User Model Transfer for Email Virus Detection

Marco Barreno Blaine Nelson Russell Sears Anthony D. Joseph
Computer Science Division
University of California, Berkeley
{barreno,nelsonb,sears,adj}@cs.berkeley.edu

ABSTRACT

Systems for learning to detect anomalous email behavior, such as worms and viruses, tend to build either per-user models or a single global model. Global models leverage a larger training corpus but often model individual users poorly. Per-user models capture fine-grained behaviors but can take a long time to accumulate sufficient training data. Approaches that combine global and per-user information have the potential to address these limitations. We use the *Latent Dirichlet Allocation* model to transition smoothly from the global prior to a particular user's empirical model as the amount of user data grows. Preliminary results demonstrate long-term accuracy comparable to per-user models, while also showing near-ideal performance almost immediately on new users.

1. INTRODUCTION

Most successful viruses and worms spread via email [18]. This work explores the use of machine learning to decrease reliance on signature-based email virus detection.

Traditional network- and host-based virus scanners rely on manually crafted signatures and heuristics and have difficulty detecting novel viruses.[1] This creates a window of vulnerability each time a virus is released. For example, anti-virus vendors took over seven hours on average to generate a virus signature for the MyDoom.BB outbreak [16].

In contrast, machine learning techniques automatically model behavioral features of normal (or abnormal) email traffic, allowing them to detect unknown attacks by recognizing subtle deviations from normal activity. Most existing machine learning approaches to virus detection use either global or per-user models.

Global models attempt to generalize across all users (or network events, etc.) to leverage the full scope of data available. They often benefit from plentiful training data but their accuracy may be limited by variations between users. Network intrusion detection systems usually build global models, in particular for anomaly detection [5, 11]. These systems build a model of typical user or network behavior and flag activity that falls outside the learned model.

Per-user models treat each user's behavior independently, as is common in personal spam detection systems [14, 15]. Separate per-user models can be more accurate in the long run but suffer from a lack of training data when a new user enters the system.

Single	Window
CharsInSubject	NumToAddrInWindow
LinksInEmail	MeanCharInSubject
AvgWordLength	MeanWordsInBody
WordsInBody	VarCharInSubject
WordsInSubject	VarWordsInBody

Table 1: The "single" column features are derived from one email, while the "window" column is computed from the five most recent emails.

Some existing approaches also combine global and per-user information in their models. The Email Mining Toolkit uses several machine learning methods, including naive Bayes classification and social network analysis, on both global and per-user levels to detect email-borne viruses. Using a back-and-forth search heuristic, it finds agreements between the models to classify sequences of malicious emails [19].

Another combined system, APE, uses both a global model and per-user models in real-time to provide dynamic containment of worms and viruses. It uses a global model to flag suspicious messages, which are then classified by per-user models [12, 13].

Our approach uses *Latent Dirichlet Allocation*, a probabilistic model that combines global and per-user information, gracefully transitioning between them as more user data becomes available [2].

1.1 Features for Email

We represent emails using the features in Table 1. Those in the "Single" column are computed from a single email, while those in the "Window" column are computed based on a sliding window. All features are modeled with the Gaussian distribution except for LinksInEmail, for which we use the Binomial distribution. We do not use message headers, attachment information, or language-based features, such as word frequency. Instead we focus on simple properties of the email text and user sending patterns. Our feature set is based on a previous study of feature selection for email anti-virus systems [12].

Dataset limitations preclude the use of attachment information (see Section 3.1). Such features are useful but are not silver bullets, since not all viruses require attachments to propagate. For example, the BubbleBoy virus spreads via a script embedded in an email. When

[1] We do not distinguish between email-based worms and viruses.

viewed by a vulnerable mail client, it infects the system.

Some features we considered, such as “Number of From addresses from one sender in a window,” trivially classify large portions of the datasets we use. However, in all these cases the feature in question could easily be spoofed by a virus. We omit such features from our system, so our results are somewhat pessimistic.

2. USING LDA FOR EMAIL

Our system is based on the premise that different users exhibit many of the same canonical behaviors when sending email, but in different proportions. Likewise, viruses all spread from host to host in some manner, so even new viruses will have some behaviors in common with known viruses. We use the Latent Dirichlet Allocation (LDA) model [2] to combine user-specific training with behavioral information learned from the full population of users and known viruses.

