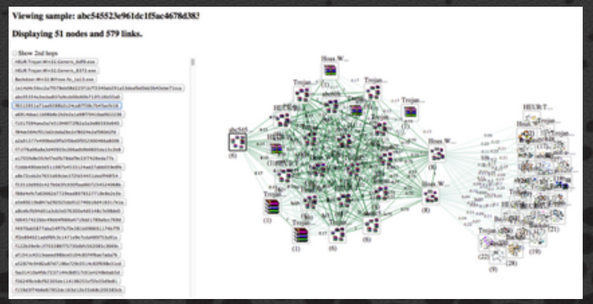
Graphic Content Ahead: Towards Automated Scalable Analysis of Graphical Images Embedded In Malware

Alex Long, Joshua Saxe, Robert Gove
Invincea Labs

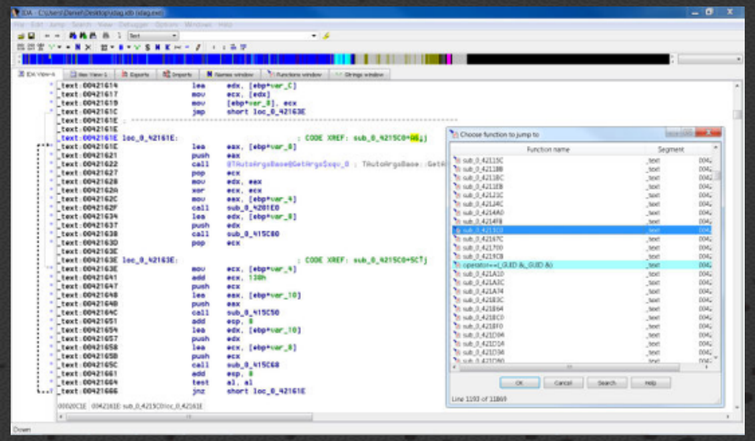
Talk Outline

1. The Status Quo of Malware Analysis
2. Hard Problems The Industry is Dealing With
3. Our Approach
4. Two Research Experiments
    1. Detecting and Visualizing Image-Sharing Relationships (Live Demo)
    2. Automatically Classifying Images by Their Semantics



Status Quo of Malware Analysis

Malware analysis treats malware as just a set of instructions



Analysis typically consists of analyzing the disassembled code and/or observing the malware's runtime behavior

Problems Facing Analysts Today

Malware could be packed or use VM detection tactics

Manual analysis of each sample is intractable given huge numbers of polymorphic variants

