# Observing Dark Worlds (Final Report)

Bingrui Joel Li (05100079)

***Abstract* — Dark matter is hypothesized to account for a large proportion of the universe's total mass. It does not emit or absorb light, making detection of this elusive matter really hard. However, this highly massive structure creates extensive space-time warping which alters the path of light from background galaxies. As a result, the ellipticity of galaxies that is observed on earth is altered and this provides the opportunity to deduce the position of dark matter halos from the shapes of these galaxies. In this work, I aim to use machine learning algorithm to tackle this challenging problem.**

## I. Introduction

Dark matter is postulated to constitute 84% of the matter in the universe and 23% of the mass-energy[1]. Einstein's general theory of relativity predicts that light passing by a highly massive object would bend and deviate from its original path because of warped spacetime, creating an effect known as gravitational lensing[2]. The result of this effect is galaxies in the background have their ellipticity altered and this provides an opportunity to detect dark matter halos. However, even without the presence of dark matter, galaxies are naturally elliptical and the challenge is to separate the true ellipticity of a galaxy from those resulting from the gravitational effect that dark matters exerts on the light path.

## II. Experimental Data

### A. Dataset

The dataset used in this work is obtained from Kaggle[3], which is a platform that hosts predictive modeling competitions. One hosted competition is to predict the positions of dark matter halos from a sky filled with galaxies. The data consists of a training set of 300 skies with the positions and ellipticity (e1, e2) of each of the 300-700 galaxies in the skies. e1 describes the elongation of the galaxy in the x, y direction while e2 is the elongation in the 45 degrees angle direction, as shown in Fig. 1[3]. The training data also lists the positions of dark matter halos in each of the 300 skies. The skies have been split into 3 equal sets of 100 skies with each set having 1, 2 or 3 skies respectively. An example of a sky is shown in Fig. 2 where the center of the blue spots show the position of dark matter halos and white spots are the galaxies[4]. The final test dataset is the same as the training dataset except that the positions and number of halos in each of the test skies are not listed. These have to be predicted with the machine learning algorithm.

The training and test data were separated into 3 different groups, which have 1, 2 and 3 halos respectively. Skies 1-100 have 1 halo, skies 101-200 have 2 halos and skies 201-300 have 3 halos.

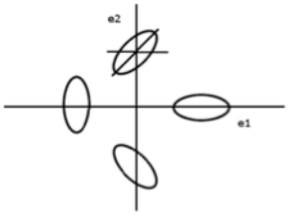


Fig. 1 Definition of $e_1$ and $e_2$ to describe galaxy ellipticity

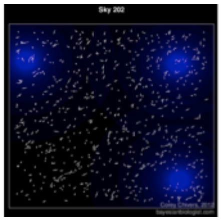


Fig. 2 An example of sky with three dark matter halos

### B. Data Processing

The dimensions of each sky is 0 to 4200 units in both the x and y direction and has been split into pixels with a size of 100 by 100 units. This means that we have a total of 42 x 42 = 1764 pixels for each sky. Each pixel is a training example with features describing the ellipticity and distance of each galaxy from this pixel. The class value of each pixel can take on a value of 1 (indicating the presence) or 0 (absence of a dark matter halo).

As can be seen, this is a binary classification problem with an unbalanced dataset. In subsequent sections, I will talk about some work on handling this unbalanced dataset to improve the classification.

### C. Evaluation Metric

The evaluation metric used to evaluate the algorithm's ability in identifying dark matter halos is identical to the one described in Kaggle. The metric is basically $m = F/1000 + G$ where F is the average radial distance from the user estimate to the true position of the halo and G is represented by

$$G = \sqrt{\left(\frac{1}{N}\sum_{i=1}^{N}\cos(\phi_i)\right)^2 + \left(\frac{1}{N}\sum_{j=1}^{N}\sin(\phi_j)\right)^2}$$

where $\phi$ is the angle between the predicted position with respect to the centre of the true halo position.

## III. Experimental Approach

### A1. Machine Learning Algorithm

The machine learning algorithm tried was the support vector machine (SVM), implemented with the libsvm-3.13 software, a library for support vector machines (SVM) obtained from http://www.csie.ntu.edu.tw/~cjlin/libsvm/.

