Using Biometric and Spatial Simulations to train Machine Learning Systems in Identification of Anomalous Behavior for Public Safety Applications

Kaushik Kasi, Dublin High School

Mentor: Dr. Victor Castillo, LLNL

Question/Problem

- What kind of systematic approach can be used to improve public security?
- How can anomaly detection through Machine Learning be used for public security applications?
- What data sources can be used to solve this problem, and how can they be used?

Hypothesis

- Given insightful data, applying Machine Learning Techniques can be an effective approach towards public safety.
- A more systematic approach using tangible data towards public safety can yield better results than subjective screening.
- The insightful data can come from biometric and spatial sources.

Purpose

- Public safety, security and health is an important concern to all
  - Billions of dollars is spent every year on public health, security, asset protection
- Most of these methods ultimately boil down to human monitoring in some way
  - Subjective to each person’s personal prejudices
  - Humans lose attention and aren’t good at piecing data together (ACLU)
  - Not based on tangible data, or handled systematically (TSA GAO 2012 Report)

End Goals

- Machine Learning systems should be designed with purpose of being usable for a real-world application
- Multiple ML systems should take data from multitude of sources, similar to data that would be available in a real-world use-case
- Data should be preprocessed, and ML techniques for anomaly detection should be applied effectively
- Performance should be measured in a quantifiable way

Approach

- Different sources of data used as surrogates for real-world data/situations
- Thermal images will be obtained from public dataset
  - Machine Learning system designed to identify anomalous samples within the dataset
- An agent-based simulation will be designed in order to simulate how people move in a public system
  - Anomalous agents embedded within the simulation
  - ML system designed to identify anomalous agents based on collected metrics
  - Persistics algorithm designed about collecting this info in a real-life use-case

Identifying Anomalies Biometrically Using Thermal Infrared Images

- Thermal Images can show temperature and distribution of heat across one's facial profile (Gaunt, A.)
- This information can be surprisingly accurate as to projecting one's emotions/inner state
- In a public area, someone with an anomalous thermal distribution might be a point of concern

On the right, a chart is showing how different emotions are represented through infrared (thermal) imaging.

Source: http://3.bp.blogspot.com/-jNMOxRydhas/UUn6HBvdY-I/AAAAAAAAAwc/F5Cv6kxrQ0Q/s1600/EMOTIONS_THERMAL_1.jpg

Thermal Images will be obtained from an open-source dataset. The dataset consists of hundreds of samples from all races and both males and females.

These thermal images will be preprocessed so that they fit within the system's memory and can therefore run. After the preprocessing, the image will be converted to a n x 1 vector, and added to an array with all of the other images.

The Support Vector Machine Image Classifier will take the data array as an input, and construct an SVM to test on. 80% of the data will be used to construct the SVM, and 20% will be used to test its effectiveness.

Approach of Image Classifier

- In the dataset, emotion ex1 (surprise) will be used as a surrogate for an anomalous thermal profile
- A kernel-based Support Vector Machine will be used to classify the thermal images (from over 500 samples from different people).
  - Using optimal parameters (C, gamma)

On the right, a raw image from the dataset is pictured. It depicts the thermal distribution across this man's face. After this data is preprocessed, it will be used as an input for the Machine Learning SVM Image Classifier. This is an example of an anomalous image, where there are two other facial expressions in the dataset. The image classifier will try to identify images with a similar expressions as this one.

Source: Central Ohio University

Preprocessing of Images

- Images need to be compressed and numerically represented for the purposes of our Image Classifier
- Images are converted to grayscale, compressed down to 100x100
- Vectorized implementations are added to the data array w/labelling