LDA is a probabilistic model that represents items (in our context, emails) in terms of *topics* that group items by shared characteristics. We represent an email as a vector of features. Due to our choice of features, a topic in this setting corresponds to a type of user behavior or style of email (e.g., we have observed a topic that contains primarily long forwarded emails and another that has short bodies and empty or one-word subject lines). A topic groups emails that share characteristics described by our feature set. A user is represented as a multinomial distribution over topics. In other words, LDA extracts common behaviors and represents each user as an individual pattern of those behaviors.

The remainder of this section describes mixture models and LDA in the context of email virus detection. Figure 1 shows graphical model representations [8] of the joint probability distributions of these models.

2.1 Variables and Notation

Let K be the number of topics (chosen as a model parameter) and F be the number of features. In Figure 1, x is a vector of F components, z and α are scalar values, θ is a vector of K components, and β is a $K \times F$ matrix of parameters. M is the number of users, and E is the number of emails sent per user.[2] Variables inside the rectangular plates are replicated, so each model depicts a total of $E \times M$ variables x, and so on; we do not distinguish these notationally.

2.2 Mixture Models

A mixture model is a statistical tool for modeling datasets containing multiple subpopulations, each with a simple distribution (such as the Gaussian distribution). Mixture models can be used for global or per-user modeling. In Figure 1a we show an example of a global mixture model for email. A corpus of messages is represented by a single mixture model, in which each topic is a subpopulation. Each email x is assigned a topic z, which selects the parameters of the email’s feature distributions. There is a global distribution θ over topics and a global set of parameters β for feature distributions. There is no differentiation between users in this model.

Another approach is to use a separate mixture model for each user, as shown in Figure 1b. Here again each topic is a subpopulation, but now each user has their own feature parameters β and topic distribution θ. There is no sharing of information across users in this model.

2.3 LDA: Modeling Email Users

The graphical model representation of LDA appears in Figure 1c. Like both mixture models, each email x belongs to a particular topic z that determines the distributions for the features of x. Like the global model, LDA has one shared β for all users; like the per-user model, each user has a separate θ.

The LDA model can be described as a generative process, with a global prior α on topics from which a multinomial parameter θ is drawn for each user. When a user sends an email, the email’s topic z is drawn from θ, and then an email x is produced according to the corresponding distribution from β. Each row of β contains the parameters of one topic’s distribution.

Exact inference in LDA is intractable. We use a variational approach with surrogate parameters γ and ϕ for approximate inference and parameter estimation, as in the original LDA paper [2].

2.3.1 Extending LDA for Features

In the original presentation of LDA, β holds parameters of multinomial distributions. In a population of users modeled as collections of emails, however, a multinomial is not rich enough to represent an email.[3]

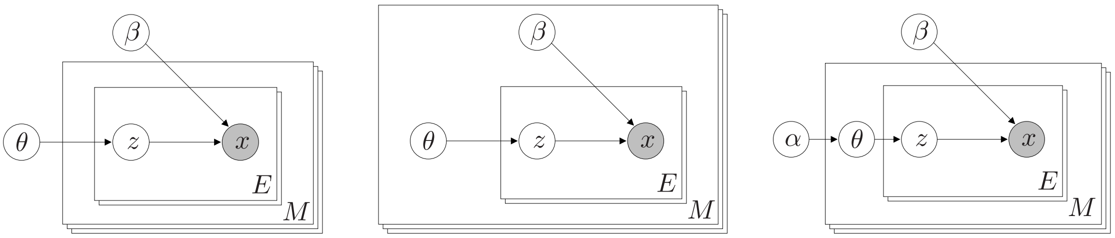
We extend LDA to model each email x as a vector of features. The distribution of an email is a fully factored naive Bayes model: given the topic, the features (which can have different distributions) are independent. Other models for the joint distribution of an email’s features could also be used in place of naive Bayes.

