Mon. Not. R. Astron. Soc. **000**, 1–5 (2014) Printed 20 January 2016 (MN LaTeX style file v2.2)

Space Warps Extended! Snappy Titles!

Philip J. Marshall,[1,2]* Aprajita Verma,[2] Anupreeta More,[3] Christopher P. Davis,[1] Surhud More,[3] Amit Kapadia,[4] Michael Parrish,[4] Chris Snyder,[4] Julianne Wilcox,[5] Elisabeth Baeten,[5] Christine Macmillan,[5] Claude Cornen,[5] Michael Baumer,[1] Edwin Simpson,[6] Chris J. Lintott,[2] David Miller,[4] Edward Paget,[4] Robert Simpson,[2] Arfon M. Smith,[4] Rafael Küng,[7] Prasenjit Saha,[7] Thomas E. Collett[8]

[1] *Kavli Institute for Particle Astrophysics and Cosmology, Stanford University, 452 Lomita Mall, Stanford, CA 94035, USA*
[2] *Dept. of Physics, University of Oxford, Keble Road, Oxford, OX1 3RH, UK*
[3] *Kavli IPMU (WPI), UTIAS, The University of Tokyo, Kashiwa, Chiba 277-8583, Japan*
[4] *Adler Planetarium, Chicago, IL, USA*
[5] *Zooniverse, c/o Astrophysics Department, University of Oxford, Oxford OX1 3RH, UK*
[6] *Dept. of Engineering Science, University of Oxford, Parks Road, Oxford, OX1 3PJ, UK*
[7] *Department of Physics, University of Zurich, Winterthurerstrasse 190, 8057 Zurich, Switzerland*
[8] *Institute of Cosmology and Gravitation, University of Portsmouth, Dennis Sciama Building, Portsmouth P01 3FX, UK*

to be submitted to ?!?!

ABSTRACT

To Do: Chris: Do abstract!

Key words: gravitational lensing – methods: statistical – methods: citizen science

1 INTRODUCTION

2 DATASET

3 FORMALISM

To Do: Chris: This section will describe the different ways we can use the data we got from SpaceWarps to do a beter job with classifications. So we can then look at the way SpaceWarps updates the PL, PD, and p's of the objects, either via offline, using both/either/neither training and test data (in either online or offline contexts), manipulating initializations, etc. We could also see what the benefits are for using the known lens information. Finally, we should examine the benefits of using a validation dataset (using information we already have!) to improve our estimates. Chris: I don't like using p^0 to both represent the true prior prior and our estimate of the correct prior according to the EM algorithm.

Space Warps keeps track of the following parameters:

- C_{ij}, the classification the i-th volunteer made of the j-th image. C_{ij} may take on three values: 0, 1, or empty. Since volunteers do not see most images, the vast majority of C_{ij} are blank.
- PD_i, the probability, given that the image is a dud, that i-th the volunteer will classify it as a dud. The probability, given that the image is a dud, that the volunteer will classify it as a lens follows as $1 - PD_i$.
- PL_i, the probability, given that the image is actually a lens, that the i-th volunteer will classify it as being a lens. The probability, given that the image is a lens, that the volunteer will classify it as a dud follows as $1 - PL_i$.
- p_j, the probability that the j-th image z_j is a lens given the current observations and skills of the volunteers who classified it.
- p^0, the prior probability that an object is a lens.

3.1 The Online System

In Space Warps, this is fixed at 2×10^{-4}, or the expectation that around 100 lenses will be found in 430,000 images. Because Space Warps is an online system that constantly reevaluates most of the above parameters (except p_0 and

* pjm@slac.stanford.edu

any non-blank C_{ij}) in order to promote likely lenses[1] or to retire likely duds,[2] we augment p, PL, and PD as p_j^k, the evaluation of p_j at time k. Space Warps uses Bayes's Theorem to update $p_j^{(k+1)}$ for some new evaluation C_{ij}[3]:

$$p_j^{(k+1)} = \left(\frac{C_{ij} PL_i^k}{PL_i^k p_j^k + (1 - PD_i^k)(1 - p_j^k)} + \frac{(1 - C_{ij})(1 - PL_i^k)}{(1 - PL_i^k) p_j^k + PD_i^k (1 - p_j^k)} \right) p_j^k \ , \qquad (1)$$

The first term on the right hand side is the probability update for evaluating the object to be a lens, while the second term is the probability that the image is a lens if the volunteer evaluates it to be a dud. (For example, an obtuse volunteer who always perfectly incorrectly classifies an image will actually change the probability exactly the same as one who always perfectly correctly classifies an image, given that the estimate of the obtuse volunteer's skill ($PL_i = 0$) is accurate.)