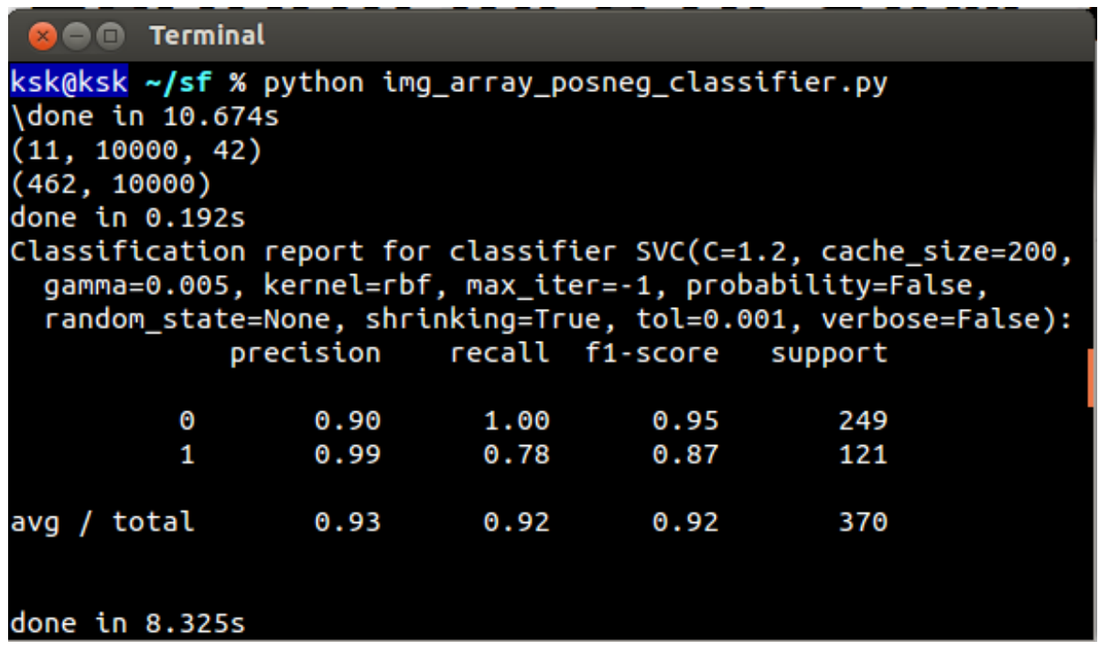
$$X = \quad = \begin{bmatrix} Pixel(0)^{(1)} \\ Pixel(1)^{(1)} \\ \vdots \\ \vdots \\ \vdots \\ Pixel(n)^{(1)} \end{bmatrix}$$

Pictured left, is what one data sample looks like. An image is taken from the original 320x240 size, and compressed down to 100x100. The image is converted to grayscale, converted to a vector, and added to an array of all the data samples (images).

Performance of SVM Image Classifier and Interpretation

- The SVM Image classifier was able to classify 90% of the negative samples, and 99% of the anomalous samples accurately (precision)
- Out of the identified samples, all of the samples thought to be negative were negative, and 78% of all of the identified anomalous samples were indeed anomalous.

To the right is the computer output of the SVM Image Classifier. It shows how well our classifier performed based on the given data, and given parameters. 0 represents a negative sample, and 1 represents a positive (anomalous) sample.



Conclusion and Implications of SVM Image Classifier for Anomaly Detection

- Our SVM Image classifier was able to capture 99% of the anomalous samples (with some false positives)
  - The classifier is effective, especially considering there were 10,000 features
- This data could be captured through thermal infrared cameras, in public locations.
- This data could indicate if someone is very sick and needs medical assistance.
- This data could indicate if someone is very anxious or has a high body temp, prompting further review

Agent-Based Public Area Simulation

- How people move can be very indicative of their intent or behavior in a public area
- Different patterns of movement can construe what someone is doing in an area
- Anomalous movement patterns can be a point of concern for further human review
  - Walking in circle/being in one area extensively
  - Being extremely averse towards other people
  - Erroneously meeting up and moving away from specific people

