Final Project Instruction

Posted date: 11 Mar 2022
Proposal due date: Fri 25 Mar 2022 (have to **meet me 1-1 for proposal approval before turning in**)
Final project presentation + code date: Week of 9 May 2022 or Week of 16 May 2022

Objectives

Propose your final project proposal. The proposal should have about 2 pages.

Timelines

- Discuss the project directly with the instructor (**important!**)
- Turn in project proposal [1-2 pages Word]
- Weekly meeting update [jupyter lab + 1-3 update slide presentation] ~ 2 min present + 2 min comment
    - Project can be changed along the way, but not recommended
    - Reading, learning, or installing can be a weekly update but only for 1-2 weeks
- Final project presentation [jupyter lab + slides + present to the class] ~ 10 min present + 5 min questions & comments

Grade

- Proposal [10%]
- Weekly meeting presentation [45%]
- Final project [code 30% + presentation 15%]

Choosing final project topics

The final project topic is very open. Students cannot pick the same project. There are 3 main paths:

Three main project directions

1. The project that **relates to your Master's thesis** or something you are very interested in. Please discuss directly with me (recommended)
2. The project**(s)** competes in the **online competition** or **solves real-world problems** (recommended):
    a. Example competition from https://app.datacamp.com/learn/competitions
    b. Work on simple unsolved questions/issues posted on stacks OverFlow or GitHub. Example from https://github.com/pandas-dev/pandas/issues, https://stackoverflow.com/questions/tagged/numpy
3. The project **contains knowledge outside of the classroom** as long as it can still run in Jupyter lab or Google Colab. Here are the examples:
    a. Using Github
    b. Big data tools: PySpark
    c. Machine learning tools: Scikit-learn, Pytorch
    Use `Pytorch` over `TensorFlow` since the `Pytorch` is newer and more popular in the research field and easier to use
    d. Data analysis with interactive plots

e. Web scrapper (extract data from the internet)

Examples of the final project but aim for a bit easier problem
https://github.com/stephanieirvine/Udacity-Data-Scientist-Nanodegree, https://github.com/pgoel05

Ideas for finding datasets

In case you want to focus on some analysis, but you don't have a dataset on your own. Data from these links might interest you

- Kaggle datasets. Kaggle is famous for various data science competitions, but it has a large list of datasets. You will find that many of these already have published analyses, so be careful to check what's out there, or chose something that has not been done before.
- Google dataset search. Try the link, you'll get the idea.
- World Bank data. The site has a lot of data on global development, and related issues. The gender_data.csv dataset we have been using is a processed version of https://data.worldbank.org/data-catalog/gender-statistics.
- UK government open data.
- Project Gutenberg - full text of many books.
- UK data service "The UK Data Service collection includes major UK government-sponsored surveys, cross-national surveys, longitudinal studies, UK census data, international aggregate, business data, and qualitative data."

What is expected in the proposal?

1. Project topic
2. Short Background information about the project + Inspiration (optional)
3. Expected scope of knowledge to solve the problem
    a. i.e. Pandas; Exploratory Data Analysis; Machine learning - classification
4. Timeline things to do + deliverables (หลักฐานการทำงาน)
   topic: use the online weather forecast to predict the corps yield (ผลผลิต)
    a. learn and read about web scrapper + deliverables: notes
    b. extract test data, daily weather + deliverables: Jupyter Lab to show the work, plot
    c. extract data in the project + deliverable: Jupyter Lab to show the work, plot data
    d. clean data, process data + deliverable: table compare pre-process data, processed data
    e. analysis, and business insight + deliverable: plot or metrics to assist the decision making

What is expected in the weekly update?

Warning:

Don't believe in a one-night miracle! Even if you only turn in just 2-3 slides PowerPoint each week, it takes a lot of time to do proof (deliverables) and write **a reasonable plan** for next week.