Space Warps only updates the volunteer's PL_i and PD_i after volunteer C_i classifies a training image:

$$PL_i^{(k+1)} = \frac{PL_i^k (NL_i^k + M) + \mathbb{I}[C_{ij} = z_j]}{NL_i^k + M + z_j} \qquad (2)$$

$$PD_i^{(k+1)} = \frac{PD_i^k (ND_i^k + M) + \mathbb{I}[C_{ij} = z_j]}{ND_i^k + M + z_j} \qquad (3)$$

where ND_i^k and NL_i^k refer to the number of training lenses and training duds observed by the i-th volunteer at time k, z_j refers to the true state of the j-th image (1 is LENS, 0 is DUD), and $M = 4$ is a smoothing factor empirically derived to smooth the skill classification of new volunteers.

With these update rules plus an initialization of $PD_i = PL_i = 0.5$ and $p^0 = 2 \times 10^{-4}$, the online update system is fully specified.

3.2 An Offline Expectation Maximization Approach

Using the above notation but expanding p^0 to p_{ij}^0 (allowing, e.g. for the distribution of training images to differ for each volunteer, perhaps based on the number of images they have observed, or to allow a particular image to be more likely to be drawn), the complete log-likelihood for this model may be specified:

$$\begin{aligned} &\mathrm{CLL}(C_{ij}, z_j, PL_i, PD_i, p_{ij}^0) \\ &= \sum_i \sum_{j \in \Omega_i} C_{ij} z_j \log PL_i + (1 - C_{ij}) z_j \log(1 - PL_i) \\ &+ (1 - C_{ij})(1 - z_j) \log PD_i + C_{ij}(1 - z_j) \log(1 - PD_i) \\ &+ z_j \log p_{ij}^0 + (1 - z_j) \log(1 - p_{ij}^0) \end{aligned} \qquad (4)$$

[1] Note that this does not change the probability that a volunteer will actually draw said image.
[2] Images whose probability of being a lens drops below a certain threshold are removed from the active dataset.
[3] There is no superscript for C_{ij} because each user only sees an image once.

where Ω_i is the set of all images volunteer i has observed in Ω, the set of all images in the program.[4] We can use this complete log-likelihood to derive an offline expectation maximization algorithm for determining the lens probabilities, user skills, and lens priors.

3.2.1 E-Step

To Do: Chris: reword. The maximization here is over the zj's, so word to make that clear. The E-Step is just taking the expected complete log-likelihood, or the expectation value over $P(\cdot \mid x, \phi)$. The E-Step is maximizing the complete log-likelihood with respect to the image probability p_j. This is equivalent to replacing binary z_j with probability p_j: **To Do: Chris: replace this with equivalent like eq1**

$$p_j = \frac{1}{N_j} \sum_{i \in \Omega_j} P(z_j = 1 \mid C_{ij}; \Phi) = \frac{1}{N_j} \sum_{i \in \Omega_j} \frac{P(C_{ij} \mid z_j = 1; \Phi) P(z_j = 1; \Phi)}{P(C_{ij}; \Phi)} \qquad (5)$$

$$= \frac{1}{N_j} \sum_{i \in \Omega_j} \frac{PL_i^{C_{ij}} (1 - PL_i)^{(1 - C_{ij})} p_{ij}^0}{PL_i^{C_{ij}} (1 - PL_i)^{(1 - C_{ij})} p_{ij}^0 + PD_i^{(1 - C_{ij})} (1 - PD_i)^{C_{ij}} (1 - p_{ij}^0)} \qquad (6)$$

where $i \in \Omega_j$ is now the set of classifications done on the j-th image and N_j is the number of classifications done on the j-th image. This makes sense: p_{ij}^0 is just the prior likelihood of an image being a lens, while PL_i is how well we would have identified a lens as such.

