Problem set guidelines

1. These problems require thought but do not require long answers. Please be as concise as possible.
2. Should you have any questions regarding this homework, please post them on Piazza. Chances are your classmates have them too. By posting and answering questions on Piazza, you help your peers to get better understanding of the class material.
3. You can discuss the problems with your classmates but do not share your code or the answers and do not use someone else's solutions.
4. Please submit the coding assignments in the form of Jupyter notebook along with the results of the execution and all the comments you like to make inline.
5. You can submit writing assignments in any form convenient for you. It could be LaTeX, MS Word, PDF or image. Please make sure it is of good enough quality.

Technical details

1. Coding assignment use Python 3.6. You can use either official distributive or Anaconda, which contains the majority of the required packages pre-installed for you.
2. Install packages from requirements.txt. You may need administrator right:
   pip3 install -r requirements.txt
3. Please use Jupyter for working with .ipynb files. In command line, type jupyter notebook.

1 Anomaly detection for datacenter

[30 points]

In this problem, you are going to build a model, which finds anomalies in the behavior of virtual machines in the datacenter.

We have collected for you a number of logs on VMs (actually, the dataset is a real data from a real datacenter). The logs are CPU load and memory load (RAM) for every VM. Your goal is to build a system detecting abnormal behavior of the VMs so that the system administrator can notice them and pay attention.

The data and Jupyter notebook file with some pre-defined code for you can be found in the folder **datacenter**.

Take a look at **system-load.csv**. This is a raw data file, partially preprocessed for you. You can open in in any text editor or table editor such as Excel. CSV is one of the most popular data format files, you will often use in in your study and at work. It is worth to investigate how it looks in details.

Please open **Problem Set 2 - Datacenter Anomaly Detection.ipynb** with Jupyter notebook, load the data and investigate the features. This part of code has already been written for you.

1a. Training Gaussian mixture model

[10 points]

Use scikit-learn implementation of the Gaussian mixture model and train it to fit the data you have.

1b. Setting up model parameters

[10 points]

Set up the number of Gaussians and abnormality threshold. Note there are no labels for the points in this dataset. We do not know which (if any) servers behave abnormally. Think how would you decide on the threshold in the real life. Visualizing the results could help.

Briefly explain your choice.

1c. Plotting the results

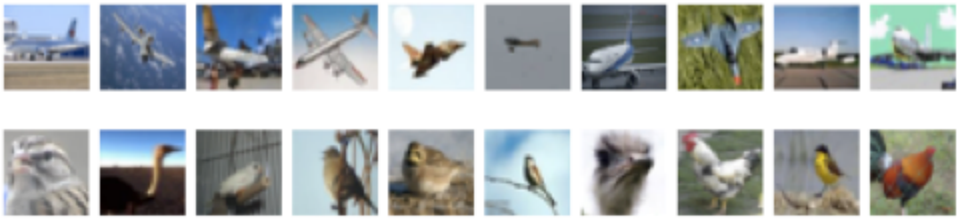
[10 points]

Visualize all the points from the dataset and density estimation of your model on top of them. Draw all abnormal points (falling below the threshold) in red.

2 Unsupervised feature learning

[40 points]

In this assignment, you will implement an image classifier that distinguishes birds and airplanes. We will see how choosing right features affects the performance of machine learning models. We will be working with the CIFAR-10 dataset, one of the standard benchmarks for image classification.



The data and Jupyter notebook file with some pre-defined code for you can be found in the folder **feature learning**.

Please open **Problem Set 2 - Unsupervised Feature Learning.ipynb** with Jupyter notebook, load the data. Please investigate this part of the code carefully. You will reuse parts of it in your own implementation of this assignment.

2a Training logistic regression

[2 points]

Train logistic regression on the raw pixel data and report the train and test set results.

2b Training SVM

[2 points]

Train SVM on the raw pixel data and report the train and test set results.