The grade of the final project weighs the most on the weekly presentation because it shows your problem-solving, consistency, and adaptability

Presentation page 1:

- Summarize last week's work done and plan to do on next week
- Thing happens 💩. Don't be afraid to put unsuccessful works in the presentation. We can get through the bugs together

Done (Week 7 Mar 2022)	Plan (Week 14 Mar 2022)
Summarize outlier detection, isolation forest method Proof: Note # with a link # green color means 100% complete	Compare isolation forest with other models Proof: plot 2 model and accuracy metric
Test example code outlier detection Proof: run on self-generated data # yellow color means 30-99% complete Proof: run on online data # python can't read online data # red color means have a bug, problem, not start, 0-29% complete	Test example code outlier detection Proof: run on online data
Summarize k-mean clustering ~~Proof: Plot of using k-mean in outlier detection~~ # it is for clustering, not for outlier detection	

Presentation Page 2-3:

- Choose some important plots or tables from proof of work. You must include (verbally or in the presentation) related information about the plot that helps the viewer understand too
    - Where does the code come from?
        - data from Hat Yai hospital
    - Relate parameters
        - data n = 1000, weight vs height or baby in Hay Yai from 2000-2001
    - What do the plots imply
        - there is some outlier from input errors, some babies weight 3000 kg
    - Important metrics
        - outlier = 0.1 %
- Print screen/ online link to your note (if have one), an example of the note
- Copy error message (if have one)

Here are some examples of the weekly update presentations

Follow the format

Example 1

Mai: install and test U-SPORF

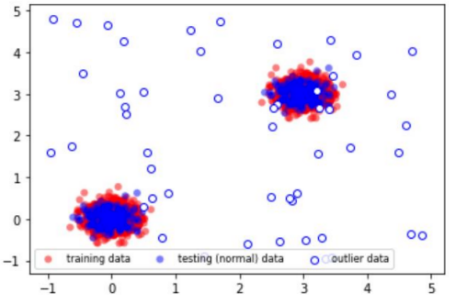
Last week

- Summarize bootstrap method, random forest isolationforest, random forest visualization (from *ML web,towards data science, scikit*)
  - DoD: Note
- Test accuracy of 'IsolationForest' (outlier detection) with 2 sample set
  - DoD: in random 2-D data + benchmarking (% correct)
  - DoD: mixed outlier data in Iris data? + benchmarking (% correct)
- Visualize decision Tree
  - DoD: from RandomForestClassifier
  - DoD: from UnsupervisedRandomForest

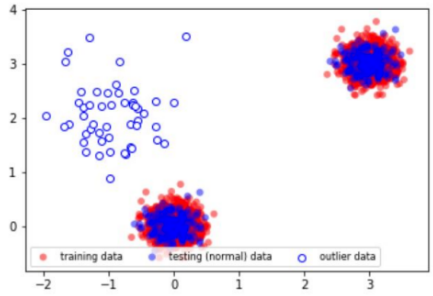
This week

- Summarize Gaussian Matrix Model (from *wiki*, *scikit*, *brilliant*)
  - DoD: Note
- Test accuracy of 'IsolationForest' (outlier detection) with 2 sample set
  - DoD: mixed outlier data in Iris data
- Find other method for outlier detection beside from
  - DoD: visualized data with performance + benchmarking (% correct)
- Proposal
  - DoD: the draft for Outlier detection before meeting with TA

IsolationForest
Training n =2000
testing normal n=400
Testing outlier n=100
Prediction Accuracy #correct prediction / #tot



Normal Prediction Accuracy: 0.93
Outliner Prediction Accuracy: 0.96



Normal Prediction Accuracy: 0.9225
Outliner Prediction Accuracy: 1.0

Example 2

Mai:

This week

- Summarize Extended Isolation Forest (EIF) [1,2]
  - DoD: Note
- Demonstrate the anomaly (outlier) score contour of EIF v.s. standard Isolation Forest (IF); Compare the outlier detection performance between EIF and IF on real data from ODDS[3]
  - DoD: Code
- ~~GMM from clustering to outlier detection~~