3.2.2 M-Step

The M-Step is done by maximizing the expected complete log-likelihood with regard to the input parameters PD_i, PL_i, p_{ij}^0. Doing the maximization process, we find:

$$PL_i = \frac{\sum_{j \in \Omega_i} C_{ij} p_j}{\sum_{j \in \Omega_i} p_j} \qquad (7)$$

$$PD_i = \frac{\sum_{j \in \Omega_i} (1 - C_{ij})(1 - p_j)}{\sum_{j \in \Omega_i} (1 - p_j)} \qquad (8)$$

$$p_{ij}^0 = p_j \ , \qquad p_i^0 = \frac{\sum_{j \in \Omega_i} p_j}{\sum_{j \in \Omega_i} 1}$$

$$p_j^0 = p_j \ , \qquad p^0 = \frac{\sum_i \sum_{j \in \Omega_i} p_j}{\sum_i \sum_{j \in \Omega_i} 1} \ , \qquad (9)$$

where the possible specializations of p_0 are also given. These mirror quite closely the online systems, except that skill is now assessed against the majority expectation of the probability of an image being a lens, instead of its true value.

In practice we have training images where p_j is known. In those cases we can use the true value when doing the M-Step. We can also choose to perform the M-step using only the training images (a supervised learning approach), giving us another handle on quantifying the impact and importance of simulated training images.

[4] Because the probability that a viewer views a given image (given it is a training or test image) is random, I choose to simply ignore the unobserved images.

Figure 1. ROC curve of online vs offline for both stages

Figure 2. Scatter plot of p values from online and offline for both stage 1 and stage 2. Then a second plot of a similar thing but for the confusion matrices. Show if there are any systematic differences that need to be explained

Finally, we also include Laplace smoothing into the M-Step in order to handle pathologic cases, such as when users never identify any lenses ($\sum_{j\in\Omega_i} p_j = 0$). The estimators for PL_i and PD_i now become:

$$PL_i = \frac{M + \sum_{j\in\Omega_i} C_{ij} p_j}{2M + \sum_{j\in\Omega_i} p_j} \tag{10}$$

$$PD_i = \frac{M + \sum_{j\in\Omega_i} (1 - C_{ij})(1 - p_j)}{2M + \sum_{j\in\Omega_i} (1 - p_j)} \tag{11}$$

where for Laplace smoothing, $M = 1$.

We choose to specialize p^0 to vary with image, p_j^0, because images are taken out of the SPACE WARPS system if they reach too low a probability, clearly changing the prior when we evaluate at the end of the run; training images also have a different prior on being a lens as well. Finally, if the image is a training image with known $p_j \in (0, 1)$, then the known p_j is used instead of the current estimate.

An easy way to conceptualize this section is to note that the Expectation Maximization takes advantage of the fact that each classification is supposed to be independent of the others and so the dataset can be treated as though all classifications were made at the same time. The repetition of the E and M steps ensures that the parameters are self-consistent.

4 TESTS

4.1 Offline vs online

Chris: This section was somewhat talked about in SPACE WARPS paper 1, but it needs reiterating here.

4.2 Unsupervised vs supervised

4.3 Importance of initialization

4.4 Simulated vs known strong lenses

Chris: This section will talk about training on the small set of known strong lenses. I cut down the users only to those who observed strong lenses. I could use as a prior the distribution of confusion matrices and probabilities from the training, or maybe some sort of dirichlet model. I think it will be vital that I use an informative prior, since in my previous explorations of this matter there simply wasn't enough data to go off of.

Figure 3. This figure will show three sets of ROC curves: supervised only, unsupervised only, and both together.

Figure 4. Scatter plot of p values using supervised only vs unsupervised + supervised and unsupervised only. Then a second plot of a similar thing but for the confusion matrices. The idea is to show that (probably) there is no major systematic shift in evaluations.

Figure 5. This figure will show ROC curves for different initialization parameters.