2.3.2 Shared Global Behaviors

Information is shared between users via the global parameters α and β. The prior distribution on user parameters θ is Dirichlet with parameter α, and estimating α from training data yields a prior from which to draw θ parameters for new users. To understand how LDA balances between this prior and the empirical data accumulated for a user, assume that the email topics z are known. Then for each user we can count the emails from each topic. Given the Dirichlet parameter α, the posterior distribution of θ is Dirichlet with parameter $\alpha + \mathbf{N}$, where $\mathbf{N}$ is a vector that contains the number of emails from each of the K topics [1]. As the number of emails for a user increases, the expectation of θ smoothly transitions from the prior based on α to the empirically ob-

[2] For convenience our discussion assumes that each user sends the same number of emails, but this assumption is not important and is in fact not true for our experimental data.

[3] Note that a document in the original LDA paper corresponds to one of our users, while one of their words corresponds to one of our email messages.



(a) Global Mixture Model (b) Per-User Mixture Model (c) LDA Model

Figure 1: Representations of the mixture models using the graphical model formalism [8]. Each node represents a random variable in the model, and the graph represents their joint probability distribution. x is an email, z is the topic of that email, θ is a user's distribution over topics, α is a prior over user distributions, and β represents the parameters for feature distributions (indexed by topic). The plates show replication of variables, indicating M users each with E emails. In all models there is one x and one z per email per user, and the most significant difference between models is whether θ and β occur once per user (inside the M plate) or just once in the model (outside both plates).

served distribution of topics. Although email topics are not actually observed, this provides intuition into how empirical data eventually outweighs the Dirichlet prior.

2.4 Classification

We use a generative approach for classification. We train two models; one learns normal behavior while the other learns virus behavior. To classify an email, we compute the likelihood that the email would be generated under each model and choose the class with the higher likelihood. The likelihood depends on the user that sent the email. For the normal model, we use the sender to compute likelihood; for the virus model, we choose the virus most likely to generate the email.

3. EXPERIMENTS

We perform experiments to compare LDA's ability to learn a new user's behavior with models that use only global or per-user information. The learners we compare against are a global mixture model (GMM) and per-user mixture model (PMM) as described in Section 2.2, as well as a linear support vector machine (SVM) [7].

3.1 Datasets

Our experiments use the Enron email corpus [4] and emails generated by real-world viruses.

The Enron corpus consists of emails subpoenaed as trial evidence and made public. As the only large, commercial, real-world email dataset, it is a useful resource despite several concerns: most attachment information has been stripped out, some emails have been redacted, some email addresses are malformed or missing, and many messages are duplicated [3, 9, 10, 17]. At least two groups have cleaned the dataset by removing duplicate messages, standardizing email addresses, and providing the data in the form of a database [6, 17]. Our experiments use the dataset from USC [17].

We use existing email traces generated by the Bagle.a, Bagle.f, Bagle.g, BubbleBoy, MyDoom.b, MyDoom.m, MyDoom.u, Netsky.d, and Sobig.f viruses [13]. Each virus infects a virtual machine and the emails it sends are recorded. The virtual machines are seeded with an actual user's address book so that the viruses can exhibit realistic sending behaviors. Two of these viruses are particularly interesting: BubbleBoy does not require an attachment to propagate and uses Outlook rather than its own SMTP engine, and MyDoom.m makes use of highly polymorphic message bodies and subject lines.

3.2 Experimental Procedure

Our primary experiment compares the models' performance on the nine viruses in our dataset for varying numbers of training users and model topics. For each experiment, we select a training set of the appropriate number of users and a test set of one user. We choose the users uniformly at random without replacement, except that we require the test user to have sent at least 100 emails.[4] For each virus, we simulate an infection by injecting 100 emails from that virus' trace into the test user's email stream.

LDA and the mixture models classify generatively by choosing the best fit between normal and virus models, while the SVM is discriminative and produces a classification without modeling the classes themselves. The normal user models are trained on the randomly selected user training set and the virus models are trained on the traces from the eight other viruses; the SVM's training set includes both sets with appropriate labels. We use default settings for the SVM [7].