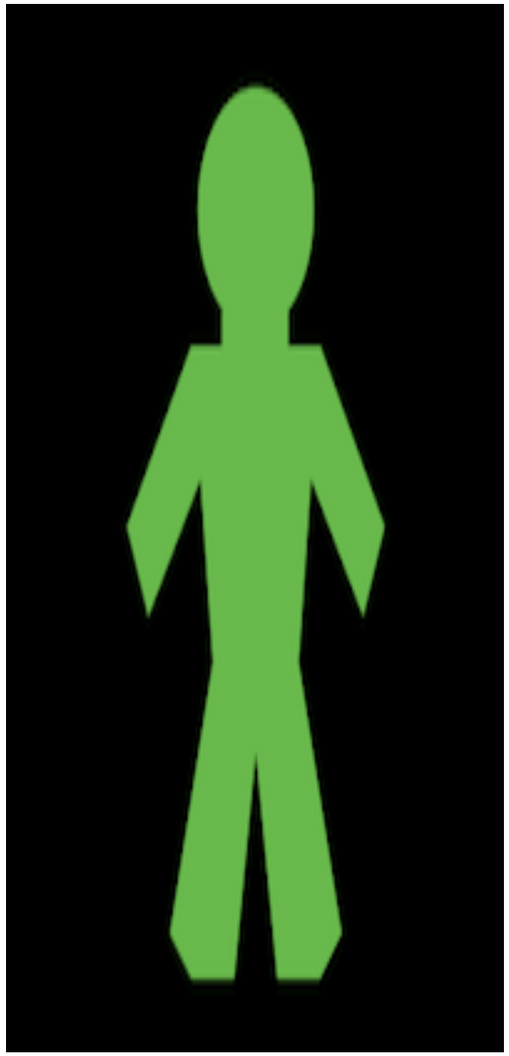
Persistics Algorithm (in theory), takes coordinate locations of different agents from every frame, and pieces them together to create a concrete route of every individual.

For this project, a simulation is created of an open-system public area. The simulation includes both normal and anomalous agents, which act in different ways. Embedded randomness is there to make it more realistic.

The simulation is run, and metrics are collected from the agent as data (speed distance etc). This data gives a good identity of what the agent is behaving like.

A kernel-based Support Vector Machine uses the collected metrics as features, and a SVM classifier is constructed on the data. The anomalous agents will be identified by the SVM through their collected metrics.

Agents Represented in the Simulation



Normal Agent

- 4 different normal agents
- Each move in a general cardinal direction
- Each have a 60° window of R/L movement
- Embedded randomness to emulate real - world situation

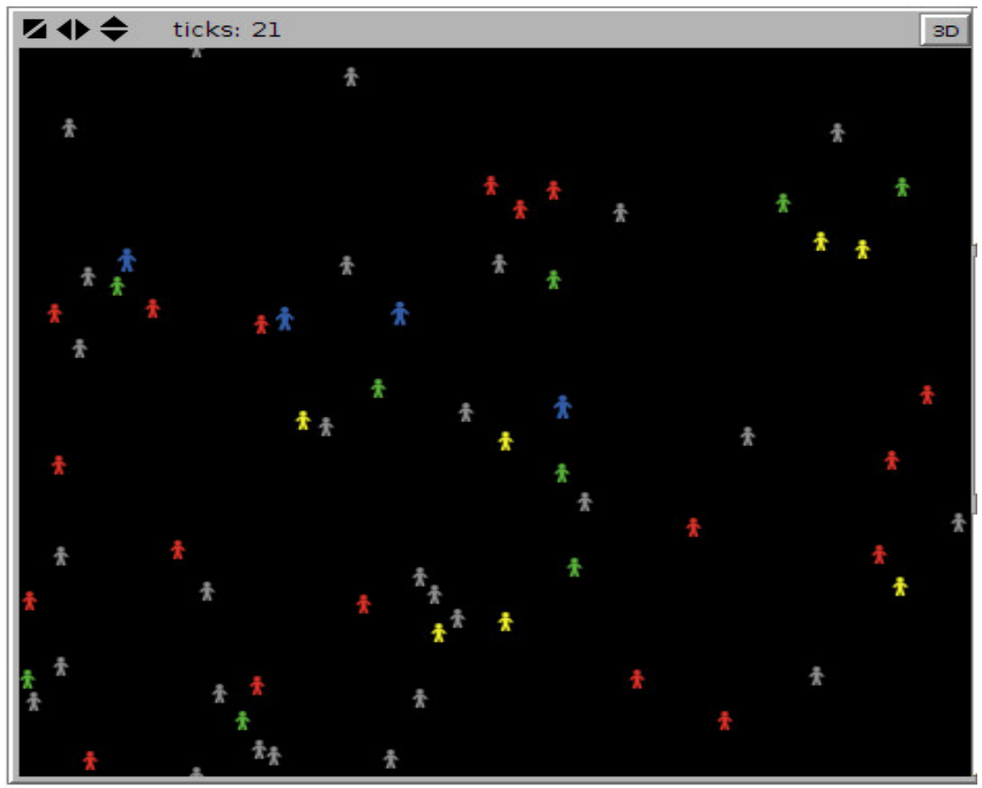
Anomalous Agent

- Direction much more defined
- Semi-circular movement pattern
- Only a 30° window of R/L movement
- Designed to avoid and move away from agents excessively (paranoid)

