

**Semi-Supervised  
Learning on Email  
Characteristics  
for Novel Worm  
Detection**

**Anil Sewani (anil@cs)**

**and**

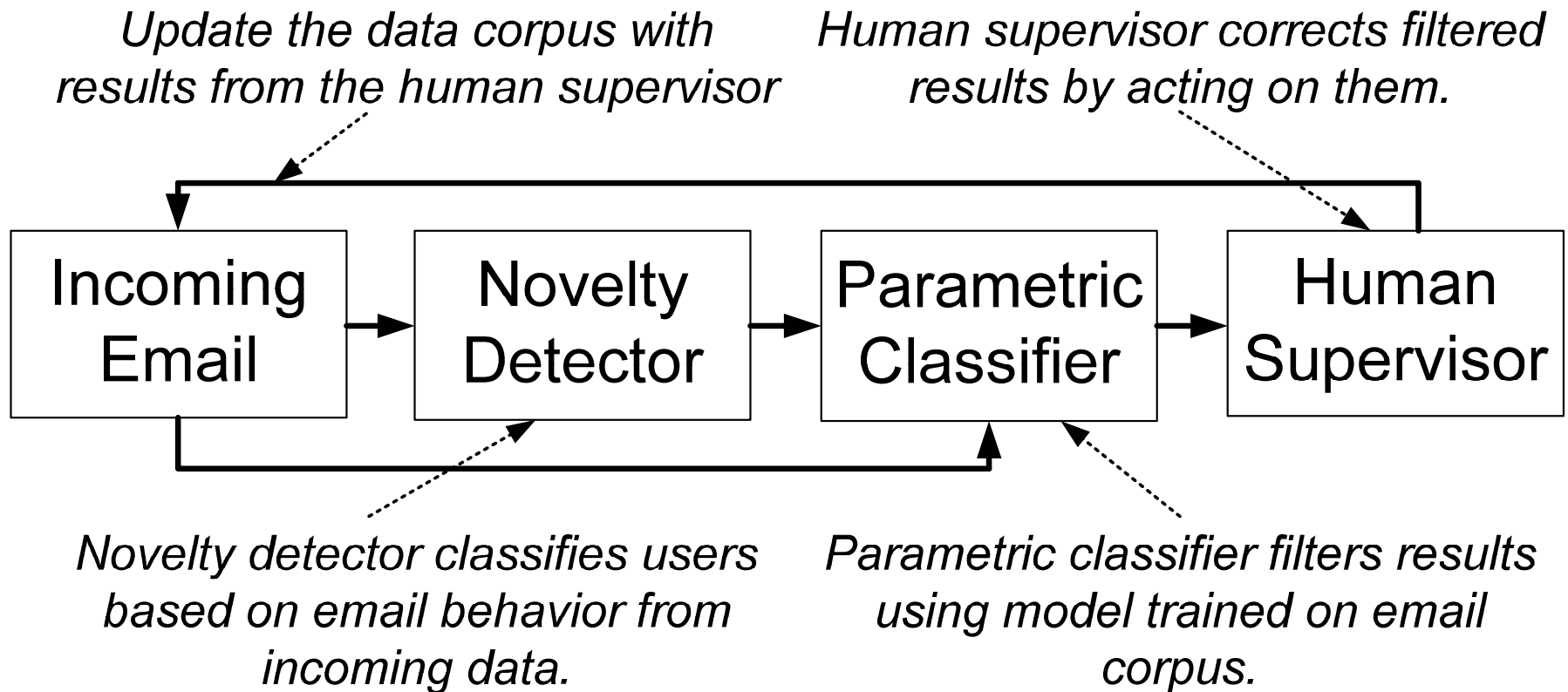
**Steve Martin (steve0@cs)**

# The Problem

- Worms are a huge global security problem.
  - Example: recent MyDoom epidemic.
  - Modern worm propagation is incredibly quick.
- Current anti-virus solutions use a completely reactive model.
  - Infection must occur before countermeasures can be developed and deployed.
- Human element slows reaction times.
  - Generate/install antivirus sigs, update firewalls, etc.
  - By then epidemic is typically in full swing.