Expanding the Frame

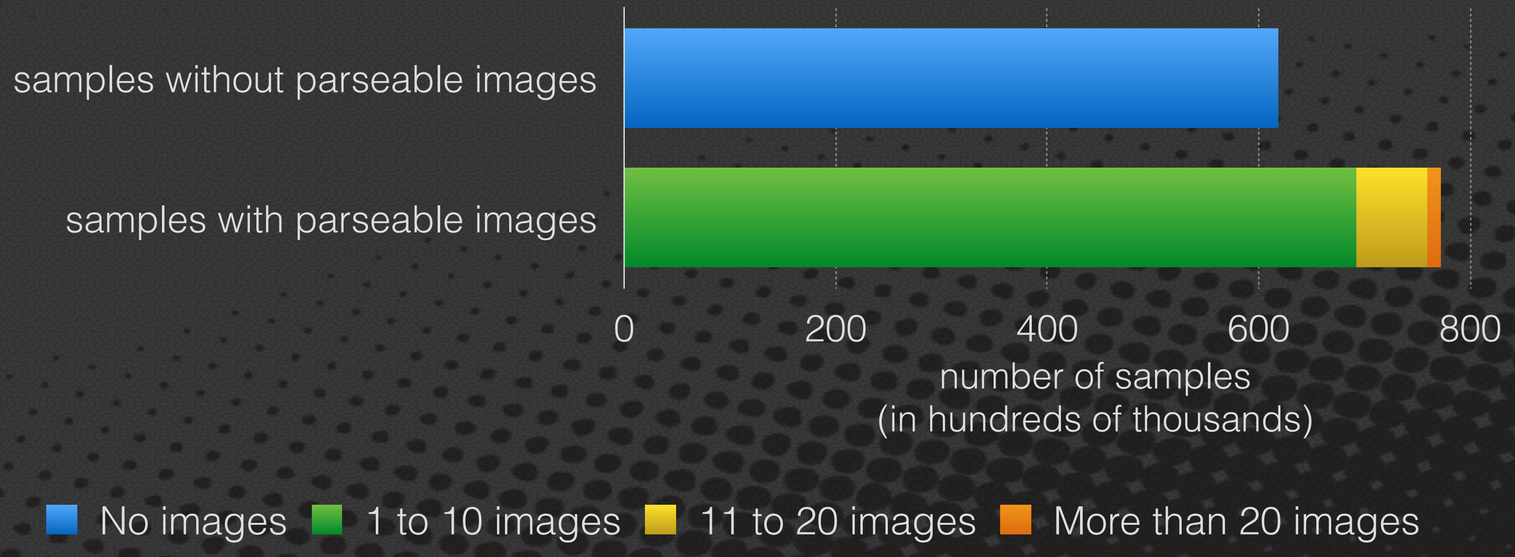
Malware is not just code,
it's also

- natural language
- documents
- audio
- video
- images

The Setting

- **Problem:** Graphical assets are an untapped resource in the malware analysis space; image analysis done manually.

Of a collection of 2 million malware samples provided to us by DARPA, over half had at least one image embedded.



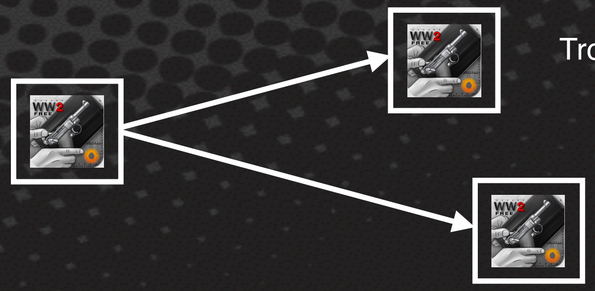
How Image Analysis is Useful

A packed Trojan still needs an attractive icon to lure a user into executing it

Images can hint at the ways in which attackers are tricking the user and the purpose of a binary artifact.

Game-related

By exploring the malware's "social network" through shared rare images, you can learn about an otherwise hard-to-reverse sample.



