CSC791 Spatial-Temporal Data Mining Project Report

Wei Fu, George Matthew, Qiang Zhang

***Abstract*—Crater detection has been one fundamental task in planetary science, which has drawn great attention from fields of data mining and image processing. In this project, typical spatial data mining algorithms for crater detection are discussed, implemented and analyzed. Based on the results of different algorithms, potential improvements are proposed.**

I. Introduction

Craters are key structures formed by the collisions of datameteorioids, asteroid or comet with planetary surface. To investigate past and future geological informaiton, craters detection methods or tools with high precision are demanded. Manual inspection of images is one mainly used method. However, due to the increasing huge amount of craters needed to be identified in very high resolution planetary images, this widely used method seems out-dated but automatic detection is the only practical solution to such tasks. Several reasons [1] can explain this:

- The lighting conditions result in different qualities of the images, which would affect the detecting accuracy.
- Some geographical features, like volcanos, have similar morphological characteristics as craters.
- Crater rims are frequently eroded due their formation millions of years ago.

In this project, we're trying to learn and implement algorithms that address all these issues. We choose [2] as a main reference to build a robust and practical crater detection framework. Based on this, a new detection algorithm will be designed to further improve the performance .

II. Problem definition

The automatic detection algorithms should solve the above challenges. Specifically, two problems are to address: (1) based on the candidates of craters, how to classify them accurately into crater and non-crater classes? (2) given a training dataset including crater and non-crater examples, how to make the algorithm applicable to detect the craters in other data sets.

III. State-of-the-art

Salamuniccar[3] provided an extensive review of previous work on crater detection algorithms, which can be divided into two categories: supervised and unsupervised-learning methods.

The unsupervised approcahes identify crater rims in an image as circular or elliptical features[4], [5], [6], [7], [8]. All of the methods fall in this category would use Hough Transform or similar image processing methods to improve the performance of crater edge detection Specifically, in [7], a new robust method is proposed to infer curvature estimation from noisy space data to compute a dense saliency map corresponding to the position and shape of the craters. The detected craters in the image are matched with the craters projected from a 3D model. Authors of [4] presented a methodology that brought number of techniques in the fields of image processing and pattern recognition, including candidate selection, template matching and final identification based on the probability to detect impact craters on images. Rie et al.[6] proposed to group original images according to spatial frequency patterns and both optimized parameters setts and noise reduction techniques used to identify candidate craters and false candidates are excluded using a self-organizing map approach.

Supervised methods are designed based on data mining techniques, where algorithms use the labeled data sets to train a learner and then identify craters on new data sets with this learner. Jinhao et al. [9] proposed to generate Gist feature from selected training samples and the crater detection is conducted using Gist feature vectors with random forests classification. Siyi et al.[10] used semi-supervised Active Class Selection algorithm to iteratively enrich an original small training set, without additional human labeling effort, to detect craters form a large volume of images. Ying et al. [11] proposed to integrate the Least Absolute Shrinkage and Selection Operator method with Bayesian network classifier for this task. The work we mainly referred designed a framework, which used a supervised boosting learning algorithms and transfer learning methods to classify crater candidates based on the texture features extracted from the original images.

IV. Algorithms

In this projects, we implemented tree algorithms, including Boost algorithm, Naive algorithm and Transfer Learning algorithm, proposed in [2].

The Boost algorithm generates a sequence of weak classifers $h_t(f)$ and combine them through a weighted approach to build a strong classifier $H(\hat{x})$:

$$H(\hat{x}) = \sum_{t=1}^{T} \alpha_t h_t f \tag{1}$$

, where T is the number of iterations, $t = 1, ..., T$, $\hat{x} =< f_1, ..., f_N >$ is the feature vector that describes a crater candidate; $f \in f_1, ..., f_N$ is the single feature used to construct a weak classifier $h_t(f)$ and α_t is the weight of hypothesis $h_t(f)$. The Boost algorithm iteratively selects one feature at a time and stops when reaching T iterations; each iteration the Boost algorithm will choose a best feature, which includes weak classifier learning , features selection and weight updating. More details are described in algorithm 1[2].