# Learning for Worm Detection

- Learn on a user's email behavior to decide whether or not they are infected with a worm.



# Two-Layer Classification

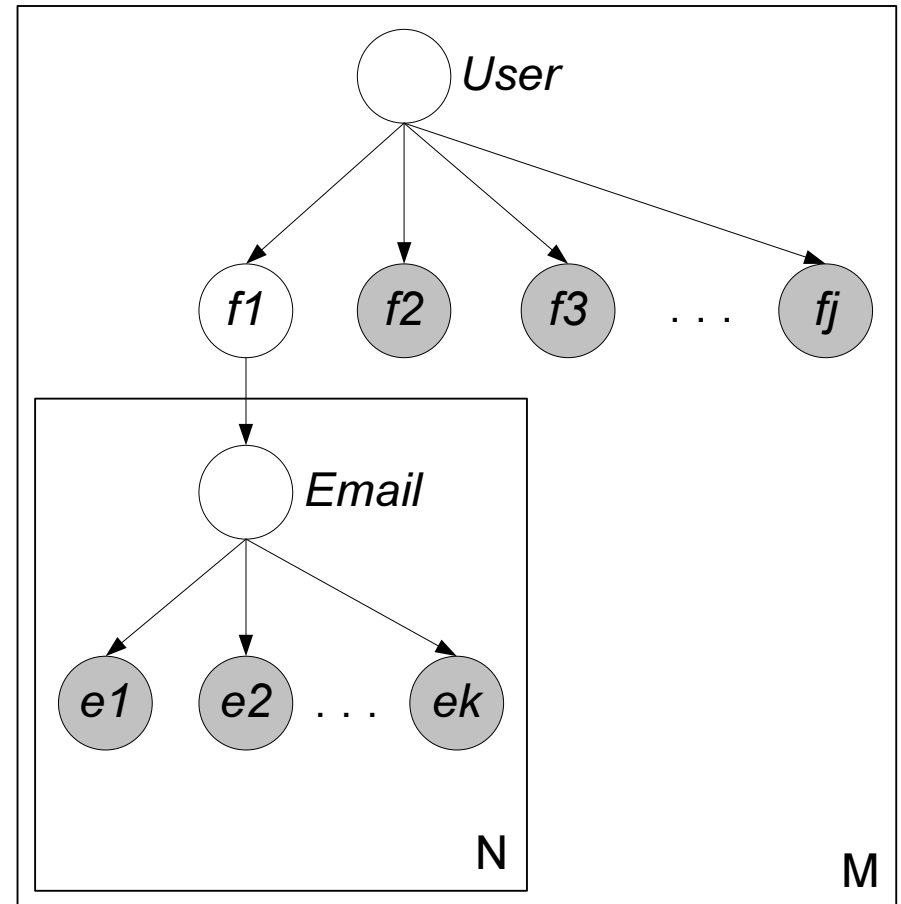
- Novelty detection is not enough by itself.
  - Must avoid false negatives; to do this, make the novelty detector model very sensitive.
  - This introduces false positives.
- Idea: filter anomalies from novelty detector
  - Leverage knowledge gained by human feedback to improve results on new data.
- Why use a generative model?
  - Allows us to fit specific distributions to feature data.
  - Allows for efficient training space-wise.
  - Easy to add new features to the model.

# Classification Process

1. Novelty detection on user email behavior
  - Modeled by false positive probability for this project.
2. Anomalies filtered through parametric classifier.
3. Filtered points go to human user for correction.
  - New corrected classifications added to data corpus.
4. Parametric model retrains over semi-supervised data using the EM algorithm.
  - Done in batches for efficiency.
  - Over time, model gains accuracy.

# Classifier Graphical Model

- $f_1 \dots f_j$  are *per-user* features that examine behavior.
- $e_1 \dots e_j$  are *per-email* features of a particular user's email.
- There are  $M$  independent users, each of which have sent  $N$  independent emails.
- Assume features are independent given the class.



# Data Sources

- Ran our own mail server and instrumented it to calculate feature data on live email.
- Conducted a user study to collect data and learn distributions of our features.
- Leveraged personal sent-mail folders for testing.
- Collected virus data from a variety of real infections using virtual machines.

