Assignment 09
kNN, SVM, Kernels
EECE-6822 Machine Learning

Cris Ababei
Electrical and Computer Engr., Marquette University

1. Objective

The objectives of this activity include: (1) run several code examples that illustrate kNN for classification; (2) run several code examples that illustrate SVM for classification; and (3) conduct an investigation of different distance measures in the context of kNN.

2. Prerequisite Readings

Murphy

- Ch. 16.1: K nearest neighbor (KNN) classification
- Ch. 17.3: Support Vector Machines (SVM)

Geron

- Ch.5: Support Vector Machines (SVM)

Raschka

- Ch.3: K-nearest neighbors – a lazy learning algorithm. Solving nonlinear problems using a kernel SVM

3. Code Examples

Example 1: SVM in SciKit-Learn

This is the example code from Ch.5 from Aurelian Geron's book.
[*B3-Geron] Aurelien Geron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, O'Reilly, 2022.
Open in your google colab and go through the code. You should read first the chapter itself from the book, before going through the code.

https://github.com/ageron/handson-ml3/blob/main/05_support_vector_machines.ipynb

The chapter explains the core concepts of SVMs, how to use them, and how they work.

Example 1: kNN, SVM in SciKit-Learn

This code example is from Ch.3 from:
[*B3-Raschka] Sebastian Raschka, Yuxi Liu, and Vahid Mirjalili, Learning with PyTorch and Scikit-Learn: Develop machine learning and deep learning models with Python, Packt Publishing, 2022.
The source code (both Python code an Jupyter Notebook) is located at the GitHub repository. You should read first the chapter itself from the book, before going through the code.

https://github.com/rasbt/machine-learning-book/tree/main/ch03

To execute the code, I found the easiest usually to run directly the Python code (same code as in the notebook) in Anaconda Spyder. However, because the source code was developed with earlier versions of various libraries, you may need to fix little things here and there to make it work.

The last short section of this chapter shows how to use "**from sklearn.neighbors import KNeighborsClassifier**" to define a knn classifier for the Iris flow dataset. This chapters also shows how to use a SVM model "**svm = SVC(kernel='rbf', random_state=1, gamma=0.10, C=10.0)**".

Example 2: kNN, SVM in SciKit-Learn

This is an example suggested by K. Murphy on his github repository for the textbook:
https://github.com/probml/pyprobml/tree/master/notebooks/book1/16

Scroll to the bottom of the above page to find a notebook, **knn_demo.ipynb**, that also demonstrates "**KNeighborsClassifier**":

https://colab.research.google.com/github/probml/pyprobml/blob/master/notebooks/book1/16/knn_demo.ipynb

Another example on SVM can be found at (adapted actually from Geron' book):

https://github.com/probml/pyprobml/blob/master/notebooks/book1/17/svm_classifier_2d.ipynb

Example 3: mlm tutorials on kNNs

Here, we look at several tutorials from **machinelearningmastery (mlm)**. Read the following tutorials and run the code where applicable:

--Develop k-Nearest Neighbors in Python From Scratch

https://machinelearningmastery.com/tutorial-to-implement-k-nearest-neighbors-in-python-from-scratch/

--4 Distance Measures for Machine Learning

https://machinelearningmastery.com/distance-measures-for-machine-learning/

--How to Train to the Test Set in Machine Learning

https://machinelearningmastery.com/train-to-the-test-set-in-machine-learning/

--Imbalanced Multiclass Classification with the Glass Identification Dataset

https://machinelearningmastery.com/imbalanced-multiclass-classification-with-the-glass-identification-dataset/

--Support Vector Machines for Machine Learning

https://machinelearningmastery.com/support-vector-machines-for-machine-learning/

--Cost-Sensitive SVM for Imbalanced Classification

https://machinelearningmastery.com/cost-sensitive-svm-for-imbalanced-classification/

--Imbalanced Multiclass Classification with the E.coli Dataset

https://machinelearningmastery.com/imbalanced-multiclass-classification-with-the-e-coli-dataset/

--Imbalanced Multiclass Classification with the Glass Identification Dataset

https://machinelearningmastery.com/imbalanced-multiclass-classification-with-the-glass-identification-dataset/

4. Assignment

First, you must read and do the following tutorials from mlm:

[T1] Develop k-Nearest Neighbors in Python From Scratch

https://machinelearningmastery.com/tutorial-to-implement-k-nearest-neighbors-in-python-from-scratch/ (focus especially on section "**Iris Flower Species Case Study**" – the first complete example in this section that reports "Mean Accuracy").

[T2] 4 Distance Measures for Machine Learning

https://machinelearningmastery.com/distance-measures-for-machine-learning/

Part 1: You must change the code from T1 and redo it for all the other three distance measures from T2 (currently T1 example uses Euclidean distance). Record "Mean Accuracy" for each of the four distances and create a plot to show them. Present a discussion around which one is the best.

Part 2: In this part, you must bring additional code from this example:

https://colab.research.google.com/github/probml/pyprobml/blob/master/notebooks/book1/16/knn_demo.ipynb

into the code from Part 1. You must use the **KNeighborsClassifier** instead of using the implementation of kNN from the example from Part 1. You objective here is to to generate the plot "**train err and test err with different k**" for each of the four distance measures, for the following values of **k = {1,2,3,5,10}**. So, you must generate four such plots and include in your report and discussion of results.

5. Deliverables

You must write (typed) a report and upload it as a PDF file on D2L. The report should be named "**hw9_report_LastName.pdf**". You should also create a .zip archive with all your code and implementations of all parts of the assignment. Upload also this archive .zip file with the name "**hw9_implementation_code_LastName.zip**" to D2L. Hence, your D2L should contain two items: the report and the .zip file. **Do not include the report inside the .zip and upload only the .zip. They should be two separate items!**

The report should include the following sections and subsections. Make sure section titles are in bold font and pages are numbered.

1) **Title + course info + your name**
2) **Summary.** Describe in one paragraph what the objective of the assignment is.
3) **Description of Experiments and Discussion.** Describe the experiments you did. All tables and figures should be numbered and should have captions. All plots in all figures should have axes labels and titles. Present a meaningful discussion with the interpretation of the results you obtained. Explain if you expected the results or not; discuss the intuition behind it.
4) **Conclusion.** Present your conclusions; highlight what are your main takeaways that you learned from this assignment. Describe what issues you encountered and how you solved them.
5) **References.** Include all references that you used, as a numbered list. Cite them in the report itself; do not just list them! If your report has References that are not numbered and cited in the report, points will be deducted!