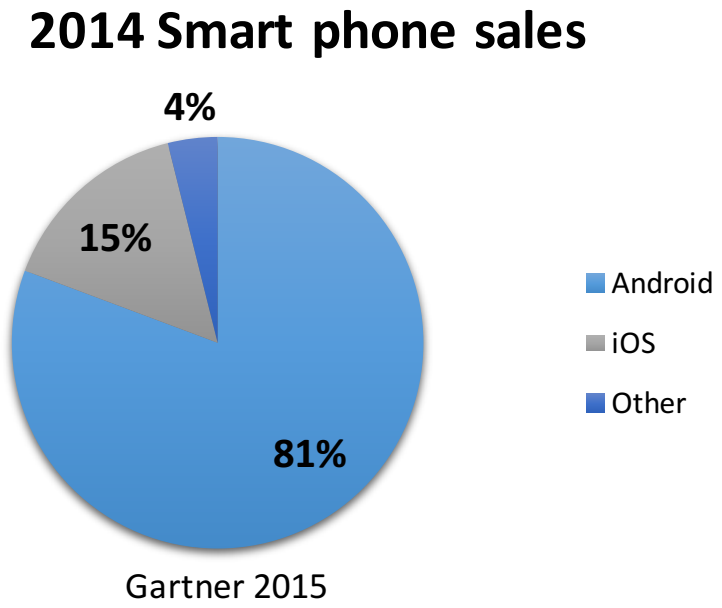
Android Malware Detection with Weak Ground Truth Data

Jordan DeLoach[1], Doina Caragea[1], Xinming Ou[2]

The 3rd International Workshop on Pattern Mining and Application of Big Data

[1] Kansas State University [2] University of South Florida

Android Malware



- Android dominates market share world wide
- Common malware behavior:
  - Leaking personal data
  - GPS tracking
  - SMS messages to premium numbers
- Reported levels of malware in the Google Play Store vary anywhere from Google's self-reported less than 1% to 7% or higher [5][11].
- Machine learning has been proposed as a way to take old apps that are malware or benign and learn classifiers from.

Machine Learning Limitations

- Often there exists no strong ground truth for an app, even with expert investigation.
- Often “time will tell” holds true, but research has shown [10], it is more likely to first be considered benign than malware, than the other way around.
  - We see 1681 times that a benign app is relabeled as *high confidence malware*, where we see the opposite happen only 1 time
- This creates an environment where we are comparatively certain of the malware label, but not of the benign label.

Semi-supervised Learning

- Semi-supervised learning allows for training over of unlabeled data, in addition to labeled data
- The intuition of a strong belief in one label, but less so with another fits well with a one-class semi-supervised learning algorithm.

Label Regularized Logistic Regression

- We use Label Regularized Logistic Regression (LR-LR), a modified version of Logistic Regression.
- LR-LR allows us to train over just the positive (malware) instances and regularize those expectations via an expert provided value, $\tilde{p}$.

$\tilde{p}$	Expert prediction of distribution of unlabeled
$\hat{p}_\theta^{unlab}$	Current classifier expected distribution
$\log p_\theta(y_i \mid x_i)$	Log likelihood over the data`
$D(\|\|)$	The Kullback-Leibler Divergence between two distributions
λ^U	Label Regularization Parameter
λ^{L^2}	L^2 Regularization Parameter

$$O(\theta) = \sum_i^N \log p_\theta(y_i \mid x_i) - \lambda^{L^2} \sum_k \theta_k^2$$

Logistic Regression

$$O(\theta) = \sum_i^{N_p} \log p_\theta(y_i \mid x_i) - \lambda^U D(\tilde{p} || \hat{p}_\theta^{unlab}) - \lambda^{L^2} \sum_k \theta_k^2$$

Label Regularized Logistic Regression

Dataset & Features

- ~1 million apps from the Google Play Store, VirusShare, and Arbor Networks
- Ran through VirusTotal to get multiple anti-virus label opinions
- 10 or more are *high confidence malware*, exactly 0 are considered *benign*, and the rest are excluded from the study
- Feature vector representation come from the semantic meaning of the app binary including permissions, APIs used, intents, etc.