# Example Email Features

## Per-Email Features

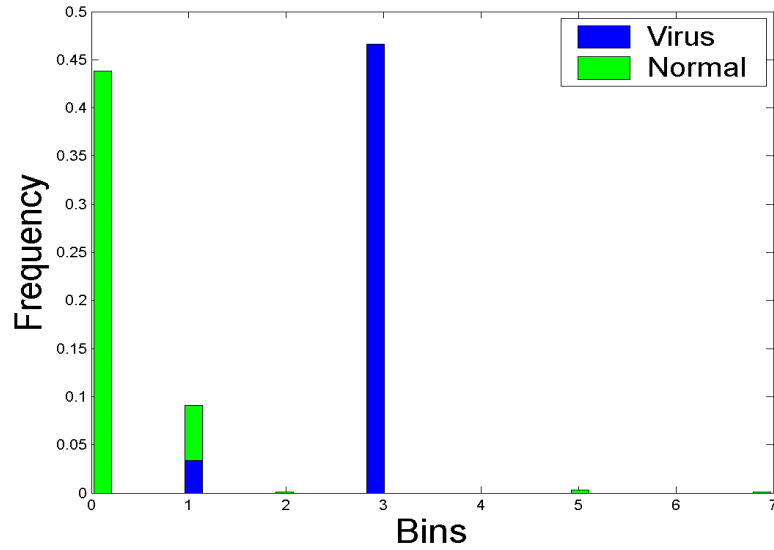
- Average word length
- Chars in subject
- Email sending freq.
- HTML in email
- Images in email
- Links in email
- Scripts in email
- Num. of attachments
- No. of words in body
- No. of words in subject
- Attachment type

## Per-User Features

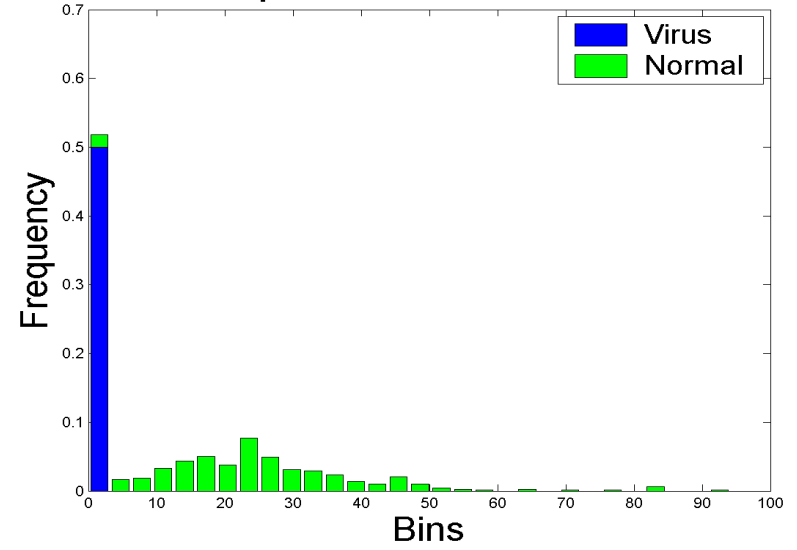
- Mean chars in subject
- Mean words in body
- Number of addresses sent to over last n emails.
- Ratio of emails sent with to without attachments
- Var. of attachment size
- Var. of chars in subject
- Var. of words in body