A radial basis function (RBF) kernel was used for the SVM with different weights applied to the 2 different classes of 0 and 1. This is done to counter the effect of an unbalanced dataset with very few ‘1’s. The algorithm was initialized with a cost C of 10, weight w1 of 1e6 and gamma is set at default value of 1/(number of features). The absolute value of the decision value obtained from the svmtrain function in libsvm is used to assess the confidence of the algorithm’s classification. The higher the decision value, the more confident the algorithm’s classification is.

A potential issue that can occur is regions near the halo center will also have high decision values but do not necessarily correspond to halo positions. In order to overcome this, the algorithm has been specified to ignore pixels that are within a certain distance to the pixel that has the highest probability of being a halo center. The distance criteria used initially is one-tenth of the length of the sky image.

### *A2. Features*

I designed 8 features to describe the ellipticity and distance of each galaxy in a sky to a pixel. The features are the tangential ellipticity, $e_{tan}$, of a galaxy as defined by the following equation described in Kaggle, [3].

$$e_{\tan} = -(e_1 \cos(2\varphi) + e_2 \sin(2\varphi))$$

Features 1, 3, 5 and 7 are based on this tangential ellipticity value but differ in the number of galaxies included in the calculation. Galaxies within a certain distance from a pixel are included in the calculation and this distance differs among features 1, 3, 5 and 7. Features 1, 3, 5 and 7 have distance values of 4200, 3150, 2100 and 1050 units respectively. The rationale behind this is the effect that a halo has on a galaxy drops off as 1/r, where r is the distance between a galaxy and a halo. Understanding that galaxies closer to a halo will experience a larger effect than galaxies further from it, we form features 1, 3, 5 and 7 to consider galaxies at various distances to a pixel. And each of these features value is the mean ellipticity of the considered galaxies. This is done to take into consideration that each feature includes different number of galaxies in its calculation.

Features 2, 4, 6 and 8 are based on the product of r and $e_{tan}$ and this makes physical sense since $e_{tan}$ varies with 1/r. Again, the features represent the mean values of all considered galaxies at distances 4200, 3150, 2100 and 1050 units respectively.

Since every sky has a different number of halos and galaxies, it makes sense to normalize the feature values for each sky and each feature. Normalizing the features is also a requirement for the radial basis function of the SVM. This is accomplished by finding the z-score of each feature which is basically subtracting the mean from the actual value and dividing over the standard deviation.

## *B. Diagnostics and Optimization*

### *B1. Learning Curve*

Due to the intensive computation required for the large number of training examples (300 skies * 1849 pixels/per sky = 554700 pixels), I am choosing a selected group of skies to begin training. The training examples have equal representation from the three different sky types (one halo, two halo and three halos). For a start, I chose features 1, 2, 7 and 8 to train the algorithm since they represent distance values at opposite ends and present as a good initial choice. The SVM parameters used are stated in section A.

To determine if the algorithm has a high bias or high variance, I plotted the results from the experiment as a learning curve in Fig. 3. This curve shows the training and cross-validation (CV) metrics against number of training examples. The benchmark on the graph is the test metric value for one of the publicly available algorithm, single halo maximum likelihood. The desired performance is the test metric obtained from the best performing publicly available algorithm, Lenstool Maximum Likelihood. As can be seen, the training metric slightly increases as the number of training skies increases while the CV Metric stays high. Both training and CV metrics have almost similar values. This learning curve exhibits the characteristics of an under-fitted algorithm with high bias. One way to overcome this is to introduce more features.

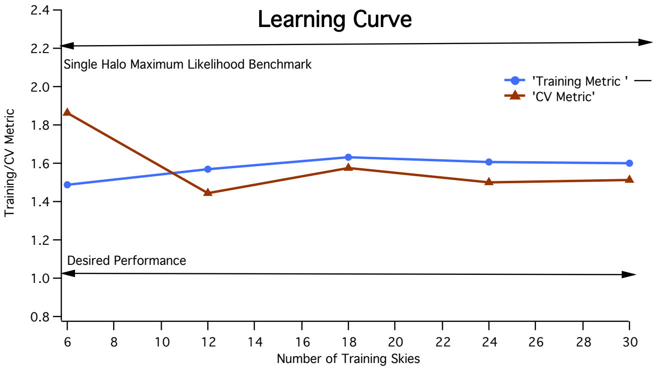


Fig. 3 Learning Curve

### *B2. Training Set Selection*

In order to do proper feature selection, it is important to choose a good training set first.