What Automation Makes Possible

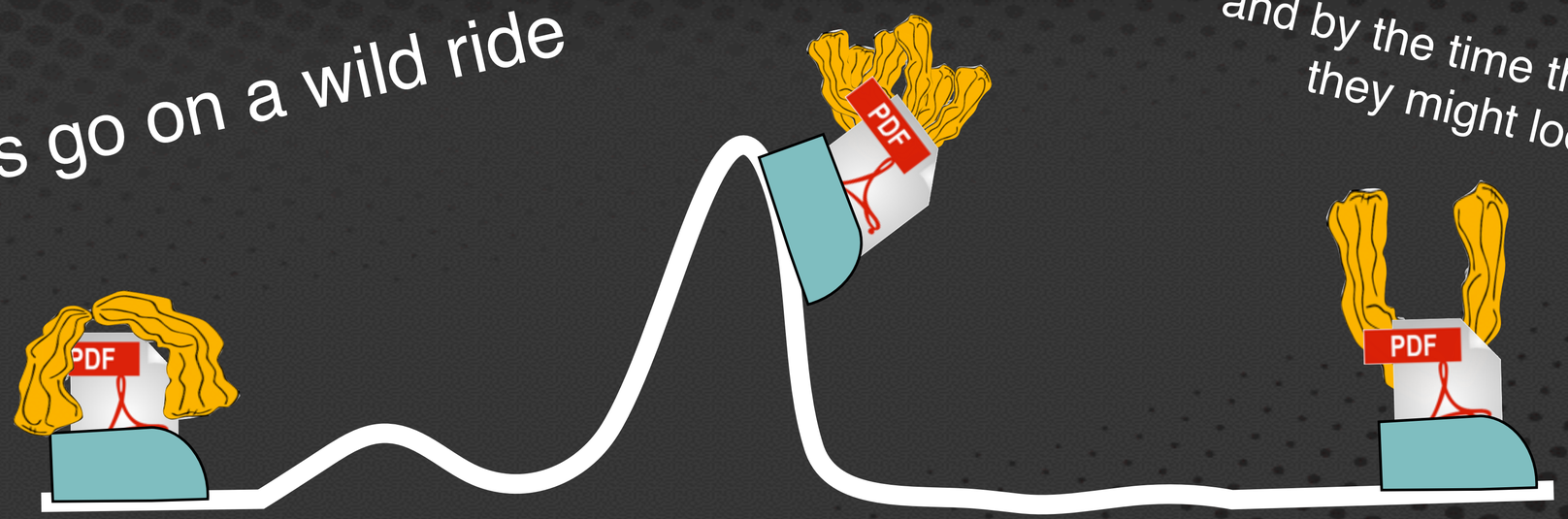
Survey of the threat landscape

We're seeing an up-tick in malware masquerading as PDFs, let's alert our employees to be on the lookout

Quick analysis of new samples

Our system has found a previously analyzed sample that shares an image with this email attachment

Why Not Just Compare Hashes?