5 EXTENDED SPACEWARPS

To further our understanding of the SPACE WARPS model, we reran our analysis on a subset of objects in the original dataset. This subset of objects includes all the known lenses, objects which were deemed 'unlucky misses' from having an unlucky streak of viewers push it out of the online system, and objects that had a large discrepancy between the offline and online systems (both supervised and unsupervised plus supervised). In general, we found that very few objects were missed by the offline system and found by the online system. We reran on this dataset with the hope of finding yet more lenses.

6 EXTENSIONS TO EXISTING MODEL

6.1 Expanding Number of Classifications

To Do: Chris: In this section, we expand possible classifications from LENS and NOT to, say 0, 1, 2, 3. In effect, I take the stuff of 3.2 and extend it to a multinomial model. I should put in the generative model here as well as the new confusion matrix. I have written the first several steps below (not properly formatted and all that).

Chris: In the below formalism everything can only be classified once. That is, classification from user i on subject j as type u is $C_{ij} = u$. If we want to expand classification numbers, we could say instead $C_{iju} = 1$ for 1 classification as type u. Then, if you draw M total classifications, you replace Uniform (p) with Mult (p, M).
Now you may ask: Chris, why would we ever make more than one classification on a subject? And I would agree, except that this provides an avenue for analyzing how to use SPACE WARPS to deal with the markers (where there usually are multiple classifications).

We can extend the SPACE WARPS binomial model to include multiple classifications and multiple categories – we can talk about things like $P(\text{“1”}|\text{LENSED QUASAR})$ where a user assigns an object a rank "1" and wish to know the likelihood that a lensed quasar would yield that category.

Unfortunately our PL and PD terms must be generalized. We introduce instead P_{ivu} to represent the i-th user's probability of making classification "u" given that the ob-

Figure 6. Scatter plot of p values. Then a second plot of a similar thing but for the confusion matrices. The idea is to show that (probably) there is no major systematic shift in evaluations.

ject is of type v: $P_{ivu} = P(\text{"}u\text{"}|v)_i$. Naturally, the conditional probabilities are normalized over the possible classifications "u": $\sum_u^U P_{ivu} = 1$. Overall there are U types of classifications a user can make, representing V types of objects.

We also must expand p^0 to represent more than a binary classification. p^0_{jv} is the prior probability that object j is of type v. Naturally, $\sum_v p^0_{jv} = 1$. The latent variable of subject j as object v is $z_{jv} = 1$ with all other $z_{jv} = 0$. Similarly, the classification of user i on subject j as type u is $C_{iju} = 1$ with all other $C_{iju} = 0$.

This is our generative model with explicit latent variables:

- Draw latent subject indicator vectors from the prior probability:

$$z_j \overset{\text{iid}}{\sim} \text{Uniform}\left(p^0\right)$$

- Given latent subject indicator z_j and the i-th user's confusion matrix P_i, independently draw a classification vector from the z_j-th column of P_i:

$$C_{ij}|z_j \overset{\text{ind}}{\sim} \text{Uniform}\left(P_{i,z_j}\right)$$

Hence, we can also make the following statements:

$$p(z_{jv}) = p^0_{jv} \tag{12}$$

$$P(C_{iju}|z_{jv}; p^0_{jv}, P_{ivu}) = P_{ivu}^{C_{iju}} \tag{13}$$

Thus our complete log likelihood is (ignoring irrelevant normalization terms)

$$\log p\left(C_{iju}, z_{jv}; p^0_{jv}, P_{ivu}\right) \propto \sum_i \sum_{j\in\Omega_i} \sum_v z_{jv} \log p^0_{jv} + \sum_i \sum_{j\in\Omega_i} \sum_v \sum_u C_{iju} z_{jv} \log P_{ivu} \tag{14}$$

6.1.1 E-Step

Now we repeat the exercise earlier of computing the expected complete log likelihood under the conditional distribution $p(z_j|C_{ij}; p^0, P_i)$. Call $p_{jv} = p(z_{jv}|C_{iju}; p^0_{jv}, P_{ivu})$. It is sufficient to compute p_{jv} to compute this step. By Bayes's Theorem, we know that Chris: Is this an abuse of notation? I want for a single z_{jv}, but given $C_{iju}, p^0_{jv}, P_{ivu}$ for *all* i and u. So really it is more correct to say $p(z_{jv}) \prod_i \prod_u p(C_{iju}|z_{jv}; p^0_{jv}, P_{ivu})$ since each classification C_{ij} is independent. Should I do that? Alternatively, I could put one of those $\cdot$ in place of each i and u...