Algorithms

Supervised

- Naïve Bayes
- Support Vector Machine (untuned)
- Logistic Regression

Semi-supervised

- Label Regularized Logistic Regression

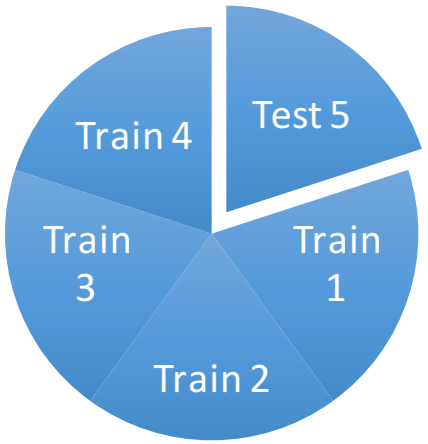
RQ1 - Injected Noise

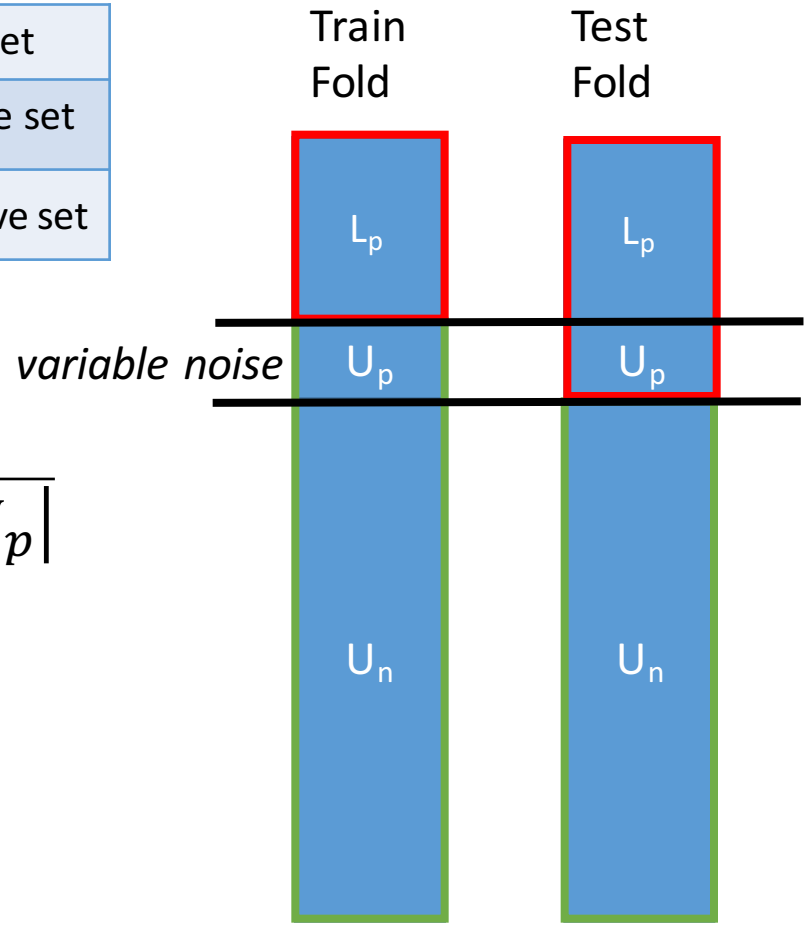
- We know in our benign dataset there exists noise, or apps that are mislabeled that will impair classifier performance.
- We seek to study how noise affects classifier performance, and which classifier algorithms are best at handling noise.
- To do this, we intentionally inject noise and study how performance changes on the “real labels” while testing as there is noise induced via “fake labels” in the training.