We train the models and then hold out the first 50 emails from the test user and (for LDA and the per-user mixture model) update the user-specific parameters based on up to 50 held-out emails. We then test on the 51st onward. This allows us to measure the performance of the algorithms against the number of emails seen from a new user. Each setting is run five times with different training and test sets for each number of held out emails from 1 to 50.[5]

We assume that our system has access to every email

[4] This excludes 40 of the 151 Enron employees in the dataset.

[5] We train on the most recent emails in the held-out set to avoid the introduction of gaps in a user's stream of messages.

sent from a network of end-user machines and is able to accurately determine which user or machine sent each message. Some viruses attempt to bypass an organization's outgoing email servers (e.g., by including their own SMTP engines), however, transparent SMTP redirection or stateful packet inspection by a firewall can be used to enforce our assumption.

3.3 Results

We show graphs of the learners' performance on five viruses in Figure 2 and final numbers for all nine viruses in Table 2. A false positive (FP) is a normal email misclassified as a virus and a false negative (FN) is a virus email misclassified as normal. The FP graphs for all nine viruses show very similar trends and provide little information that is not available in Table 2, so we only show one FP graph. Behavior on FNs is more varied, and we show the five most interesting viruses. Bagle.g, MyDoom.u, and Netsky.d have FN graphs nearly identical to Bagle.a. MyDoom.b has a graph similar to Bagle.f, but for MyDoom.b, LDA and GMM do 6–7% better and the SVM does much worse at 89% FNs.

The graphs in Figure 2 are averaged over five runs each of three and ten topics for LDA, the GMM, and the PMM. LDA and the GMM both have around 7% fewer FPs but 4% more FNs with ten topics, and the PMM has 10% fewer FPs and 5% more FNs with ten topics.

The SVM consistently has the lowest false positive rate. Although LDA generally improves by a few percent with more per-user data, the FP rates of GMM and LDA are nearly identical. The interesting part of Figure 2a is the PMM curve: it begins with a near-100% FP rate as it overfits to the first few emails (classifying everything as a virus), but with more data it improves and approaches the GMM and LDA.

The false negative rates are more interesting, as we see varying performance across viruses. The SVM has the *highest* FN rates in all graphs, while the GMM and LDA again have similar performance. The PMM starts off with very low FNs due to overfitting the user model, but as it generalizes it starts to misclassify virus emails.

The FN graph for MyDoom.m stands out, with terrible performance by all learners. MyDoom.m is a polymorphic virus that exhibits large variation in both subject line and body, which makes it difficult to classify. Again the PMM initially classifies everything as a virus, but its FNs increase quickly. No learner correctly classifies more than 26% of MyDoom.m emails in the end.

Table 2 shows the FP and FN rates for our experiments. While we see high variation in FN rates across viruses, the FP rates remain relatively constant. This is not surprising: between any two experiments, the only difference relevant to a FP is a substitution of one of eight training viruses. FNs, however, show more variability because the email being classified is different.

Three related observations on our results merit investigation: the PMM does not tend to perform as well as the GMM even after training on 50 user emails, LDA does not show significant improvement over the 50 emails, and the GMM performs almost equivalently to LDA. We would likely see better per-user performance for the PMM and possibly LDA with more hold-out emails, since the PMM clearly improves up to the 50th email. The size of the dataset, however, hampers our ability to do this. It is also possible that our feature set does not encode the type of information that would allow the per-user models to gain significant advantage over global models. We conjecture that closer attention to feature selection and increased numbers of hold-out emails would allow LDA to consistently outperform the GMM.

	False Negatives				False Positives			
	LDA	GMM	PMM	SVM	LDA	GMM	PMM	SVM
Bagle.a	2%	2%	2%	22%	18%	17%	22%	6%
Bagle.f	33%	28%	14%	38%	16%	15%	18%	5%
Bagle.g	2%	2%	3%	23%	20%	16%	18%	5%
BubbleBoy	3%	7%	21%	23%	16%	15%	21%	5%
MyDoom.b	26%	22%	9%	89%	14%	12%	12%	3%
MyDoom.m	81%	83%	74%	95%	11%	10%	11%	3%
MyDoom.u	3%	3%	2%	22%	16%	15%	23%	5%
Netsky.d	2%	2%	2%	22%	17%	16%	22%	5%
Sobig.f	2%	2%	4%	22%	18%	17%	22%	5%

Table 2: False positives/false negatives.