In skies with multiple halos, it is reasonable to expect there can be more fluctuations/variations because each galaxy’s ellipticity is a combination of random noise and effects from multiple halos. This hints towards the possibility that training on all 3 groups of skies may not necessarily give the best CV/test metric value. I would like to investigate if this is the case by carrying out an experiment to identify which training types or groups of types would be most suited as the training set. Types here refer to the skies with a certain number of halos. The three types are one halo, two halos and threes halos' skies.

In training groups with one type, I selected training skies 1-10 (one halo) to be group [1], skies 101-110 (2 halos) to be group [2] and skies 201-210 (3 halos) to be group [3]. For training groups with 2 types, I chose skies 1-5, 101-105 as group [1, 2], skies 1-5, 201-205 as group [1, 3] and skies 101-105, 201-205 as group [2, 3]. For the combined group [1, 2, 3], I selected skies 1-4, 101-104 and 201-204. The CV skies for all runs are skies 91-100 (one halo), 191-200 (two halo) and 291-300 (three halo). The selection is done in this manner so that the total number of training skies used in each run is roughly the same.

The result of this experiment is shown in Fig. 4, which demonstrates how CV metric changes with feature used. "CV Metric [1]" refers to CV metric obtained from training on skies with 1 halo and "CV Metric [1,2,3]" refers to CV metric obtained from training on skies with 1,2 and 3 halos.

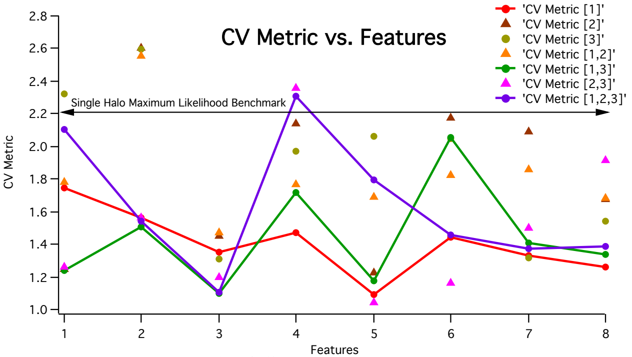
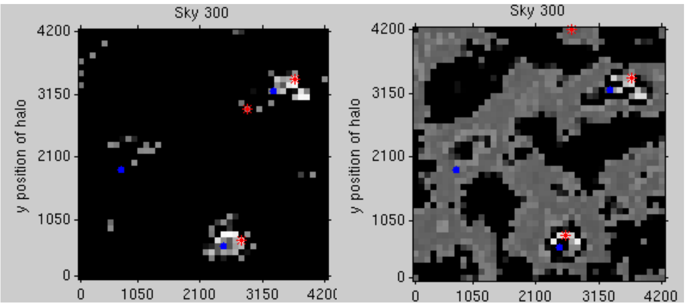


Fig. 4 CV Metric vs. Features

As can be seen from Fig. 4, training group [1] (red line) and group [1, 3] (green line) generally outperforms the other groups across features. Group [1, 2, 3] (purple line) generally performed worse than groups [1] and [1, 3] across the features.

To understand why this is the case, I created heat maps of decision values, Fig. 5, obtained from training with groups [1], [1,3] and [1,2,3]. Henceforth, I will call these maps, 'decision maps'. The bright regions correspond to areas that the algorithm has identified as potential halo locations and the brightness has been scaled for each sky. The brighter the spot, the higher the possibility of it being a halo compared to other regions in the sky. Blue spots correspond to actual halo positions and red spots are the predicted halo positions. The left and right decision maps on the first line in Fig. 5 represent the cases when the algorithm is trained on groups [1] and [1,3] respectively. And the figure on the next line represents the decision map obtained by the algorithm when trained on group [1,2,3].