Noise, Cross-Validation & Evaluation

L_p	Labeled positive set
U_p	Unlabeled positive set
U_n	Unlabeled negative set

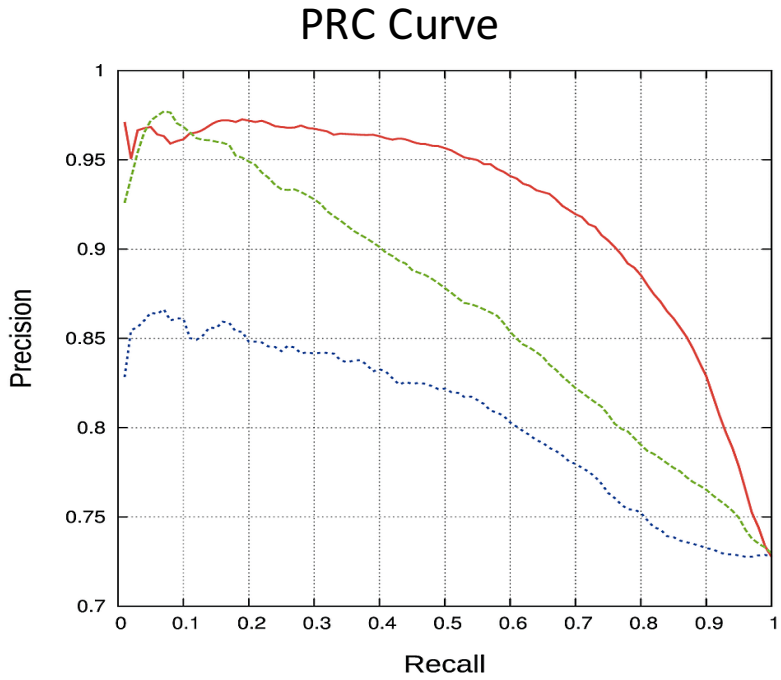
$$noise = \frac{|U_p|}{|U_n + U_p|}$$



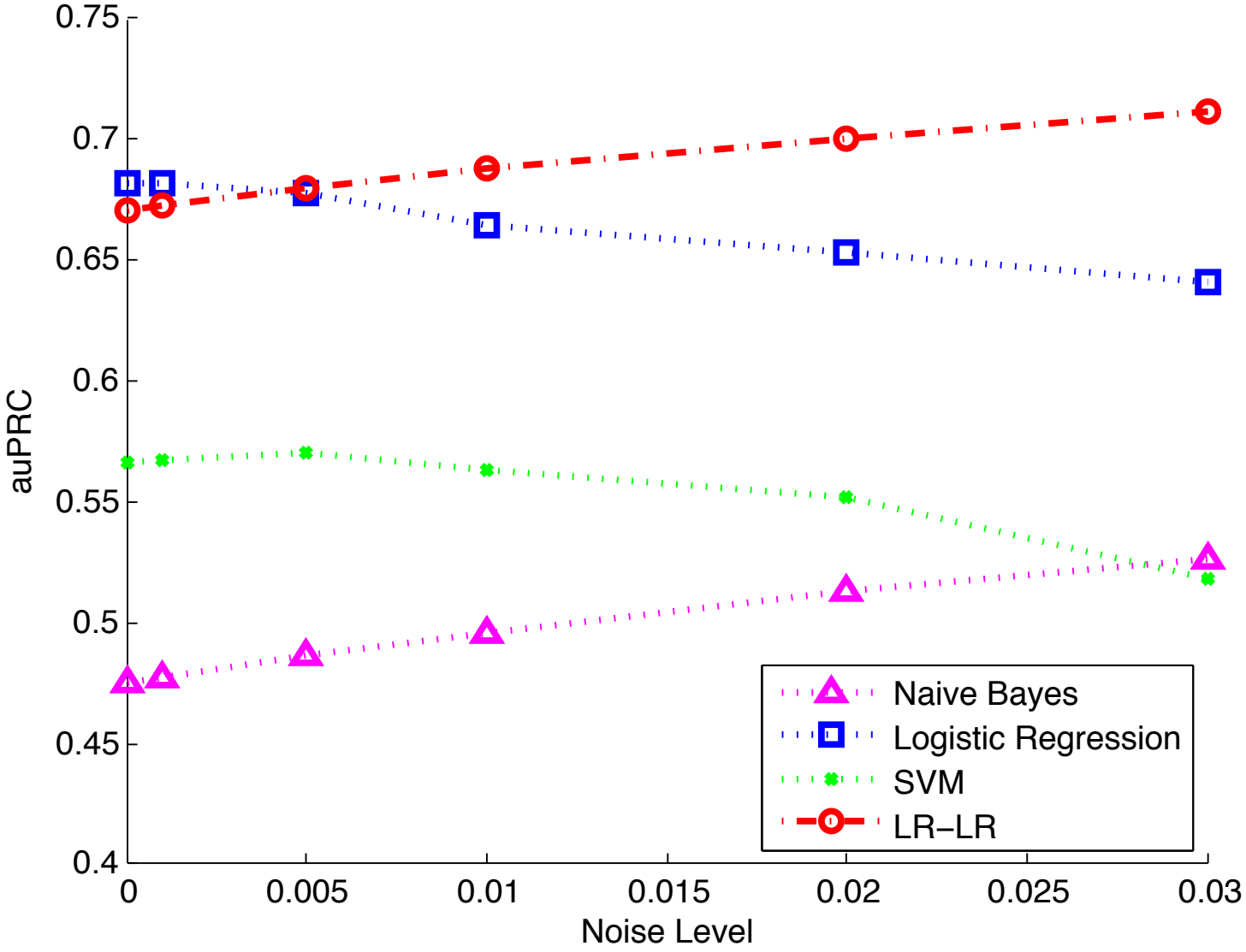


Precision	$\frac{TP}{TP + FP}$
Recall	$\frac{TP}{TP + FN}$

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$



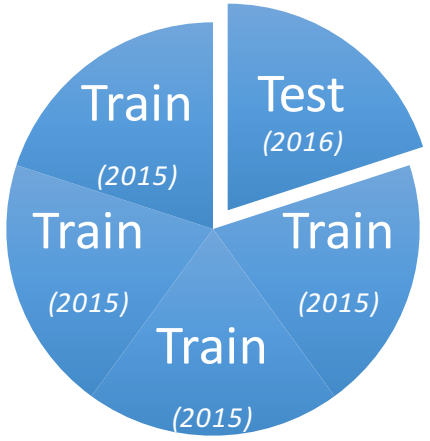
RQ1 - Injected Noise - Results



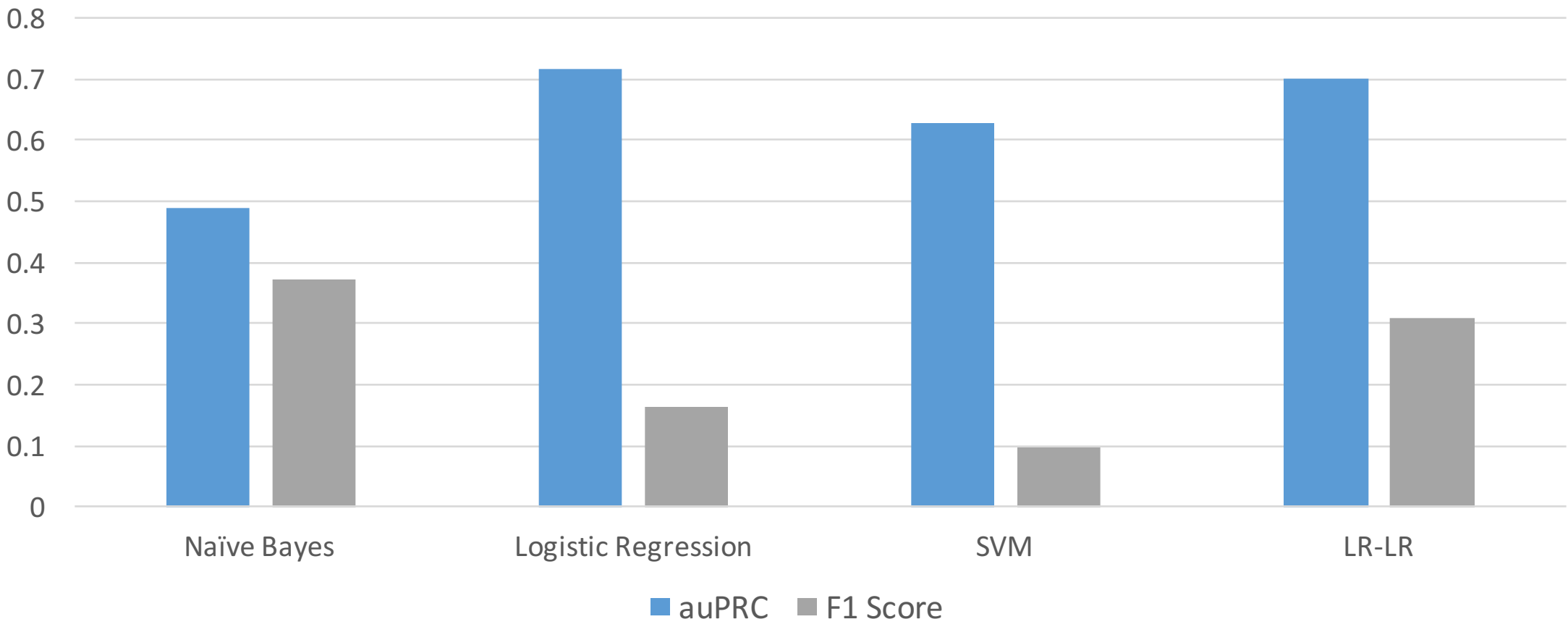
RQ2 - Future Predictability

- We know initial labels are often incorrect, and if they change, it will generally be from benign to malware
- We study what algorithm is best suited to predict the actual labels (2016 labels), given the initial mislabels (2015 labels) in training
- We evaluate using F1 Score on the 1682 apps with changed labels
  - With only 1681 false negatives, 1 false positive, auPRC is overly optimistic as precision is near 1.0 at most threshold values, F1 threshold at 0.5 presents a better view
  - We also show auPRC for the entire dataset