Simulation

- Simulation was written using the Netlogo language
- Has both normal and anomalous agents
- Data (features) collected from each agent for use in Machine Learning system

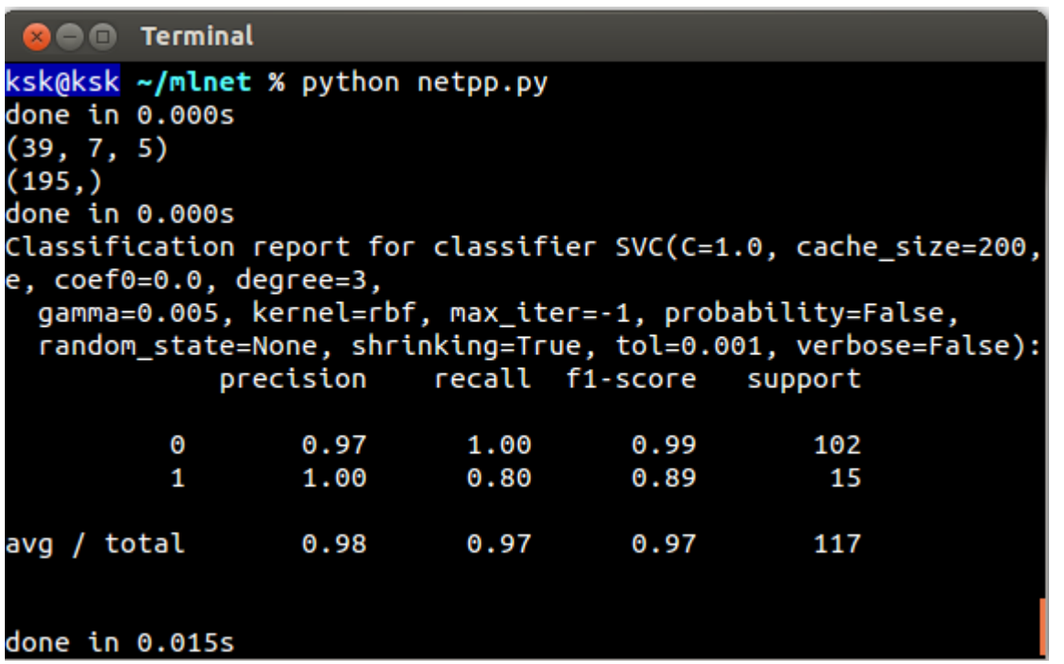
speed	Distance the agent has travelled over time
tickcount	Amount of time agent has been in the system
rdirf & ldirf	Total degrees agent moved in the right and left directions since start of the simulation
distance	Total distance agent moved in the simulation
total	Total number of agents in the simulation
nearby agents	Number of other agents within a radius of 4 units of an agent. Is a compounding value



Performance of Simulation SVM and Interpretation

- The SVM Classifier was able to identify 97% of the positive agents, and 100% of the anomalous agents accurately
- Out of the identified normal agents, all of the samples thought to be normal were correctly identified
- Out of the identified anomalous agents, 80% of those agents were indeed anomalous

Computer output of our SVM Classifier. It shows how well the classifier performed on the test data, based on the quality of training data, and the parameters on which it was run. Like the Image Classifier, 0 represents a normal agent, and 1 represents an anomalous agent.



Different sources of data used as surrogates for real-world data/situations

- Thermal images will be obtained from public dataset
- Machine Learning system designed to identify anomalous samples within the dataset

- An agent-based simulation will be designed in order to simulate how people move in a public system
- Anomalous agents embedded within the simulation
- ML system designed to identify anomalous agents based on collected metrics
- Persistics algorithm designed about collecting this info in a real-life use-case