The Naive algorithm [2]is to reduce the time complexity required by Boost algorithm. This algorithm uses the same weak classifier learning step and selects the top T best features using the weighted error sum in the step of feature selection as a criterion without any further iterations on weight updating.

The Transfer Learning algorithm has the same three steps as Boosting algorithm, but the weight updating step is different. In Boosting algorithm, both training and testing data sets are drawn independently and identically from an underlying distribution and it is not expected to perform well if the testing data has a different distribution from the training data. The method used here is to build the strong classifier sequentially. Details are presented in *weight updating*step in Algorithm 3[2].

Algorithm 1 Boost: A boosting algorithm for feature selection and classification

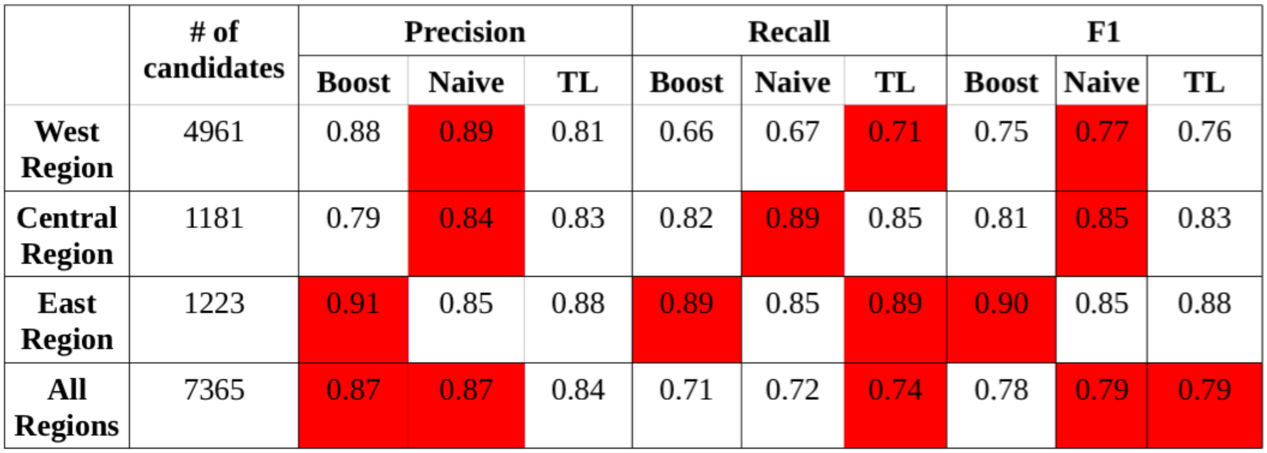
```
Require: .
    (1) Given crater candidates $(\widehat{x_1}, y_1), \dots, (\widehat{x_n}, y_n)$ where $y_i = 0, 1, i = 1, \dots, n$ for non-crater and crater examples respectively.
    (2) Initialize weights $w_i = \frac{1}{2m}$ if $y_i = 0$, $w_i = \frac{1}{2l}$ if $y_i = 1$, where m and l are the number of non-crater and crater examples respectively.
for $t = 1 \dots T$ do
    Normalize the weight, $w_{t,i} = \frac{w_{t,i}}{\sum_{j=1}^{n} w_{t,j}}, i = 1, \dots, n$
    so that $w_t$ is a probability distribution.
    Select the best weak classifier with respect to the weighted error
    $\epsilon_t = argmin_{f,p,\theta} \sum_i w_i |h(\widehat{x_i}, f, p, \theta) - y_i|$,
    For each feature, $f$, train a classifier $h$, which is restricted to using a single feature.
    Define $h_t(\widehat{x}) = h(\widehat{x}, f_t, p_t, \theta_t)$, where $\widehat{x}, f_t, p_t, \theta_t$ are the minimizers of $\epsilon_t$.
    Update the weights:
    $w_{t+1,i} = w_{t,i} \beta_t^{1-e_i}, i = 1, \dots, n$
    where $e_i = 0$ if a crater candidate $x_i$ is classified correctly, $e_i = 1$ otherwise, and $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$.
    The final strong classifier is:
    $h(\widehat{x}) = \begin{cases} 1 & \sum_{t=1}^{T} \alpha_t h_t(\widehat{x}) \geq \mu \sum_{t=1}^{T} \alpha_t \\ 0 & otherwise \end{cases}$
    where $\alpha_t = ln \frac{1}{\beta_t}$ and $\mu$ is a user-defined threshold.
end for
```