Hash comparison will fail if the image was

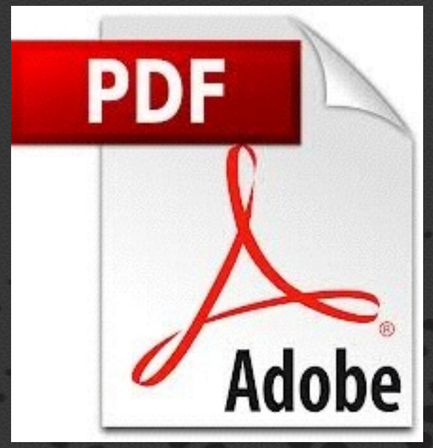
compressed

copied and pasted

modified deliberately

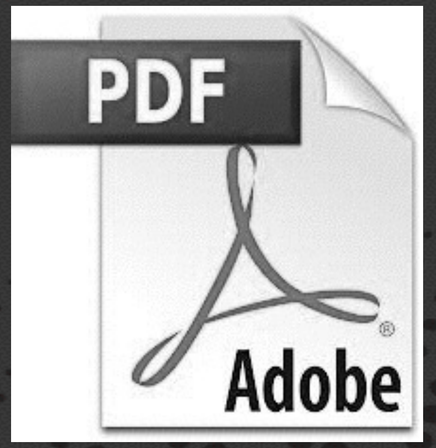
Average Hash

1. Take an image



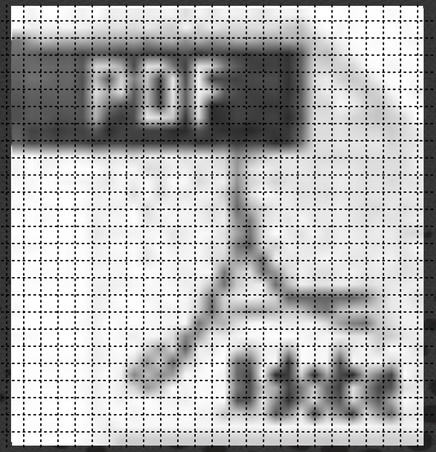
Average Hash

1. Take an image
2. Reduce to grayscale



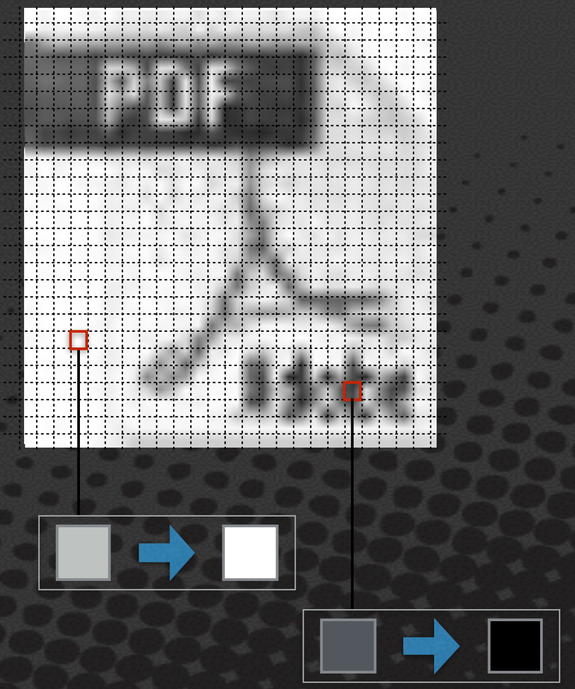
Average Hash

1. Take an image
2. Reduce to grayscale
3. Stretch/shrink to 32x32



Average Hash

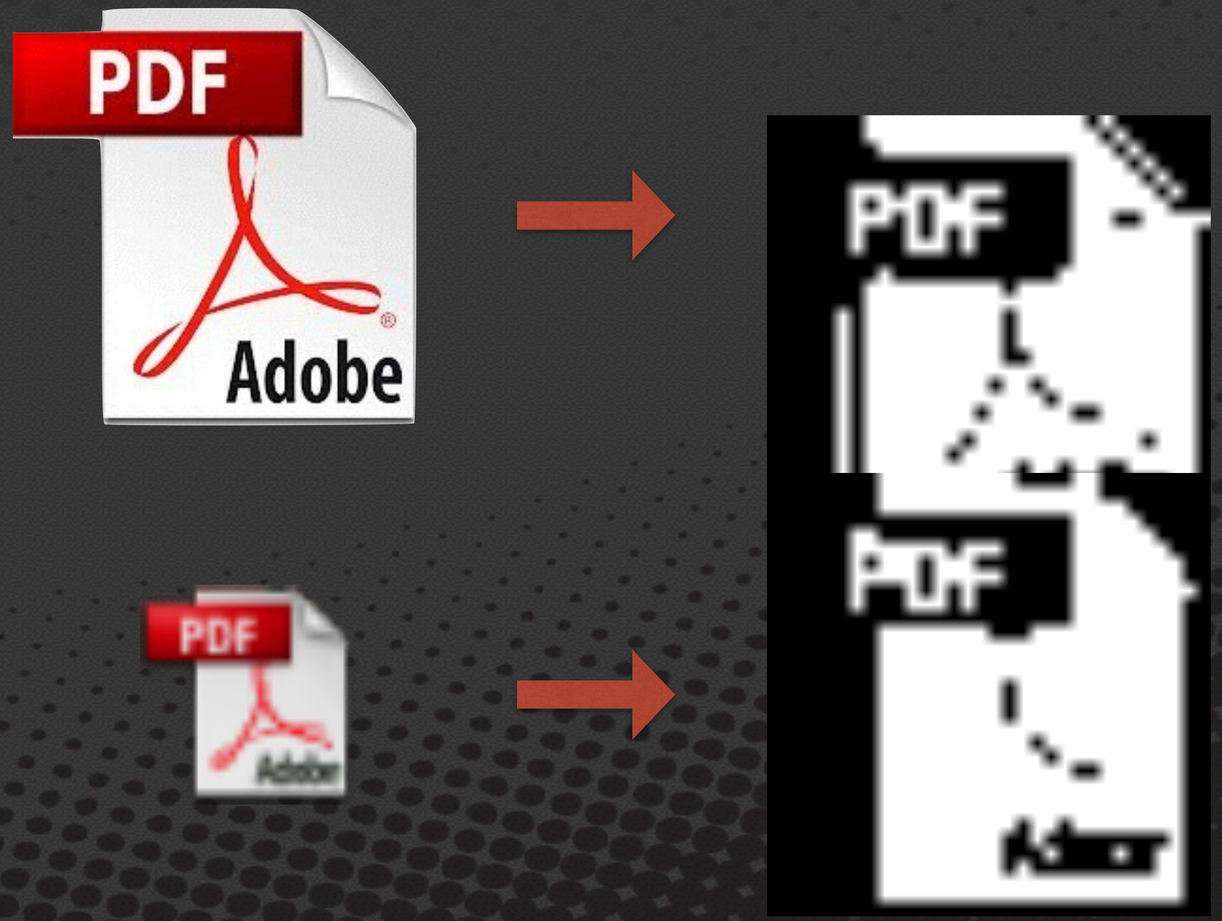
1. Take an image
2. Reduce to grayscale
3. Stretch/shrink to 32x32
4. Convert to high contrast
   a. Get average value of pixels
   b. For each pixel,
      if above average, set to 255
      if below average, set to 0