Next

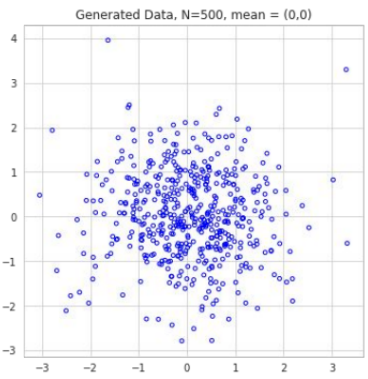
~~This week~~

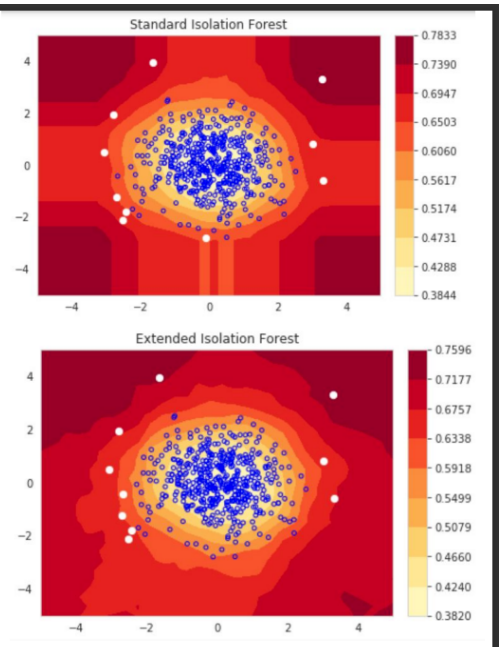
- Reread SPORF, USPORF paper and take note on how can it detect outliers
  - DoD: Note
- Compare the outlier detection performance between USPORF, EIF, and IF on real data from ODDS[3]
  - DoD: Code

1: Paper Hariri, S., Kind, M. C., & Brunner, R. J. (2018). Extended Isolation Forest. *_arXiv preprint arXiv_*:1811.02141.
2: Toward Data Science
3: Outlier Detection DataSets (ODDS) → Vertebral dataset → n_tot=240, d=6, n_outlier=30 (12.5%)

$$s(x,n) = 2^{-\frac{\mathbb{E}(h(X))}{c(n)}} \text{ when,}$$

- $\mathbb{E}(h(X))$ is the length of edge from the root node to the terminal node
- $c(n)$ is the average length
- x is the data point
- n is sample population
- $0 < s(x,n) < 1$: the closer $s(x,n)$ to 1 the more possibility the data is to be an outlier.





Data
Outlier Detection DataSets (ODDS) → Vertebral dataset

n_tot=240, d=6, n_outlier=30 (12.5%), label = y = {0,1}

	0	1	2	3	4	5
0	63.03	22.55	39.61	40.48	98.67	-0.25
1	39.06	10.06	25.02	29.00	114.41	4.56
2	68.83	22.22	50.09	46.61	105.99	-3.53
3	69.30	24.65	44.31	44.64	101.87	11.21
4	49.71	9.65	28.32	40.06	108.17	7.92
5	40.25	13.92	25.12	26.33	130.33	2.23
6	53.43	15.86	37.17	37.57	120.57	5.99
7	45.37	10.76	29.04	34.61	117.27	-10.68
8	43.79	13.53	42.69	30.26	125.00	13.29
9	36.69	5.01	41.95	31.68	84.24	0.66
10	49.71	13.04	31.33	36.67	108.65	-7.83

```
if_eif = iso.iForest(X.values,
                     ntrees = 100,
                     sample_size = 100,
                     ExtensionLevel = 0)

# calculate anomaly scores
anomaly_scores = if_eif.compute_paths(X_in = X.values)
# sort the scores
anomaly_scores_sorted = np.argsort(anomaly_scores)
# retrieve indices of anomalous observations
indices_with_preds = anomaly_scores_sorted[-int(np.ceil(anomalies_ratio * X.shape[0])):]
# create predictions
y_pred = np.zeros_like(y)
y_pred[indices_with_preds] = 1
```

What is expected in the final project

Presentation

Your goal is to communicate your work to the instructor and your classmates. Please use as simple language as possible.

- **Introduction:** What is the project topic? Why is it important or interesting? Where does the problem come from?
- **Workflow & results**: focus on actual results
    - Unsuccessful attempts are already shown in weekly meeting
- **Conclusion:** What do the results imply? What can be improved on this project? Future project

Code or notebook

Make a easy to read and organize code. The flow should be similar to the presentation. The goal is to upload code to GitHub as a personal project in your portfolio.

Examples

Here is the link to the DataCamp's certificate case study instructions and grading. You can look at the grading table to get a rough idea of what should be in the final code/notebook and presentation.