Algorithm 2 Naive: A naive greedy algorithm for feature selection and classification

```
Require: .
    (1) Given crater candidates $(\widehat{x_1}, y_1), \dots, (\widehat{x_n}, y_n)$ where $y_i = 0, 1, i = 1, \dots, n$ for non-crater and crater examples respectively.
    (2) Initialize weights $w_i = \frac{1}{2m}$ if $y_i = 0$, $w_i = \frac{1}{2l}$ if $y_i = 1$, where m and l are the number of non-crater and crater examples respectively.
Normalize the weight, $w_{t,i} = \frac{w_{t,i}}{\sum_{j=1}^{n} w_{t,j}}, i = 1, \dots, n$
so that $w_t$ is a probability distribution.
Select the best t $(t = 1, \dots, T)$ weak classifiers with respect to the weighted error
$\epsilon_t = \sum_i w_i |h(\widehat{x_i}, f, p, \theta) - y_i|$,
For each feature, $f$, train a classifier $h$, which is restricted to using a single feature.
Define $h_t(\widehat{x}) = h(\widehat{x}, f_t, p_t, \theta_t)$ where $\widehat{x}, f_t, p_t, \theta_t$ are the minimizers of $\epsilon_t$, and $t = 1, \dots, T$.
$\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$.
The final strong classifier is:
$h(\widehat{x}) = \begin{cases} 1 & \sum_{t=1}^{T} \alpha_t h_t(\widehat{x}) \geq \mu \sum_{t=1}^{T} \alpha_t \\ 0 & otherwise \end{cases}$
where $\alpha_t = ln \frac{1}{\beta_t}$ and $\mu$ is a user-defined threshold.
```