Average Hash

1. Take an image
2. Reduce to grayscale
3. Stretch/shrink to 32x32
4. Convert to binary vector
   a. Get average value of pixels
   b. For each pixel,
      if above average, set to 255
      if below average, set to 0

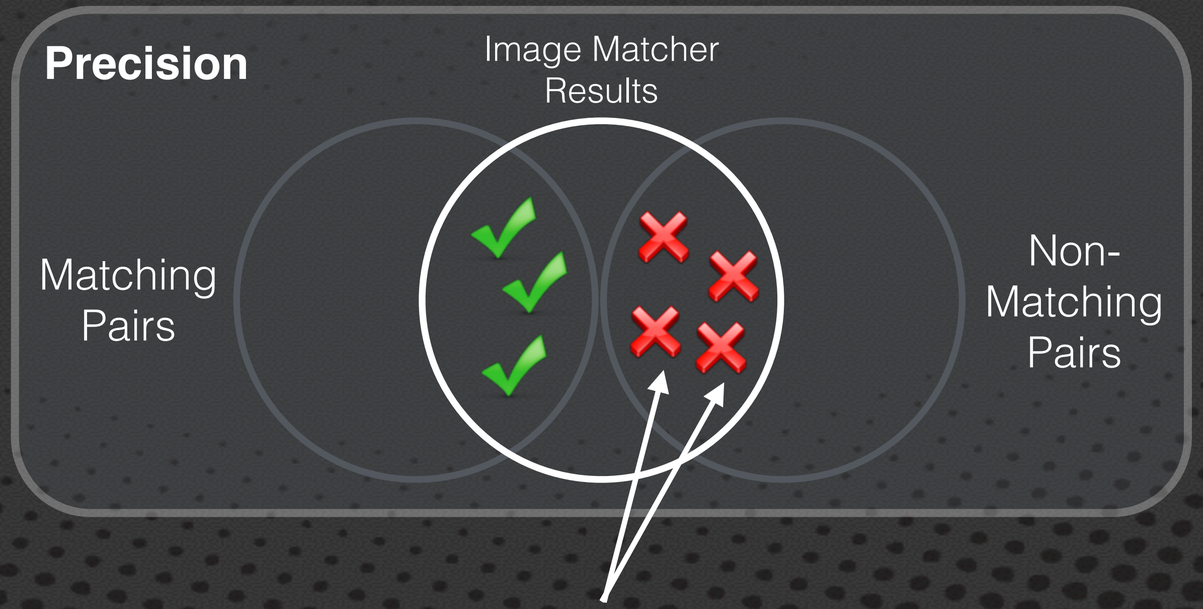
Average Hash



Statistics Refresher

Precision:

What percent of the pairs matched by the system actually had similar images?