4. CONCLUSION

In this paper, we present an LDA model that combines global and per-user components for email virus detection. Our experimental results show that this system immediately provides acceptable performance on new users even with our conservative feature set and in the long run remains competitive with more specific per-user models.

These results highlight several interesting directions for future work, including an in-depth feature selection for this setting and incorporating more per-user training information into the models. These results also show that, despite room for improvement in per-user specialization, LDA performs competitively with a simple support vector machine.

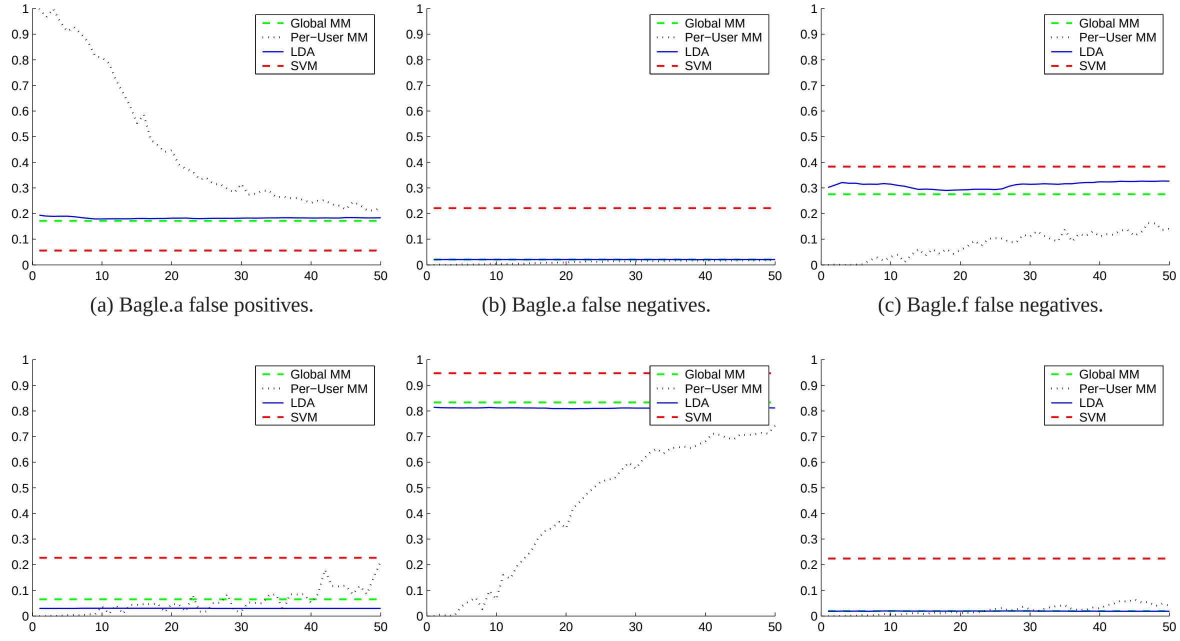
Another avenue for future work is to extend the LDA model further. An interesting potential extension would give each user different β parameters as in the PMM, but with a global prior parameter analogous to α for θ. This would give the model more freedom to adapt to each user's behavior.

The combination of per-user and global models has the potential to react quickly to global changes while providing superior long-term performance. This work describes an approach that shows promise for increasing the ability of machine learning systems to defend users from novel viruses.

ACKNOWLEDGMENTS

Thanks to Steve Martin and Anil Sewani for the use of their email handling code and virus data. We also gratefully acknowledge the suggestions from anonymous reviewers.

This work was supported in part by DETER, with support



(a) Bagle.a false positives. (b) Bagle.a false negatives. (c) Bagle.f false negatives.

(d) BubbleBoy false negatives. (e) MyDoom.m false negatives. (f) Sobig.f false negatives.