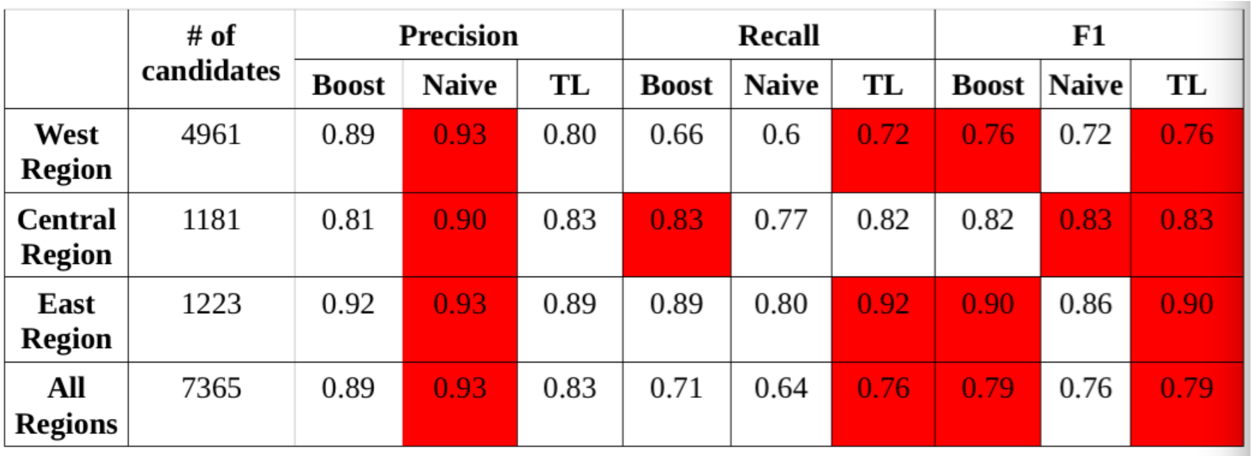
Algorithm 3 TL: A boosting algorithm using transfer learning for feature selection and classification

```
Require: .
    (1) Given a training set that includes crater candidates $(\widehat{x_1}, y_1), \dots, (\widehat{x_{n_d}}, y_{n_d}), (\widehat{x_{n_d+1}}, y_{n_d+1}), \dots, (\widehat{x_{n_d+n_s}}, y_{n_d+n_s})$, where $y_i = 0, 1, i = 1, \dots, n_d, n_d + 1, \dots, n_d + n_s$ for non-crater and crater examples respectively.
    This training set has $n_d$ diff-distribution examples $(1, \dots, n_d)$ and $n_s$ same-distribution examples $(n_d + 1, \dots, n_d + n_s)$, and $n = n_d + n_s$.
    (2) Initialize weights $w_i = \frac{1}{2m}$ if $y_i = 0$, $w_i = \frac{1}{2l}$ if $y_i = 1$, where m and l are the number of non-crater and crater examples respectively
for $t = 1 \dots T$ do
    Normalize the weight, $w_{t,i} = \frac{w_{t,i}}{\sum_{j=1}^{n} w_{t,j}}, i = 1, \dots, n$
    so that $w_t$ is a probability distribution.
    Select the best weak classifier with respect to the weighted error
    $\epsilon_t = argmin_{f,p,\theta} \sum_i w_i |h(\widehat{x_i}, f, p, \theta) - y_i|$, $i = n_d + 1, \dots, n_d + n_s$
    For each feature, $f$, train a classifier $h$ in same-distribution data, which is restricted to using a single feature.
    Define $h_t(\widehat{x}) = h(\widehat{x}, f_t, p_t, \theta_t)$ where $\widehat{x}, f_t, p_t, \theta_t$ are the minimizers of $\epsilon_t$
    Update the weights:
    $w_{t+1,i} = w_{t,i} \beta_t^{-e_i}$, if $n_d + 1 \leq i \leq n_d + n_s$ (increase the weights for the same-distribution)
    $w_{t+1,i} = w_{t,i} \beta^{e_i}$, if $1 \leq i \leq n_d$ (decrease the weights for the diff-distribution)
    where $e_i = 0$ if example $x_i$ is classified correctly, $e_i = 1$ otherwise, and $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$, $\beta = \frac{1}{1+\sqrt{2ln\frac{n}{T}}}$
    The final strong classifier is:
    $h(\widehat{x}) = \begin{cases} 1 & \sum_{t=\lceil \frac{T}{2} \rceil}^{T} \alpha_t h_t(\widehat{x}) \geq \mu \sum_{t=\lceil \frac{T}{2} \rceil}^{T} \alpha_t \\ 0 & otherwise \end{cases}$
    where $\alpha_t = ln \frac{1}{\beta_t}$ and $\mu$ is a user-defined threshold.
end for
```



	# of candidates	Precision			Recall			F1		
		Boost	Naive	TL	Boost	Naive	TL	Boost	Naive	TL
West Region	4961	0.88	0.89	0.81	0.66	0.67	0.71	0.75	0.77	0.76
Central Region	1181	0.79	0.84	0.83	0.82	0.89	0.85	0.81	0.85	0.83
East Region	1223	0.91	0.85	0.88	0.89	0.85	0.89	0.90	0.85	0.88
All Regions	7365	0.87	0.87	0.84	0.71	0.72	0.74	0.78	0.79	0.79

Fig. 1. Performance of Boost, Naive, and TL algorithms;T= 50; Boost : mu =0.475; ForGreedy: mu = 0.325; Transfer: mu = 0.5.



	# of candidates	Precision			Recall			F1		
		Boost	Naive	TL	Boost	Naive	TL	Boost	Naive	TL
West Region	4961	0.89	0.93	0.80	0.66	0.6	0.72	0.76	0.72	0.76
Central Region	1181	0.81	0.90	0.83	0.83	0.77	0.82	0.82	0.83	0.83
East Region	1223	0.92	0.93	0.89	0.89	0.80	0.92	0.90	0.86	0.90
All Regions	7365	0.89	0.93	0.83	0.71	0.64	0.76	0.79	0.76	0.79

Fig. 2. Performance of Boost, Naive, and TL algorithms;T= 150; Boost : mu =0.475; ForGreedy: mu = 0.325; Transfer: mu = 0.5.