2c Training XGBoost

[1 point]

Train XGBoost on the raw pixel data and report the train and test set results.

2d Learning better features

[15 points]

Instead of hand-designing better features, let us see if we can learn them directly from data. Each image is a 32x32 grid of pixels. We will divide the image into sixteen 8x8 "patches". Next, we will use K-means to cluster all the patches into centroids. These centroids will then allow us to use a better feature representation of the image.

Run k-means from scikit-learn to group all patches into clusters. Initially, pick the number of clusters according to your best guess. After that, visualize the centroids.

2e Representing examples in a new way

[10 points]

Now, you have the centroids defining similar groups in your patches. Represent every image in your training and test set in distances between the patch and each centroid. For example, if you used 10 clusters and each image has 16 patches, new representation of the image will be a vector of 160 elements.

2f Training classifiers

[5 points]

Train all three classifiers from the above (logistic regression, SVM and XGBoost) on the new image representation. Report the train and test set results.

2g Getting the best out of it

[5 points]

In industry, we typically try to get as much as possible out of the data we have. Try different number of clusters and different configuration of the models and report the best accuracy you got on the test set.

3 Case studies

[30 points]

This deals with few real world situations we encountered in our careers. Now, you should be able to deal with them as well.

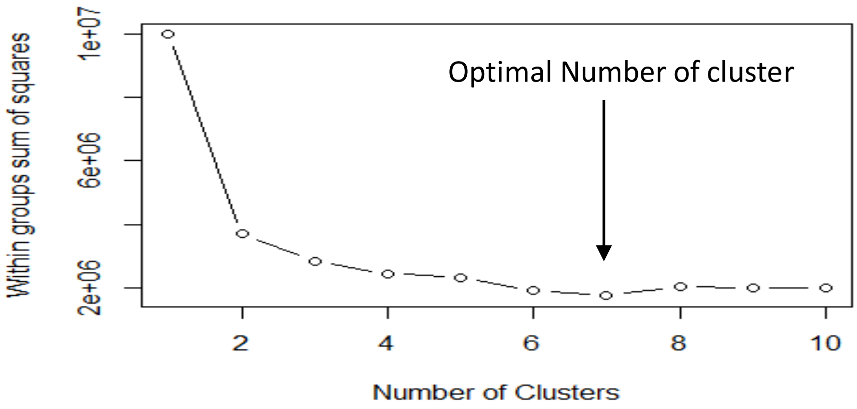
3a Clustering audit

[15 points]

A customer hired you to make an audit on the user segmentation done by their previous contractor. Among the rest of the results, you see the following in the subcontractor's report to your customer:

Clustering Technique: There are various methods that can be used for clustering. Based on research on best methods for mixed variable segmentation (categorical & continuous variables), k-means clustering has been used. K-means provides a method to select the ideal number of clusters that split the selected data set.

Optimal number of cluster (K) is determined by means of Elbow Method, which show within sum of square (WSS) value corresponding to number of clusters. In this case optimal number of segments is K = 7