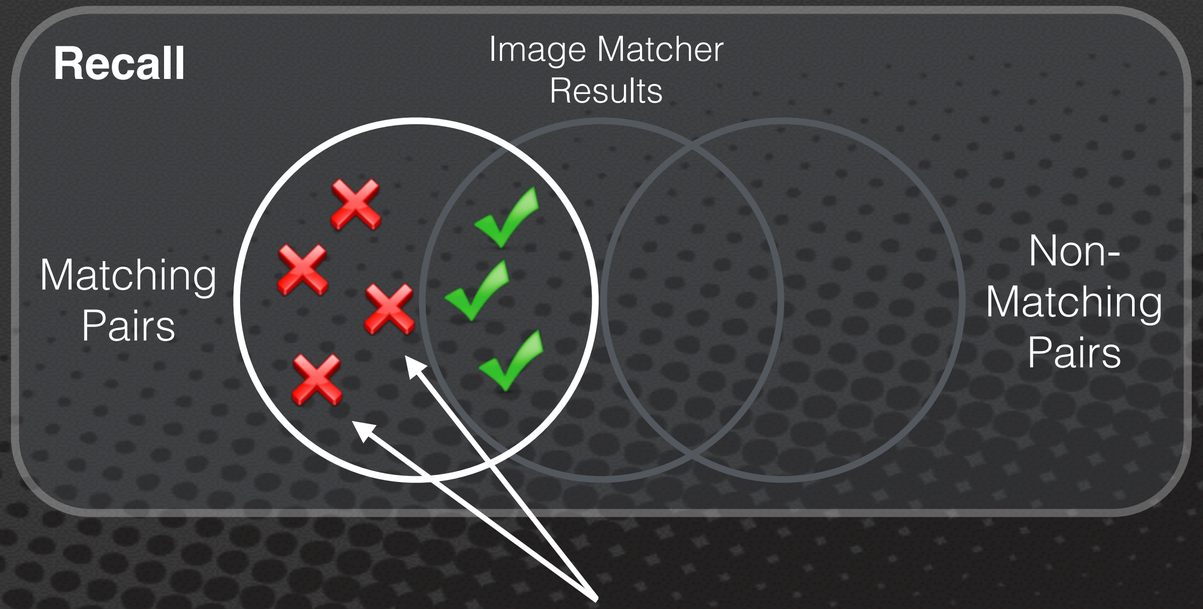


False positives

Our system says a pair of images are similar that actually are different

Recall:

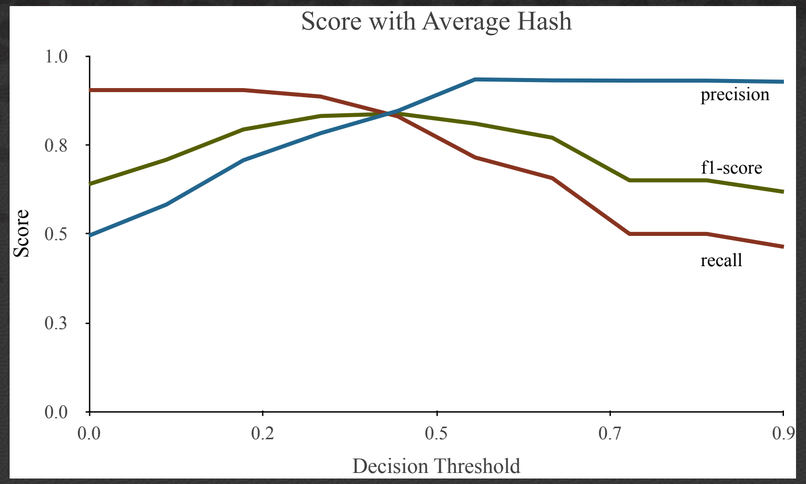
What percent of all matching pairs did the system find?

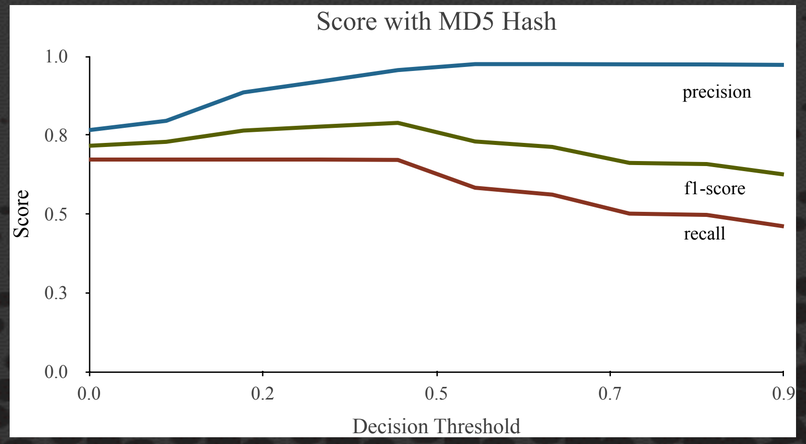


False negatives

Our system fails to say a pair of images are similar when they actually are

Average Hash vs MD5





- 200 hand clustered samples
- MD5 wins out in precision
- Average hash wins in recall
- Humans can easily detect and ignore images that don't match (false positives)
- False negatives invisible to user and lost forever
- We'd prefer to have more false positives (lower precision) in order to have fewer false negatives (higher recall)

Comparing Sets of Images

=

?