# Feature Distributions

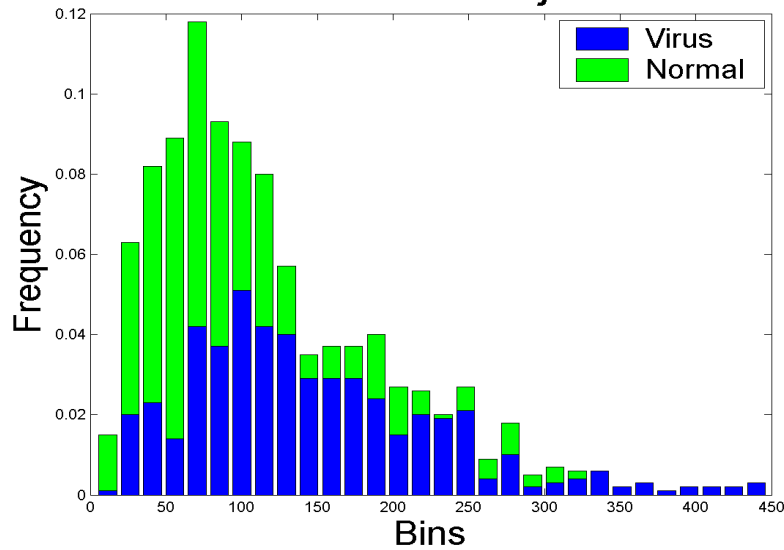
## NumAttachments



## FreqEmailSentInWindow

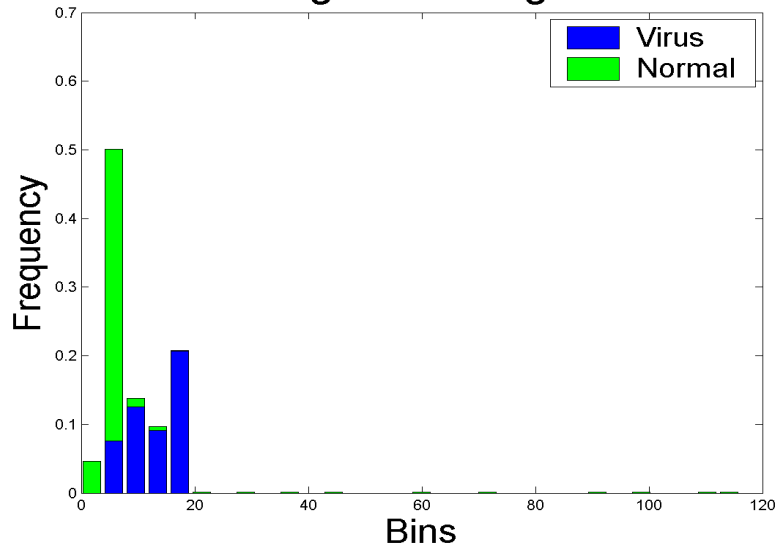


## VarCharInSubject

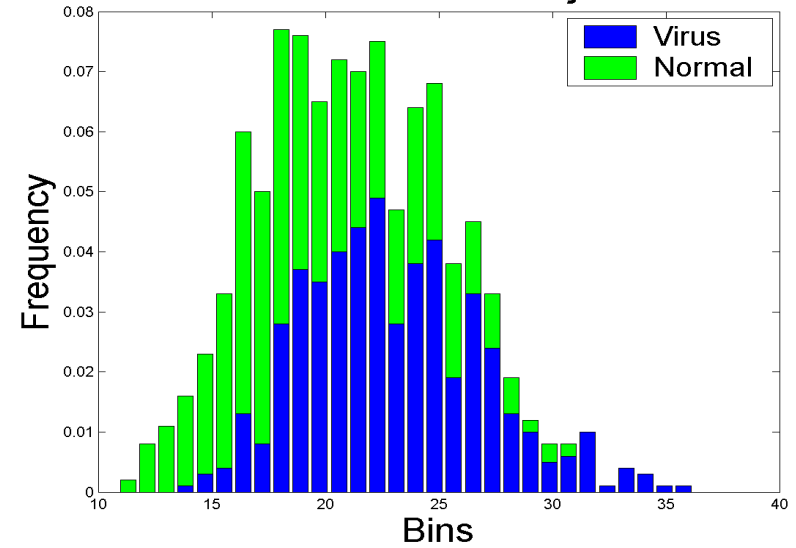


# Feature Distributions, II

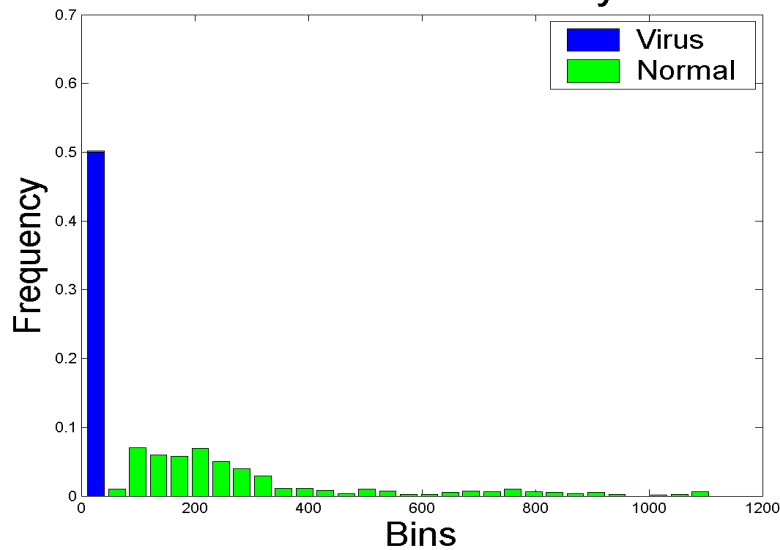
## AvgWordLength



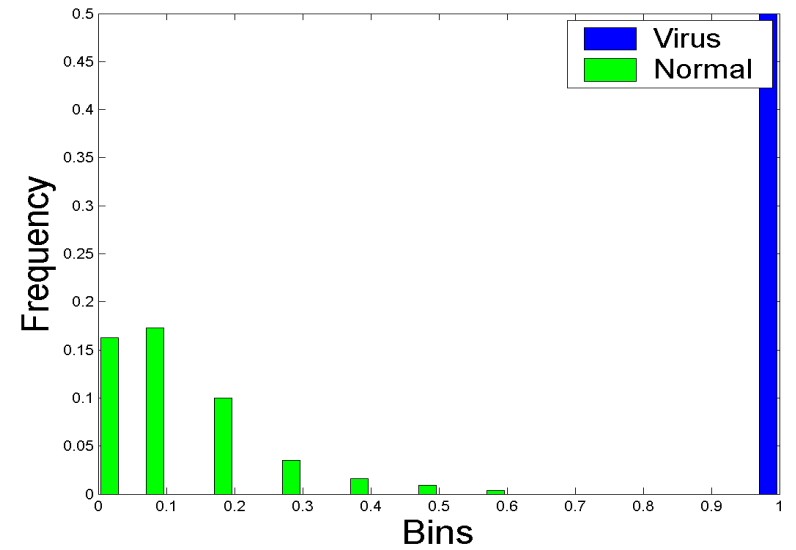
## MeanCharInSubject



## MeanWordsInBody



## RatioAttach



# How We Tested

- Gathered data from three real viruses: Klez, Loveletter.C, and Bagle.A.
  - Klez and Bagle are polymorphic, Loveletter is not.
- Constructed training set of real email traffic artificially 'infected' with two viruses.
- Tested on sets of email traffic infected with the third virus as the 'novel' virus.
- Rotated through all combinations of virus data.

# Results

**Table 1.** Parametric Filter Classification Results

| <i>Worm Name</i> | <i>Total Emails</i> | <i># Worm Emails</i> | <i># Clean Emails</i> | <i>False Positives</i> | <i>False Negatives</i> | <i>Correctly Classified</i> |
|------------------|---------------------|----------------------|-----------------------|------------------------|------------------------|-----------------------------|
| Bagle            | 1090                | 789                  | 301                   | 6<br>(0.76%)           | 6<br>(1.99%)           | 1078<br>(98.90%)            |
| Klez             | 1090                | 789                  | 301                   | 4<br>(0.50%)           | 15<br>(4.98%)          | 1071<br>(98.26%)            |
| LoveLetter       | 1090                | 787                  | 303                   | 9<br>(1.14%)           | 5<br>(1.65%)           | 1076<br>(98.72%)            |

# Future Work

- Examine additional features and refine model
  - Idea: include novelty detector as a node
  - Incorporate more dependencies
- Further testing with live data
- Consider other methods of filtering.
  - Our model is parametric; non-parametric methods may work better (decision trees, etc).
- Implement our algorithms into a deployable system.