V. Experimental results

A. Dataset description

The dataset used in this project is from the same source as [2], which is a portion of the Hight Resolution Stereo Camera nadir panchromatic image h0905. The selected image has the resolution of 12.5 meters/pixel and the size of 3,000 by 4,500 pixels. The image is divided into west region, central region and east region, which include 4961,1181 and 1223 crater candidates, respectively. For some unknown reason, these number are different from those in [2] even though we got this data set from their website.

B. Analysis

The results generated by our implementation is listed in Fig.1 and Fig. 2. We can see that the best precision is produced by the naive algorithm, the best recall is produced by the transfer learning algorithm and the best F1 score is predicted by the transfer learning algorithm. We can also see that the recall of the naive algorithm reduces when we increase the number of features but the transfer learning algorithm improves when we use more features. The results shown here are different from [2] in terms of ranks of these three algorithms for different goals. One possible reason is that the dataset we used offered by Dr.Ding is different from the ones they used in the paper.

VI. Summary

In the first phase of our projects, we did literature review, found related data sets and implemented three algorithms in the target paper. The results we got are different from the original paper. We would verify our results by another version of implementation. In next phase, we plan to implement new algorithm to improve the detection performance.

References

[1] Jung Rack Kim, Jan-Peter Muller, Stephan van Gasselt, Jeremy G Morley, and Gerhard Neukum. Automated crater detection, a new tool for mars cartography and chronology. *Photogrammetric Engineering & Remote Sensing*, 71(10):1205–1217, 2005.

[2] Wei Ding, Tomasz F. Stepinski, Yang Mu, Lourenco Bandeira, Ricardo Ricardo, Youxi Wu, Zhenyu Lu, Tianyu Cao, and Xindong Wu. Sub-kilometer crater discovery with boosting and transfer learning. *ACM Trans. Intell. Syst. Technol.*, 2(4):39:1–39:22, July 2011.

[3] Goran Salamunićcar and Sven Lončarić. Open framework for objective evaluation of crater detection algorithms with first test-field subsystem based on mola data. *Advances in Space Research*, 42(1):6–19, 2008.

[4] Lourenço Bandeira, Jose Saraiva, and Pedro Pina. Impact crater recognition on mars based on a probability volume created by template matching. *Geoscience and Remote Sensing, IEEE Transactions on*, 45(12):4008–4015, 2007.

[5] Yang Cheng, Andrew E Johnson, Larry H Matthies, and Clark F Olson. Optical landmark detection for spacecraft navigation. *Advances in the Astronautical Sciences*, 114(3):1785–1803, 2003.

[6] Rie Honda, Yuichi Iijima, and Osamu Konishi. Mining of topographic feature from heterogeneous imagery and its application to lunar craters. In *Progress in Discovery Science*, pages 395–407. Springer, 2002.

[7] Bertrand Leroy, Gérard Medioni, E Johnson, and Larry Matthies. Crater detection for autonomous landing on asteroids. *Image and Vision Computing*, 19(11):787–792, 2001.

[8] M.J. Galloway, G.K. Benedix, P.A. Bland, J. Paxman, M.C. Towner, and T. Tan. Automated crater detection and counting using the hough transform. In *Image Processing (ICIP), 2014 IEEE International Conference on*, pages 1579–1583, Oct 2014.

[9] Jihao Yin, Hui Li, and Xiuping Jia. Crater detection based on gist features. *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of*, 8(1):23–29, Jan 2015.

[10] Siyi Liu, Wei Ding, and T.F. Stepinski. Semi-supervised based active class selection for automatic identification of sub-kilometer craters. In *Image and Signal Processing and Analysis (ISPA), 2011 7th International Symposium on*, pages 615–620, Sept 2011.

[11] Ying Wang, Wei Ding, Kui Yu, Hao Wang, and Xindong Wu. Crater detection using bayesian classifiers and lasso. In *Conference Anthology, IEEE*, pages 1–5, Jan 2013.