$$< train2015Label, test2016Label, feature1, feature2, feature3, \dots >$$

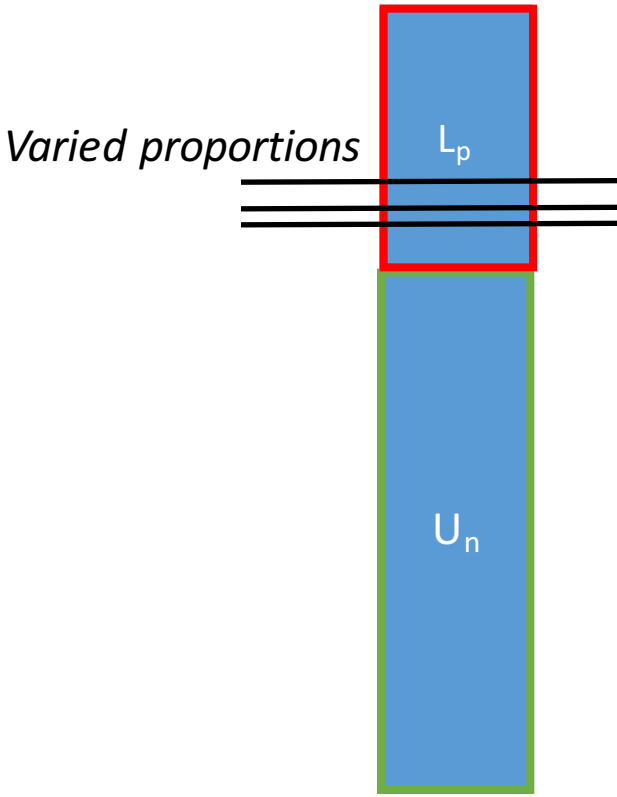


RQ2 - Future Predictability - Results

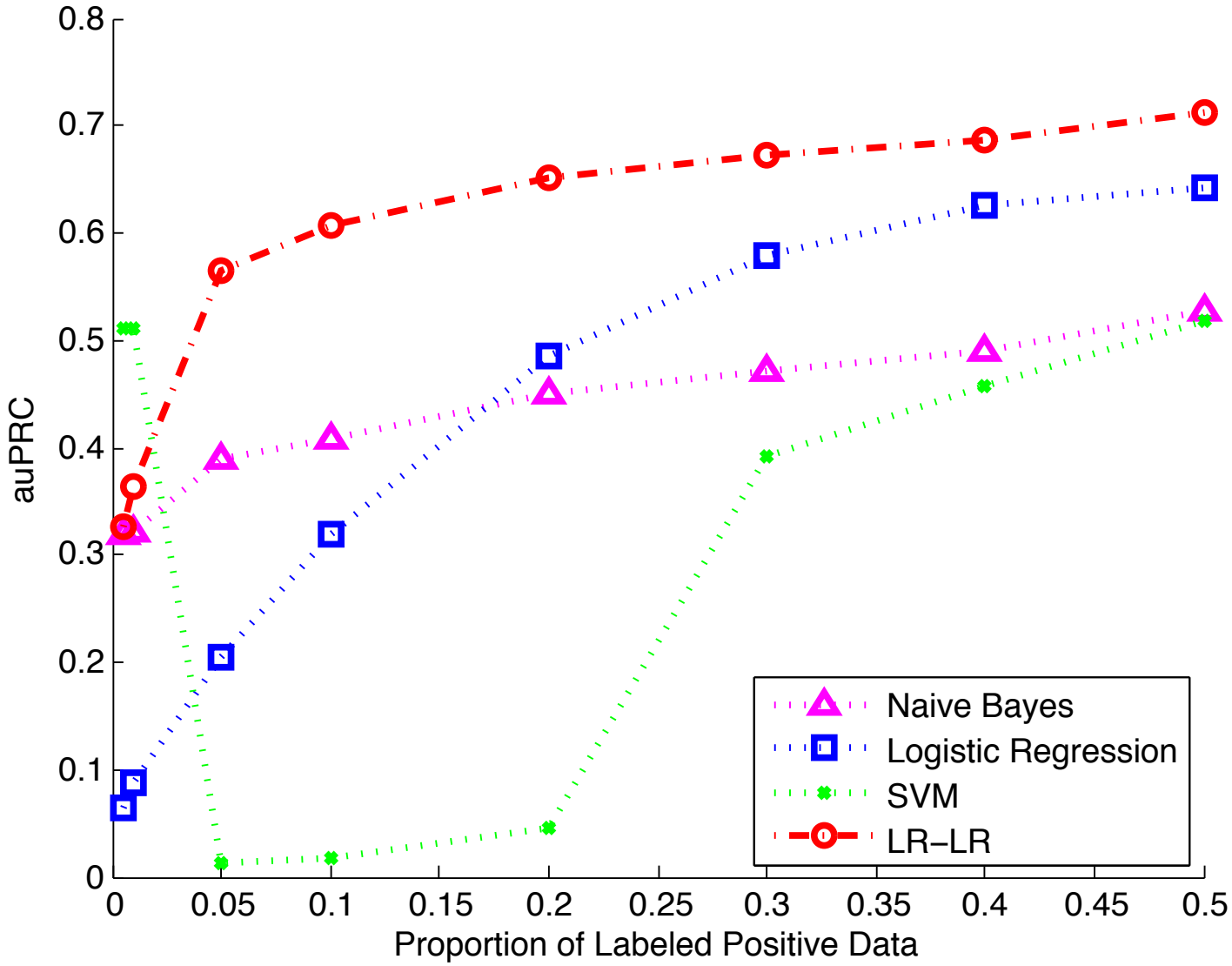


RQ3 - Minimizing Labeled Data

- Accurate labeled data in most domains is a limited commodity.
- As such, while more labeled data can be generated, at a cost, we experiment to see which algorithm can perform best with limited ground truth.
- We vary the proportion of the labeled positives in the dataset to simulate having less labeled data



RQ3 - Minimizing Labeled Data - Results



Contributions & Future Work

- Contributions
  - We carefully examine the inherent flaws in Android malware ground truth labels.
  - We are the first work to specifically and intentionally apply semi-supervised learning techniques to remedy these flaws.
  - We find LR-LR, and semi-supervised learning in general, to be a promising area for further exploration.
- Future Work
  - Finding new ways to supplement the detection process, potentially including social media elements to add further dimensions to the detection process.
  - More sophisticated ways of combining A/V recommendations.

Experiments & Spark LR-LR Implementation Available: https://github.com/jdeloach/mlbSparkExperiments