Conclusion and Implications of SVM Classifier for Behavioral Anomaly Detection

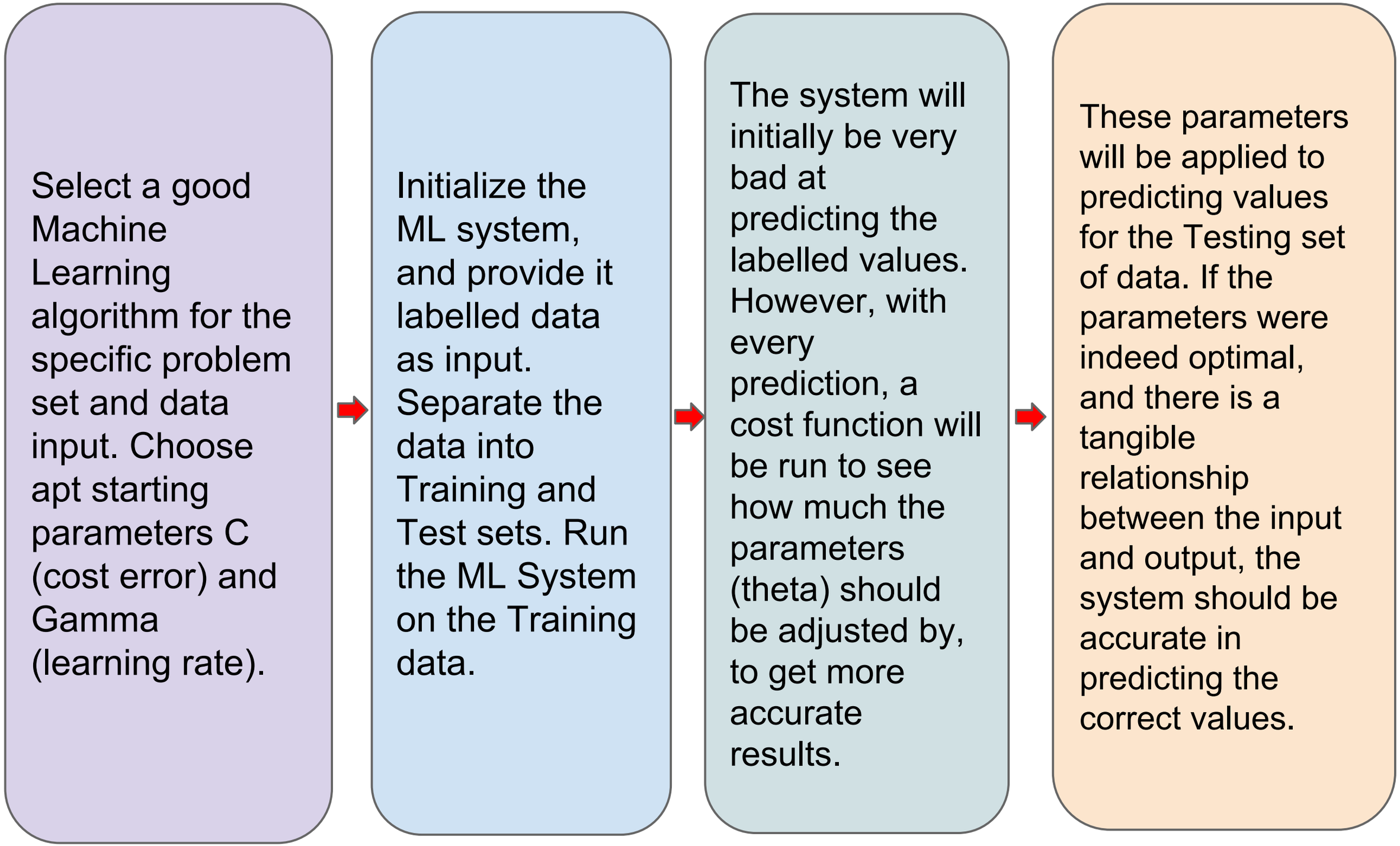
- Classifier for the data collected from the simulation was captured all of the anomalous agents (with 20% FPs)
  - Because the features were much more limited and concrete than the image classifier, the SVM Classifier had better performance
- Coordinate data and metrics could be calculated using triangulation and persistics in real-world implementation
- This Machine Learning system can be used to detect anomalous behavior, which would be useful in a variety of applications
  - Public Security
  - Asset Protection
  - Identifying medically at-risk individuals

Overall Conclusion

- The Machine Learning systems designed yielded promising results for implementation/further research
- Based on the results, thermal and behavioral data proved insightful in identifying anomalous data samples
- SVM Image Classifier was able to identify 99% of Anomalous Samples (with 22% False Positives)
- SVM Behavioral Classifier for simulation was able to identify 100% of the Anomalous Agents (with 20% FPs)
- Number of false positives can be mitigated by having a thermal and behavioral profile for each agent

What is Machine Learning?