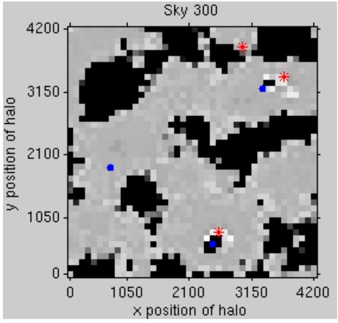


Fig.5 Decision maps obtained from training with groups [1], [1,3] and [1,2,3]

The decision maps obtained from training on skies with 1 halo, group [1], have high points in small and tight regions, shown by the left figure on the first line in Fig.5. When the algorithm is trained on skies with increasing number of halos [1,3] and [1,2,3], the decision values are high in increasing areas of the map as shown in Fig. 5. This can be explained by the multiple halo contributions on the ellipticity of a galaxy, which generally lowers a galaxy's ellipticity, making it difficult to distinguish from random noise. Because of this, it is challenging to train an algorithm using skies with multiple halos to accurately pinpoint halo positions. And training results on skies with 1 halo have resulted in better results, Fig. 4. Henceforth, group [1] will be chosen as the training group for subsequent optimization.

### *B3. Feature Selection*

To find the best feature subset, I performed a type of wrapper model feature selection algorithm known as forward search on the 8 features.

My forward search reveals that for group [1], the best feature subset is (5) with a CV metric of 0.9906.

In order to figure out if the current algorithm is under-fitting or over-fitting, a new learning curve was created, Fig. 6. Training and CV metric has both improved from before. This new learning curve exhibits the characteristic of an over-fitted algorithm with high variance. In order to improve this, the SVM parameters will be tuned.

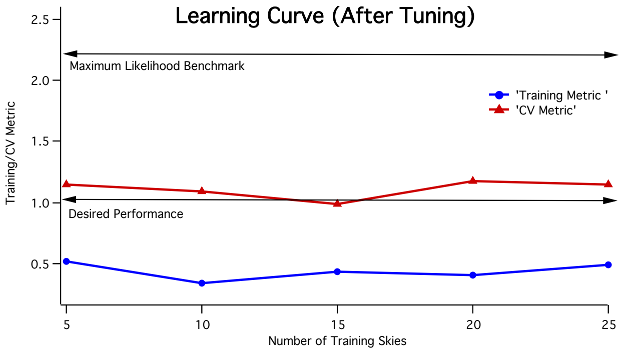


Fig. 6 Learning curve after training set and feature selection

### *B4. SVM algorithm parameters*

To tune my SVM, I varied parameters - gamma, C and w1. The CV Metric as a function of gamma, C and w1 are shown in Figs. 7, 8 and 9 respectively.