Backup Slides

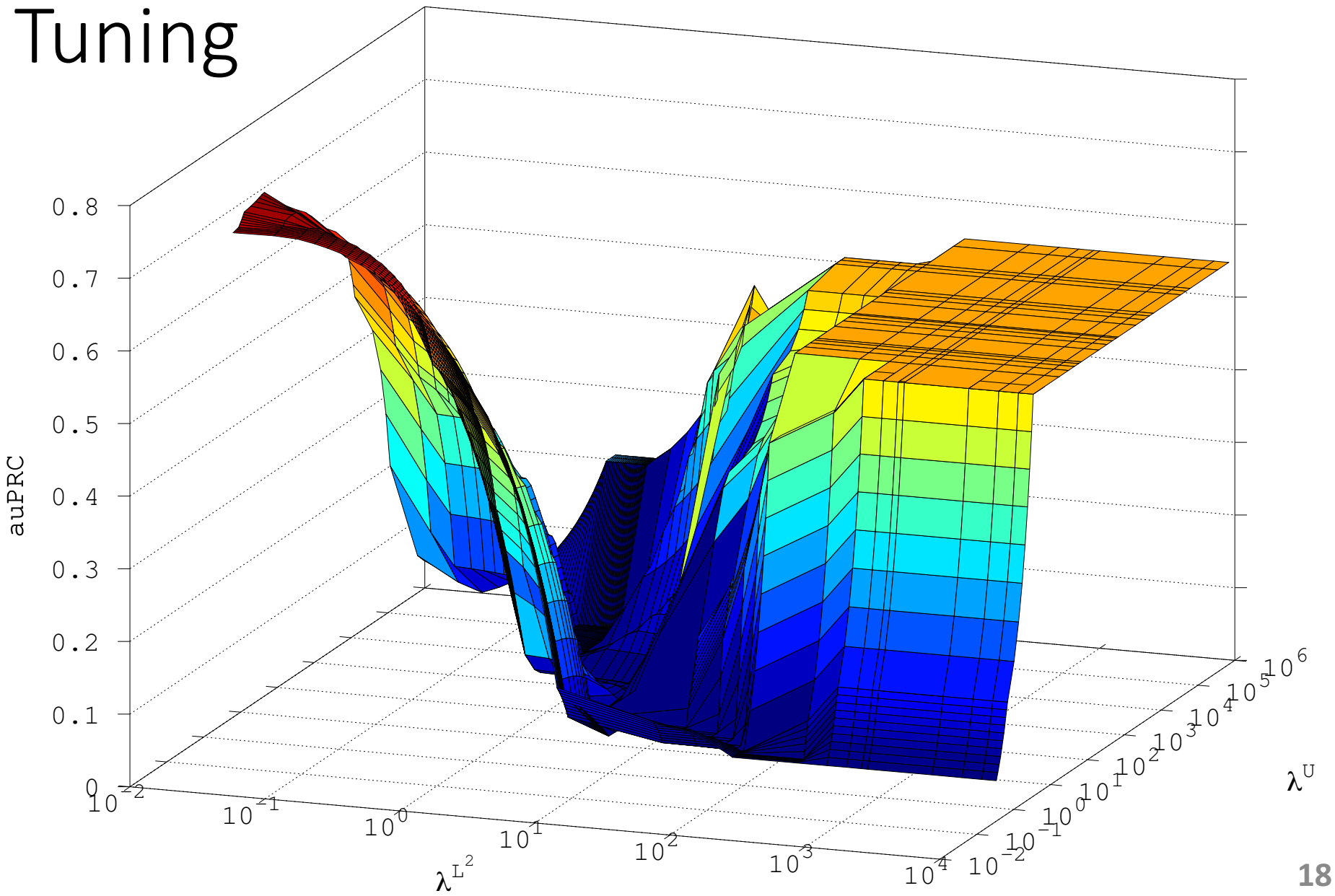
References

[5] K. Chen, P. Wang, Y. Lee, X. Wang, N. Zhang, H. Huang, W. Zou, and P. Liu. Finding unknown malice in 10 seconds: Mass vetting for new threats at the google-play scale. In 24th USENIX Security Symposium (USENIX Security 15), pages 659–674, 2015.

[10] A. Kantchelian, M. C. Tschantz, S. Afroz, B. Miller, V. Shankar, R. Bachwani, A. D. Joseph, and J. D. Tygar. Better malware ground truth: Techniques for weighting anti-virus vendor labels. In Proceedings of the 8th ACM Workshop on Artificial Intelligence and Security, AISec '15, pages 45–56, New York, NY, USA, 2015. ACM.

[11] Lookout. 2014 mobile threat report. http://bit.ly/1fktFwe, 2014.

Parameter Tuning



F1-Score (cont.)

- We use the F1-score due to the substantial imbalance of having only one negative in the 1682 apps, precision can be very misleading as there is a maximum of 1 false positive.

RQ1 (cont.)

- It is interesting to note that the performance of Naïve Bayes improves with the amount of noise.
- We speculate this is due to the fact that Naïve Bayes may be skewed towards estimating positives, and the introduction of positive noise in the train and test sets means that it gets less predictions wrong in the testing phase.

Label Regularized Logistic Regression Optimization Function

$$\frac{\partial}{\partial \theta_k} D(\tilde{p}||\hat{p}_\theta) = \frac{1}{N_u}\left(\frac{1-\tilde{p}}{1-\hat{p}_\theta} - \frac{\tilde{p}}{\hat{p}_\theta}\right) * \sum_{i=1}^{N_u} p_\theta(y_i = 1|x_i)(1 - p_\theta(y_i = 1|x_i))x_{i,k}$$