$$p(z_{jv}|C_{iju}; p^0_{jv}, P_{ivu}) \propto p(C_{iju}|z_{jv}; p^0_{jv}, P_{ivu}) p(z_{jv})$$

But we know the values of all these terms, so we find: **To Do: Chris: Double check this. I'm not sure it's strictly correct when you have more than one classification per image / type.**

$$p_{jv} = \frac{1}{N_j} \sum_{i\in\Omega_j} \frac{p^0_{jv} \prod_u P_{ivu}^{C_{iju}}}{\sum_{v'} p^0_{jv'} \prod_u P_{iv'u}^{C_{iju}}} \tag{15}$$

where $N_j = \sum_{i\in\Omega_j} 1$.

6.1.2 M-Step

We maximize the expected complete log likelihood. This is straightforwardly **To Do: Chris: Maybe not so straightforwardly if you have multiple classifications. Check that!**

$$P_{ivu} = \frac{\sum_{j\in\Omega_i} C_{iju} p_{jv}}{\sum_{j\in\Omega_i} p_{jv}} \tag{16}$$

$$p^0_{jv} = p_{jv} \tag{17}$$

$$p^0_v = \frac{\sum_i \sum_{j\in\Omega_i} p_{jv}}{\sum_i \sum_{j\in\Omega_i} 1} \tag{18}$$

where p^0_v is a possible specialization of p^0_{jv}.

6.1.3 Online

Like in the binomial case, the multinomial case can be performed online, where each new piece of information updates an existing assessment of lens classification and user skill. If a new classification $C_{iju}^{(k+1)}$ is made, then by Bayes's Theorem the updates are as follows (with i, j, u, and v acting as fixed indices): Question from Chris: In our case, there is only one classification u such that $C_{iju'} = 1$ if $u' = u$ and 0 else. Is it more clear if I instead call our classification C_{ij} and multiply over all u indices? **To Do: Chris: Double check whether the updates to p and P use the $k+1$ term of the other, and if so, the ordering of it.**

$$p_{jv}^{(k+1)} = \left(\frac{\left(P_{ivu}^k\right)^{C_{iju}^{(k+1)}}}{\sum_{v'} p^k_{jv'} \left(P^k_{iv'u}\right)^{C_{iju}^{(k+1)}}} \right) p^k_{jv} \tag{19}$$

$$P_{ivu}^{(k+1)} = \frac{C_{iju}^{(k+1)} p^k_{jv} + P^k_{ivu} \sum_{j'\in\Omega_i} p^k_{j'v}}{p^k_{jv} + \sum_{j'\in\Omega_i} p^k_{j'v}} \tag{20}$$

$$P_{ivu}^{(k+1)} = \left(\frac{p^k_{jv}}{p^k_{jv} + \sum_{j'\in\Omega_i} p^k_{j'v}} \right) C_{iju}^{(k+1)} + \left(\frac{\sum_{j'\in\Omega_i} p^k_{j'v}}{p^k_{jv} + \sum_{j'\in\Omega_i} p^k_{j'v}} \right) P^k_{ivu} \tag{21}$$

Question from Chris: Do we like the first or second way of writing out $P_{ivu}^{(k+1)}$? If we are using only a supervised learning set, then $p^k_{jv} \in (0, 1)$, and Ω_i becomes instead of the set of all images classified by user i the set of all training images classified by user i. **To Do: Chris: Put in the Laplace smoothing. Double check that it again is just M and $2M$ in the numerator and denominator, respectively.**

6.1.4 Checks

To Do: Chris: This section will make some obvious checks on the generalized model. First I should be able to write down the binary model using this formalism. Next, I should be able to show for sure that different u can lead to different probability assignments if v can only be two categories – that is, I should be able to show that '1' gives a mixture of probability for LENS and NOT that makes sense when compared with '0' or '2'.
This section can probably be removed. This is more for internal sharing to help keep everyone up to speed.