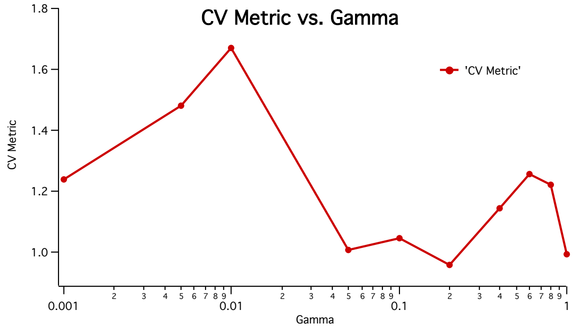


Fig. 7 CV Metric vs. Gamma

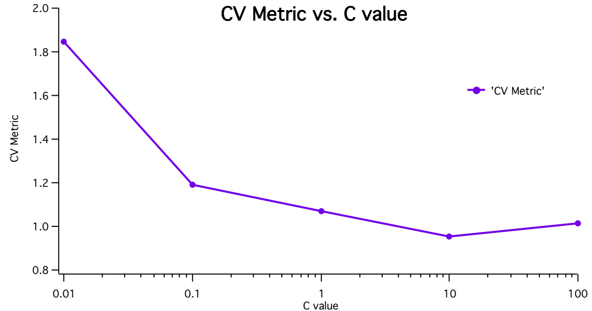


Fig. 8 CV Metric vs. C value

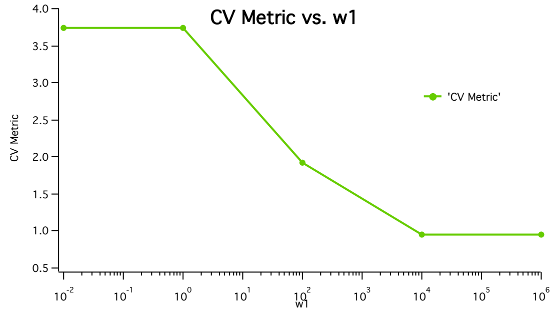


Fig. 9 CV Metric vs. weight 1

The optimal values for gamma, C and w1 were found to be 0.2, 10 and 1e6. This makes intuitive sense because if the algorithm is over-fitting (which was shown in Fig. 6), gamma should be lowered. C should be somewhat small but not too small that the algorithm starts under-fitting. Thus there is a minimum point of C =10 that minimizes CV metric. w1 should be large because of the unbalanced nature of the data set where there is a small number of 1's compared to 0's present in the data. In each sky, the ratio of 1's to 0's is given by the number of halos divide by the number of pixels and thus it will take values of 1/1764, 2/1764 or 3/1764 when sliding window algorithm is not implemented.

After performing SVM parameter tuning, I created another learning curve, Fig.10, to determine how well the algorithm is currently performing.

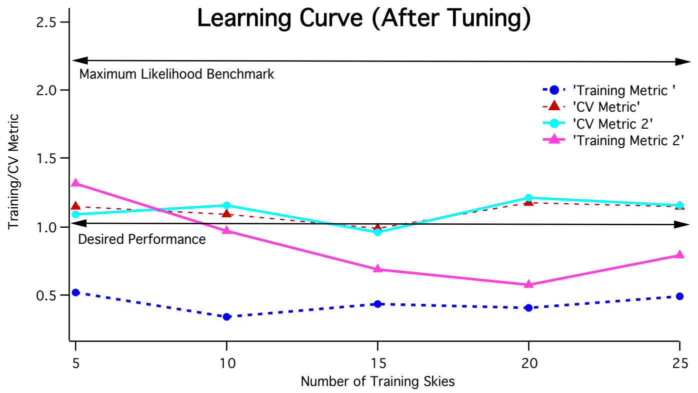


Fig. 10 Learning Curve after SVM parameter tuning

It is found that the algorithm has slightly improved the best CV metric from 0.9906 to 0.9551. The gap between training metric and CV metric has narrowed and this is expected considering that gamma was lowered which reduces over-fitting. The small gap could signify a slightly over-fitting algorithm.

### *B5. Sliding Window Implementation*

In order to tackle the slightly over-fitting issue, I implemented a sliding window algorithm. The window in my algorithm has a size of n by n pixels. Because the halos are point-like features, I do not need to implement a spatial pyramid algorithm, which works best for objects of a finite size (non-point based). A variant form of non-max suppression was implemented where regions near to a local peak will be ignored as possible halo locations. This does not alter the decision values of the points but merely ignores those points as possible halo positions. By varying the window size, it was found that the best window size is n =1, Fig. 11. This is likely because the algorithm is already performing very well and changing the window size does not help improve the algorithm anymore.

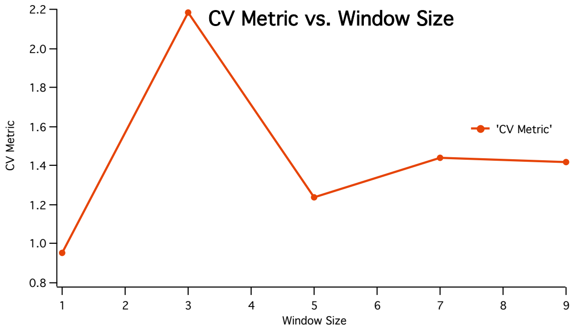


Fig. 11 CV Metric vs. sliding window size

### *C. Final Results*

After tuning the algorithm, the optimal performing algorithm has **<u>CV metric of 0.955</u>**. When the algorithm is tested on skies that it has never seen before, skies 41:50, 141:150 and 241:250, **<u>test metric of 0.893</u> is achieved**. This shows that the algorithm is predicting halo positions very well and my algorithm is performing better than my desired performance. As a comparison, publicly available algorithms online have test metric values of 2.2078, 1.58061 and 1.01880 for the Single Halo Maximum Likelihood, Gridded Signal Benchmark and Lenstool Maximum Likelihood respectively. The best performing algorithm, Lenstool Maximum Likelihood, has 42000 lines in its code.

### *D. Map Analysis*

To understand the results achieved, I created heat maps of each pixel's feature value and compared it with the decision value maps that show decision values and locations of predicted (red spots) and actual halo (blue spots) positions. The comparison is done to see if there is a relationship between feature and decision value that can be identified visually. The color scheme in the feature map, Figs. 13, 14 and 15 follows the rainbow sequence with red indicating the highest value and blue indicating the lowest value.

