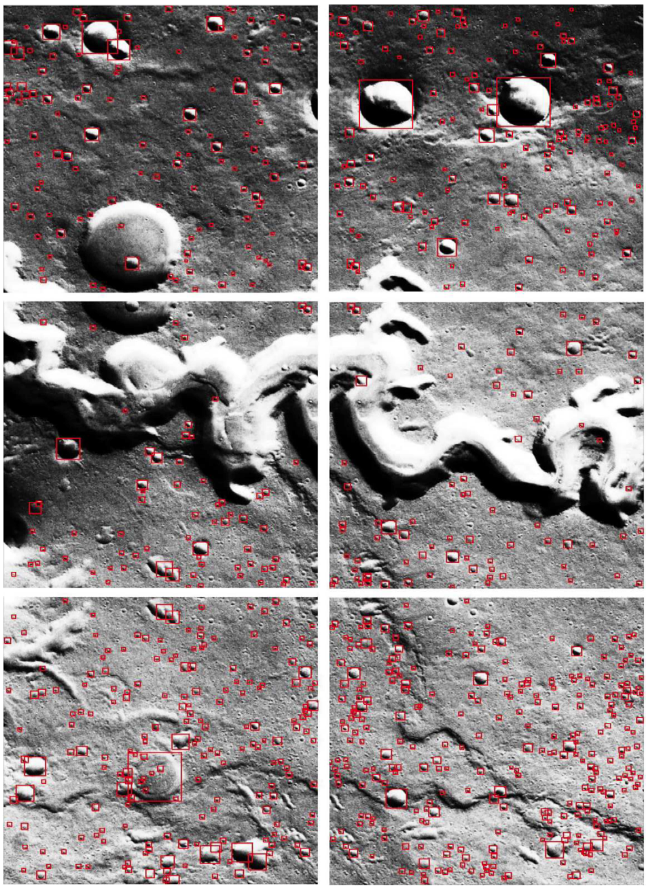

Figure 2: Detection result on the case study site. The craters marked by the red rectangles are craters detected by our algorithms which only tagets at craters that are between 16-pixel and 400-pixel in diameters.

thus k tend to more smaller. Therefore, our algorithm execute more quickly. The r of RVE is smaller than the r of our Bernoulli Trials based method with lower accuracy. It indicates that the RVE algorithms mistakenly remove more relevant features than our method.

5. CONCLUSIONS AND FUTURE WORK

The aim of this paper is to present a more efficient feature selection algorithm for the auto-detection of small craters in high resolution images of planetary surfaces. Effective and automatic crater detection from extremely large orbiter images is one of the most challenging problems in planetary science. The algorithm uses the Bernoulli Trials method to reduce the total cost of using a wrapper feature selection method. We have demonstrated that our method identifies craters with high accuracy while testing on an HRSC image of the Martian surface that presents a heterogeneous region of craters in various forms which are challenging for detection using regular algorithms. Our method can achieve accuracy nearly or more than 85%. We will explore the active learning and transfer learning on feature selection to further improve the accuracy.

Accuracy	RVE	Bernoulli	Exhaustive	Random
Top Region	83.12	90.19	84.26	85.36
Central Region	78.34	84.21	80.74	80.67
Bottom Region	81.56	86.33	82.89	81.67

Table 2: Performance results of the RVE, Bernoulli, Exhaustive and Random algorithms; Exhaustive and Random use limited time

	Runtime	r	Iterations
RVE	2503	84	1159
Bernoulli Trials based	1035	178	288
Exhaustive Search	18000	356	9153
Random Search	18000	451	8567

Table 3: Runtime comparison of RVE, Bernoulli, Exhaustive and Random algorithms; Exhaustive and Random use limited time

6. ACKNOWLEDGEMENTS

The work is partially supported by NASA grant NNX09AK86G.

7. REFERENCES

[1] P. A. Devijver and J. Kittler. *Pattern recognition: A statistical approach*. Prentice/Hall International, 1982.

[2] W. Ding, T. F. Stepinski, Y. Mu, L. Bandeira, R. Ricardo, Y. Wu, Z. Lu, T. Cao, and X. Wu. Sub-kilometer crater discovery with boosting and transfer learning. *ACM Transactions on Intelligent Systems and Technology*, 2(4), July 2011.

[3] HRSC. HRSC data browser. http://europlanet.dlr.de/node/index.php?id=209, 2010.

[4] G. H. John, R. Kohavi, and K. Pfleger. Irrelevant features and the subset selection problem. In *Proceedings of the eleventh international conference on machine learning*, volume 129, page 121âĂŞ129, 1994.

[5] C. Plesko, S. Brumby, E. Asphaug, D. Chamberlain, and T. Engel. Automatic crater counts on mars. volume 35, page 1935, Mar. 2004.

[6] D. J. Stracuzzi and P. E. Utgoff. Randomized variable elimination. *J. Mach. Learn. Res.*, 5:1331âĂŞ1362, Dec. 2004. ACM ID: 1044704.

[7] K. L. Tanaka. The stratigraphy of mars. http://adsabs.harvard.edu/abs/1986LPSC...17..139T, 1986.

[8] E. R. Urbach and T. F. Stepinski. Automatic detection of sub-km craters in high resolution planetary images. *Planetary and Space Science*, 57(7):880–887, 2009.

[9] T. Vinogradova, M. Burl, and E. Mjolsness. Training of a crater detection algorithm for mars crater imagery. In *2002 IEEE Aerospace Conference*, Mar. 2002.

[10] P. G. Wetzler, R. Honda, B. Enke, W. J. Merline, C. R. Chapman, and M. C. Burl. Learning to detect small impact craters. *Applications of Computer Vision and the IEEE Workshop on Motion and Video Computing, IEEE Workshop on*, 1:178–184, 2005.