Comparing Sets of Images

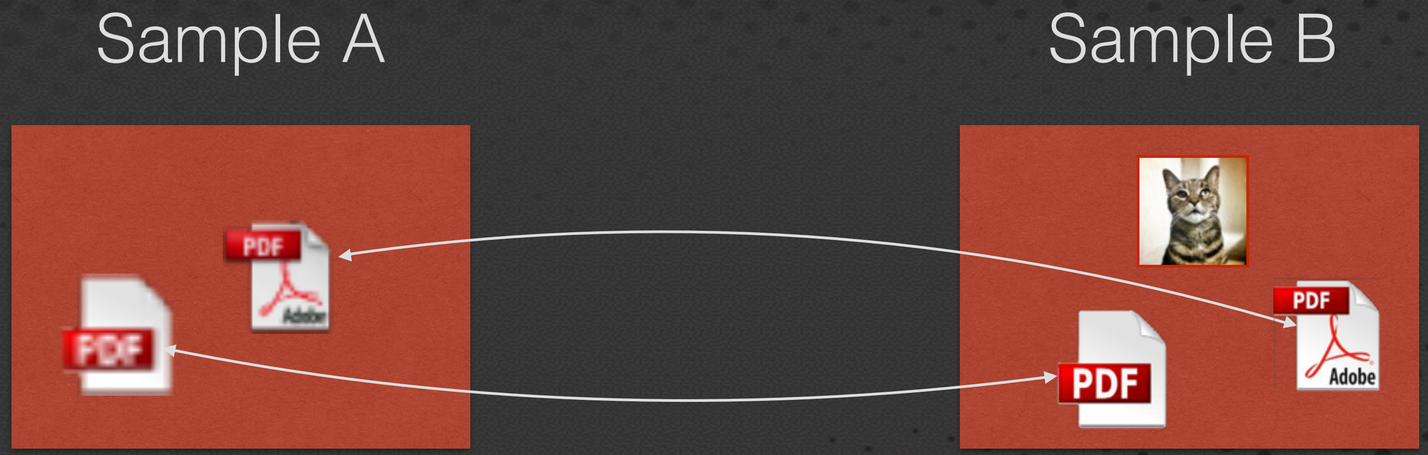
Sample A

=

Sample B

?

Comparing Sets of Images



$$\frac{\text{Number of matching pairs (2)}}{\text{Number of possible matching pairs (3)}} = 0.66 \text{ similarity}$$

Demo!