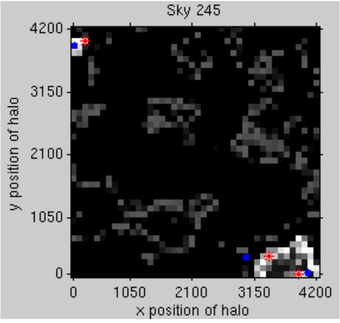


Fig. 12 Feature map of sky 245

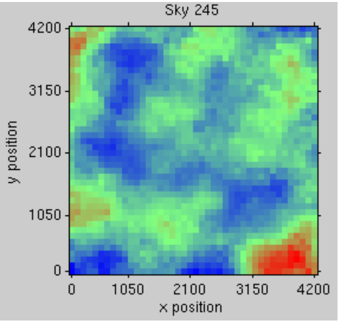


Fig. 13 Feature map of sky 245

Fig. 12 shows a sky with three halos and Fig. 13 shows the corresponding feature value. As can be seen, the blue spots (actual halo position) match up well with the predicted halo positions (red spots).

The algorithm was also successfully in identifying the three halo positions in Fig. 14.

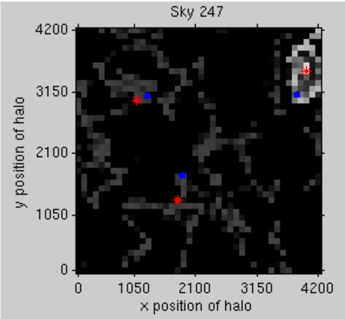


Fig. 14 Decision map of sky 247

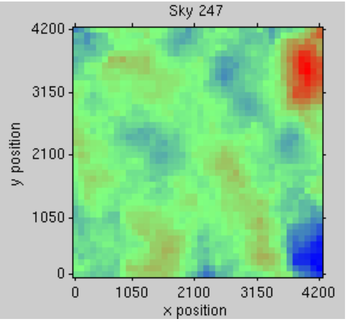


Fig. 15 Feature map of sky 247

From these two examples, it can be seen that the regions of high decision value (bright areas) correspond to regions of large gradient in the feature map. This is interesting because it signifies that the presence of a dark matter halo creates a large effect on the ellipticity of nearby galaxies with this effect dropping off rapidly with distance. And it is this rapid drop off in ellipticity from the halo center that characterizes whether a particular pixel contains the halo position. In the literature, this effect is said to drop off by 1/r, which is a rapidly decreasing function. Thus, the algorithm has come up with a decision-making process that is in line with current physics understanding of dark matter halos.

## CONCLUSION

The SVM algorithm has been very successful in predicting dark matter halos' positions. This algorithm has a **<u>CV metric of 0.955</u>**<u>4</u> and **<u>test metric of 0.893</u>**. For skies with multiple halos, the SVM is able to effectively identify the positions of the multiple halos.

My exploration of different techniques has led to the conclusion that selecting the right training set type was the key breakthrough for this predictive algorithm to work well.

Further work on this algorithm can be done on introducing features that incorporate more complex physical model such as gravitational lensing equations. Although computationally more intensive, a smaller bin size will likely help the accuracy of the predicted halos's locations.

## REFERENCES

[1] "Dark matter - Wikipedia, the free encyclopedia," *enwikipediaorg*. [Online]. Available: http://en.wikipedia.org/wiki/Dark_matter#cite_note-2. [Accessed: 20-Nov.-2012].

[2] "Gravitational Lens," *enwikipediaorg*. [Online]. Available: http://en.wikipedia.org/wiki/Gravitational_lens. [Accessed: 20-Nov.-2012].

[3] "Competitions - Kaggle," *kaggle.com*. [Online]. Available: https://www.kaggle.com/competitions. [Accessed: 20-Nov.-2012].

[4] "Observing Dark Worlds – Visualizing dark matter's distorting effect on galaxies « bayesianbiologist," *bayesianbiologist.com*. [Online]. Available: http://bayesianbiologist.com/2012/10/13/observing-dark-worlds-visualizing-dark-matters-distorting-effect-on-galaxies/. [Accessed: 20-Nov.-2012].