- “A computer program said to learn from experience E with respect to some class of tasks T and performance measure P” - Tom Mitchell
- Supervised ML - When we know what kinds of outputs we want, with the notion that there is a relationship between the input and output.


Select a good Machine Learning algorithm for the specific problem set and data input. Choose apt starting parameters C (cost error) and Gamma (learning rate).
Initialize the ML system, and provide it labelled data as input. Separate the data into Training and Test sets. Run the ML System on the Training data.
The system will initially be very bad at predicting the labelled values. However, with every prediction, a cost function will be run to see how much the parameters (theta) should be adjusted by, to get more accurate results.
These parameters will be applied to predicting values for the Testing set of data. If the parameters were indeed optimal, and there is a tangible relationship between the input and output, the system should be accurate in predicting the correct values.

Kernel-Based Support Vector Machine

- Classifies whether a sample is positive or negative, based on its similarity to already known data points

$$f_1 = similarity(x, l^{(1)}) = exp(-\frac{||x - l^{(1)}||^2}{2\sigma^2})$$

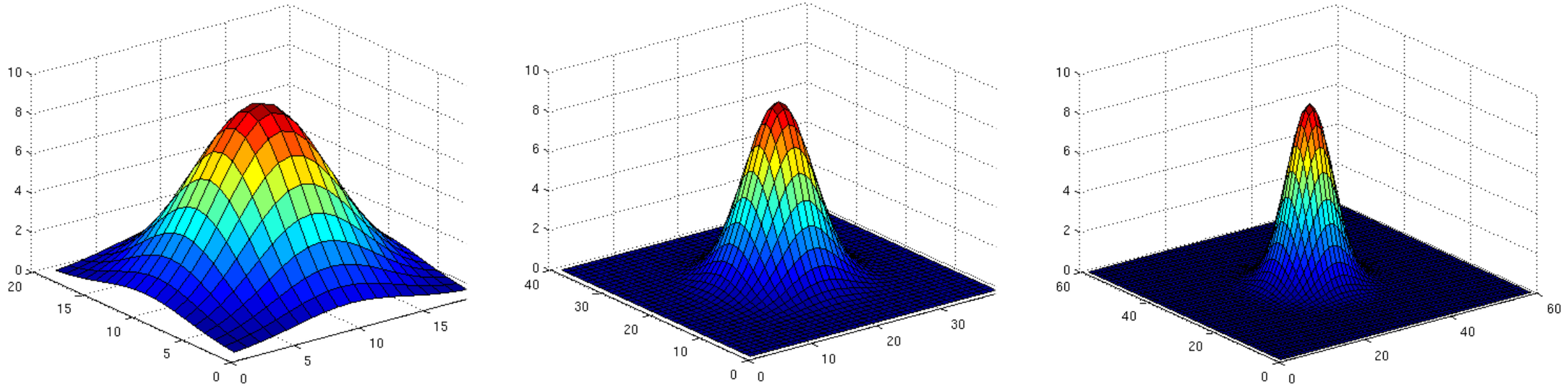
F1 is a similarity function between a hypothesis point (x), and a landmark point l. Notice that the parenthesis is being raised by the constant e, meaning the F1 value can only be between 0 and 1 (positive or negative). However, this means that the $2\sigma^2$ has an influence as to how sensitive the similarity function is.

SVM Hypothesis Function

- This is the function that determines the hypothesis in the SVM Classification algorithm.

$$h_\Theta(x) = \Theta_1 f_1 + \Theta_2 f_2 + \Theta_3 f_3 + \dots$$

The hypothesis is determined by a combination of the similarity function, and a “weight” value called theta. The similarity function from earlier is multiplied by theta, which is derived from a cost function (updates with every training example to get more accurate hypotheses). If the data is multidimensional, a similarity and cost is calculated and added for every dimension.



These figures show the effect of choosing different $2\sigma^2$ values in the similarity function. The first picture has a very small $2\sigma^2$ value, so there is larger range of accepted points that would be considered similar to the landmark point. As the $2\sigma^2$ values increase, the range of accepted points becomes more narrower.

Source: http://stackoverflow.com/questions/12606048/2d-3d-plot-of-image-processing-filters