Live Demo

Automatic Classification of Images

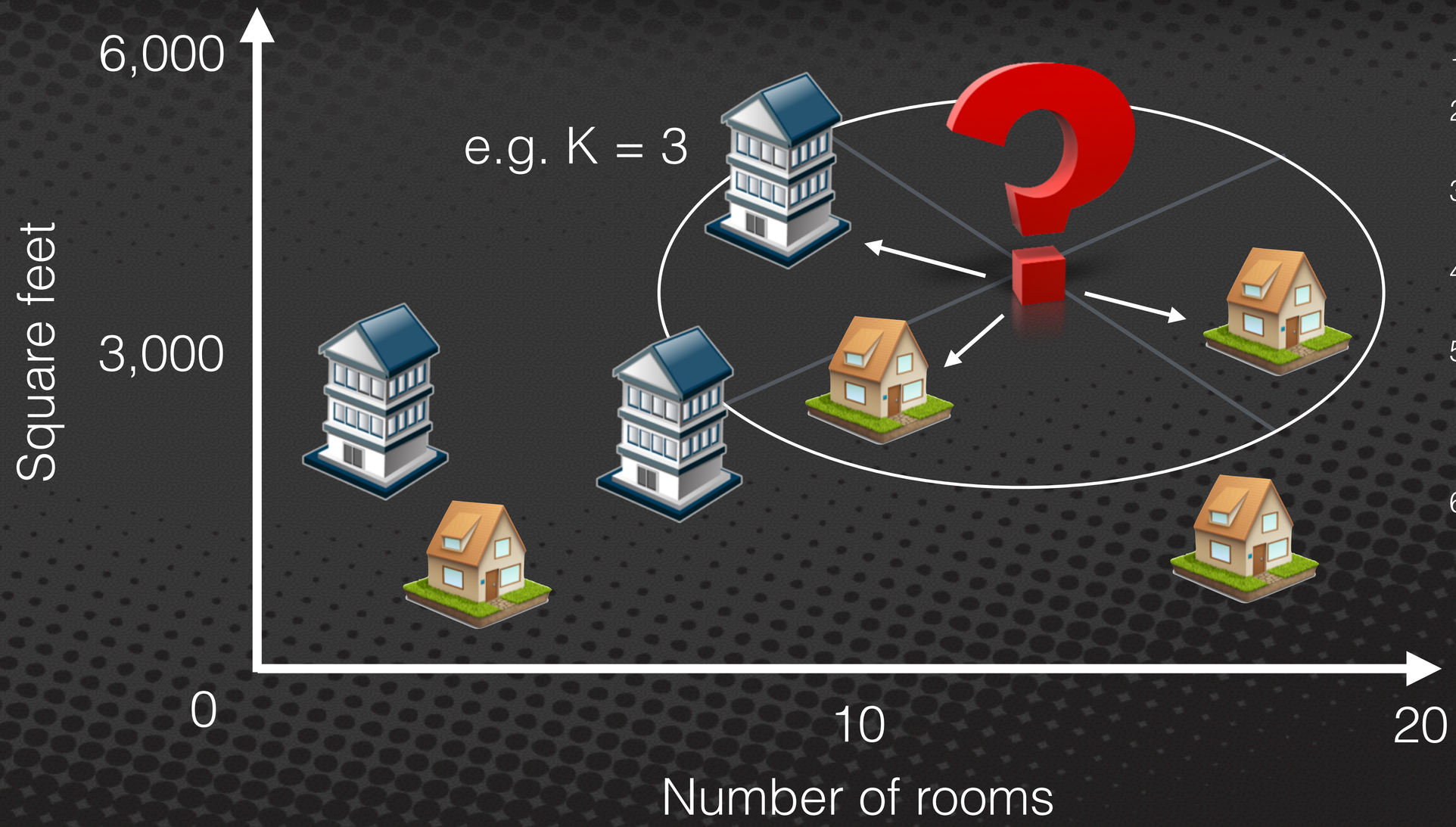
Browser

Anti-virus

Document

- Reveal purpose of malware
- Survey threat landscape
- Assign risk factor

Machine Learning 101: K Nearest Neighbors



K Nearest Neighbors

1. Plot points on graph
2. Assign classes to points (e.g. house or apartment)
3. Plot point on graph with unknown class
4. Pick a number K as appropriate for your data
5. Get the top-K nearest neighbors (AKA closest points) to the point with the unknown class
6. Classify by majority vote (e.g. if the 3 nearest points are 2 houses and 1 apartment, unknown class is house)

Classification by Color Histogram

Image with unknown class	Predicted class	K Nearest Neighbors
malware_icons/adobe flash.png	adobe flash	
malware_icons/antivirus.png	windows media player	
malware_icons/antivirus2.png	windows media player	
malware_icons/jpg file icon.png	jpg file icon	
malware_icons/real player icon1.png	windows folder	
malware_icons/unknown.png	vmware	
malware_icons/windows installer.png	windows drive icon	
malware_icons/windows installer1.png	windows installer	
malware_icons/windows installer2.png	windows media player	
malware_icons/windows installer3.png	windows internet icon	
malware_icons/windows network icon.png	antivirus	

invincea™

Conclusion

Graphical content of malware is a significantly under-utilized signal in malware analysis

Automation and visualization make this “human” signal accessible at a large scale

Alex Long alex.long@invincea.com