Figure 7. This figure will be an ROC curve of the two subtypes in stage2 using the above way of doing the confusion matrix. Also plotted will be the ROC curves of the subtypes using just the binary classification. Do we do better by actually specifying? Or are we basically as well off just taking our binary distribution and dividing up? Failure modes are: either we don't have enough sims of each subtype to make this worthwhile, or it doesn't help.

Let's make sure we recover the old system. In our new notation, u could be "LENS" or "NOT" while v could be LENS or NOT. So, $PL_i = P_{iLL}$ and $PD_i = P_{iNN}$. We use the conservation of total probability to note that $P_{iLN} = 1 - P_{iLL} = 1 - PL_i$ and $P_{iNL} = 1 - P_{iNN} = 1 - PD_i$. Meanwhile, p^0_j becomes p^0_{jL} and we use conservation of total probability to note that $p^0_{jN} = 1 - p^0_j$. We now account for the labels: $z_j = z_{jL}$ and $z_{jN} = 1 - z_j$. (There can only be one label on these objects.) Similarly for the classifications: $C_{ij} = C_{ijL}$, so $C_{ijN} = 1 - C_{ij}$.

We can hence expand the sums over $v \in (L, N)$ and $u \in (L, N)$ to find that our CLL is proportional to:

$$\begin{aligned} z_{jL} \log p^0_{jL} + z_{jN} \log p^0_{jN} + C_{ijL} z_{jL} \log P_{iLL} + C_{ijN} z_{jL} \log P_{iLN} \\ + C_{ijL} z_{jN} \log P_{iNL} + C_{ijN} z_{jN} \log P_{iNN} = \\ z_j \log p^0_j + (1 - z_j) \log(1 - p^0_j) + C_{ij} z_j \log PL_i + (1 - C_{ij}) z_j \log(1 - PL_i) \\ + C_{ij}(1 - z_j) \log(1 - PD_i) + (1 - C_{ij})(1 - z_j) \log PD_i \end{aligned} \tag{22}$$

which is the same CLL as before.

6.1.5 Test

6.2 The Prior

6.3 Validation of Training

To Do: Chris: In order to look at a validation dataset, I need to create a blacklist of agents and subjects (and decide what sort of division of either / both / neither constitutes a good validation dataset here) that SWAP.py can then read in and process.

A validation dataset is needed to prevent overfitting training data, to try to test how the training data does against real lenses, and to train the hyperparameters from any priors we might use. These need to come in two categories: actual lens environments, and simulated lenses. The need for two sets arises because of the paucity of actual lenses. In SPACE WARPS, most training objects are explicitly given to a user – if a user incorrectly identifies a training object, they are informed of that failure, and likewise for a correct classification.

Problems: 0. By telling users about failures in the sim/dud, we explicitly break the notion that users come in fully-formed. But we also need to give users incentives to keep working, by rewarding them for good behavior. Replace by current crowd assessments and telling them how they did compared with the crowd? 1. SPACE WARPS does not have any 'silent' training images which could function as validation datasets, since you have broken the training by telling users. 2. How to manage the difference between 'dud known lens fields' and 'dud sim fields'? Just put them together? 3. If you tell users about the correct sim classification, do you also tell them about a correct lens classification, and do you tell them it was a real lens? 4. Can I just use the dud fields?

Figure 8. This figure should have a purpose

6.4 Dynamic Allocations

To Do: Chris: This section will examine whether we would be better off dynamically assigning images to people (based on current estimates of skill and probability) rather than randomly assigning them. This means I will need to create some sort of 'Space-Warps Emulator'. (Actually, the other thing I can do is use the blacklist developed earlier to throw out contributions in an informed manner in order to construct different simulations.) Before I even do that, another aspect I would like to examine and is probably useful is whether we can reliably identify skilled users early on. That is, instead of looking at final skill vs effort, look at many trajectories of skill vs effort over time. That would be a useful thing to show to prove that we can find the good users in the first place – early on!

6.5 Relaxing Constant Confusion Matrices

7 DISCUSSION

8 CONCLUSIONS

ACKNOWLEDGEMENTS

This paper has been typeset from a TEX/ LATEX file prepared by the author.