Figure 2: In these graphs, the horizontal axis gives the number of training emails seen from the test user. False positive rates (normal emails classified as virus) for all algorithms are nearly identical for the nine viruses; we show Bagle.a in (a). The false negatives (virus emails classified as non-virus) show more interesting behavior. We show graphs for five different viruses in (b) through (f).

from the HSARPA and NSF grant ANI-0335241, and in part by TRUST, with support from NSF grant CCF-0424422 and the following organizations: Cisco, ESCHER, HP, IBM, Intel, Microsoft, ORNL, Qualcomm, Sun, and Symantec. The views expressed here are solely those of the authors and do not necessarily reflect the views of the supporting companies, the funding agencies or any agency of the U.S. government.

REFERENCES

[1] Bickel, P. J., and Doksum, K. A. *Mathematical Statistics: Basic Ideas and Selected Topics*, 2nd ed., vol. 1. Prentice-Hall, Inc., 2001.

[2] Blei, D. M., Ng, A. Y., and Jordan, M. I. Latent Dirichlet allocation. *Journal of Machine Learning Research (JMLR) 3* (2003), 993–1022.

[3] Chapanond, A., Krishnamoorthy, M. S., and Yener, B. Graph theoretic and spectral analysis of Enron email data. *Computational & Mathematical Organization Theory 11*, 3 (Oct. 2005), 265–281.

[4] Cohen, W. W. Enron email dataset. `http://www.cs.cmu.edu/~enron/`.

[5] Eskin, E., Arnold, A., Prerau, M., Portnoy, L., and Stolfo, S. J. A geometric framework for unsupervised anomaly detection: Detecting intrusions in unlabeled data. In *Proceedings of the Workshop on Data Mining for Security Applications* (2002), Kluwer.

[6] Fiore, A., and Heer, J. UC Berkeley Enron email analysis. `http://bailando.sims.berkeley.edu/enron_email.html`.

[7] Joachims, T. Making large-scale SVM learning practical. In *Advances in Kernel Methods — Support Vector Learning*, B. Schölkopf, C. J. C. Burges, and A. J. Smola, Eds. MIT Press, 1999.

[8] Jordan, M. I. Graphical models. *Statistical Science 19*, 1 (2004), 140–155.

[9] Keila, P. S., and Skillicorn, D. B. Structure in the Enron email dataset. *Computational & Mathematical Organization Theory 11*, 3 (Oct. 2005), 183–199.

[10] Klimt, B., and Yang, Y. The Enron corpus: A new dataset for email classification research. *Lecture Notes in Computer Science 3201* (2004), 217–226. Appeared in ECML'04.

[11] Lazarevic, A., Ertoz, L., Kumar, V., Ozgur, A., and Srivastava, J. A comparative study of anomaly detection schemes in network intrusion detection. In *Proceedings of the SIAM International Conference on Data Mining* (2003).

[12] Martin, S., Sewani, A., Nelson, B., Chen, K., and Joseph, A. D. Analyzing behavioral features for email classification. In *Proceedings of the Conference on Email and Anti-Spam (CEAS)* (2005). `http://www.ceas.cc/papers-2005/123.pdf`.

[13] Martin, S. L. Learning on email behavior to detect novel worm infections. Master's thesis, University of California at Berkeley, 2005.

[14] Meyer, T. A., and Whateley, B. SpamBayes: Effective open-source, Bayesian based, email classification system. In *Proceedings of the Conference on Email and Anti-Spam (CEAS)* (2004).

[15] Robinson, G. A statistical approach to the spam problem. *Linux Journal 2003*, 107 (2003).

[16] Seltzer, L. J. Security watch: MyDoom reappears. *Security Watch Newsletter* (Feb. 2005). `http://www.pcmag.com/article2/0,1759,1767805,00.asp`.

[17] Shetty, J., and Adibi, J. The Enron email dataset: Database schema and brief statistical report. `http://www.isi.edu/~adibi/Enron/Enron.htm`.

[18] Sophos Corporation. Top 10 viruses reported to Sophos in 2005. Online, Dec. 2005. `http://www.sophos.com/virusinfo/topten/200512summary.html`.

[19] Stolfo, S. J., Li, W.-J., Hershkop, S., Wang, K., Hu, C.-W., and Nimeskern, O. Detecting viral propagations using email behavior profiles. *ACM Transactions on Internet Technology (TOIT)* (2004).