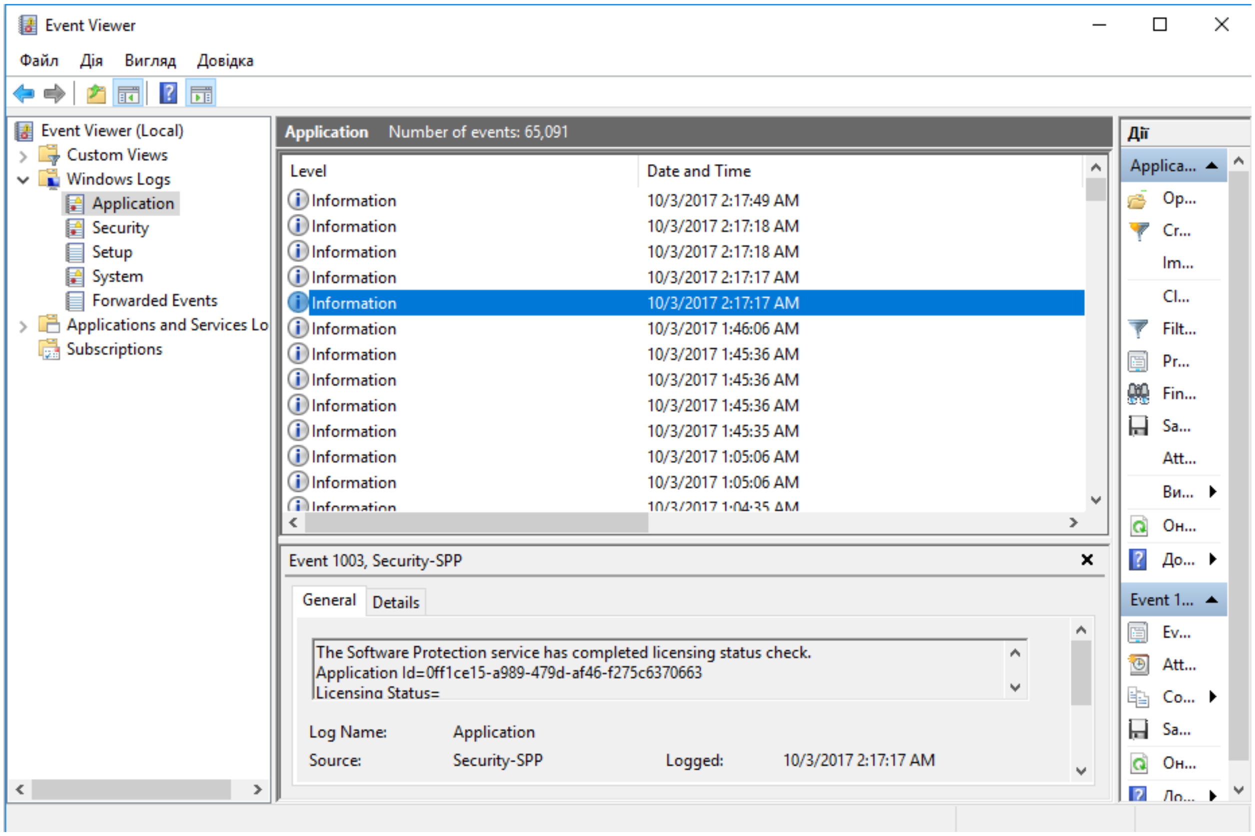


Evaluate this part of the report.

3b Anomaly detection for system events

[15 points]

You are designing anomaly detection system for datacenter but now you want to take into account not only hardware level (CPU, memory, disk I/O) but also the software level. For example, you want to rise the alert if the operating system behaves strangely even if hardware usage is (yet) normal. In order to do that, you collect syslog events. If you use Windows, you can see your syslog events with Event View tool.



Application Number of events: 65,091

Level	Date and Time
Information	10/3/2017 2:17:49 AM
Information	10/3/2017 2:17:18 AM
Information	10/3/2017 2:17:18 AM
Information	10/3/2017 2:17:17 AM
Information	10/3/2017 2:17:17 AM
Information	10/3/2017 1:46:06 AM
Information	10/3/2017 1:45:36 AM
Information	10/3/2017 1:45:36 AM
Information	10/3/2017 1:45:36 AM
Information	10/3/2017 1:45:35 AM
Information	10/3/2017 1:05:06 AM
Information	10/3/2017 1:05:06 AM
Information	10/3/2017 1:04:35 AM

Event 1003, Security-SPP

```
The Software Protection service has completed licensing status check.
Application Id=0ff1ce15-a989-479d-af46-f275c6370663
Licensing Status=
```

Log Name: Application

Source: Security-SPP Logged: 10/3/2017 2:17:17 AM

Image 1. Event viewer and syslog events

How would you represent the data in a form so that you can build Gaussian mixture model anomaly detection system? What features would you use?