Introduction to Machine Learning

ECE 580
Spring 2022

HW #6

Submission Instructions

Submit your work to the corresponding assignment in Gradescope. Although Gradescope accepts multiple file formats, they strongly recommend submitting your assignment as a single PDF file.

You are responsible for ensuring the uploaded file is: 1) the correct file, 2) complete (includes all pages), 3) legible, and 4) submitted on-time as determined by the Gradescope server system clock.

You are responsible for tagging the pages that correspond to each question. Pages may be tagged after submission, even if the submission deadline has passed. If you are submitting close to the submission deadline, submit your assignment first then immediately return to tag the pages.

When code is requested, submit a PDF print-out of your code. Submitting a URL for a cloud-based repository is insufficient.

Scoring Information

Individual questions will be scored holistically on the 9-point scale described in the syllabus. The homework assignment score will be the weighted average of the individual scores for each question. (The weight for each question is shown in parentheses to the left of the question number.)

Late Submissions

Late submissions will be accepted up to 5 days after the submission deadline, with the following point penalty applied if its late submission is not excused: [1]

1 day (0^+ to 24 hours) late: no late penalty
2 days (24^+ to 48 hours) late: no late penalty
3 days late: no late penalty
4 days late: 1.6 point deduction
5 days late: 3.2 point deduction
6 or more days late: score = 3.0 (not accepted for credit)

The late policy is designed to be minimally punitive for submissions up to 3 days late, yet encourage staying current with the coursework for our course by not allowing one assignment's late submission to overlap with the next assignment's submission.

A homework score will not drop below 3.0 as a result of applying the late penalty point deduction.

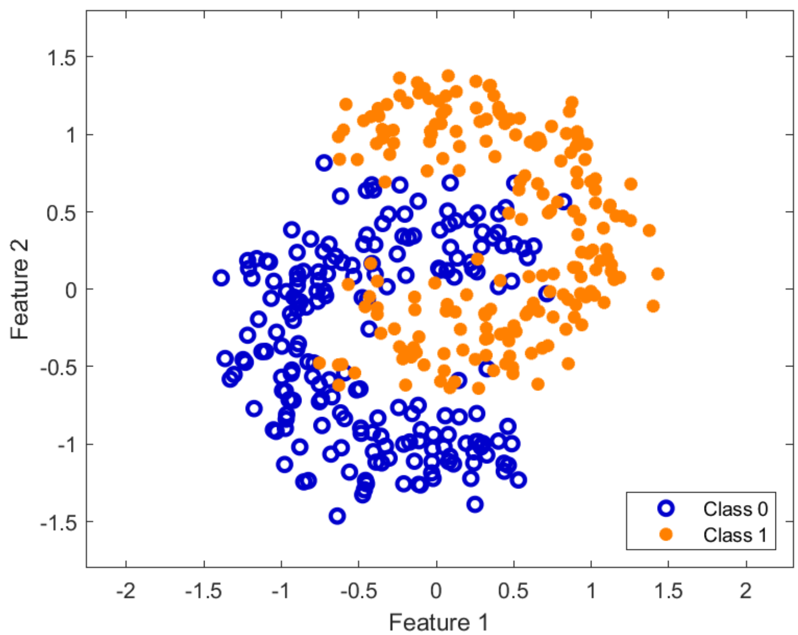
[1] One day = one 24-hour period or fraction thereof.

Exploring Support Vector and Relevance Vector Machines

Make sure you are able to apply a Support Vector Machine (SVM) classifier and that you have flexibility to specify the kernel, including a linear kernel (or no kernel), and your SVM implementation can provide the decision statistic (not only the final decision).

Make sure you are able to apply a Relevance Vector Machine (RVM) classifier and that you have flexibility to specify the kernel, including a linear kernel (or no kernel). For Matlab users, Mike Tipping's Sparse Bayesian package[2] is a good option. For Python users, James Ritche's RVM for scikit-learn[3] is a good option.

This assignment will be using the 'Horseshoe' data set used previously in Homework 4. The data is provided as a `csv` file, `dataSetHorseshoes.csv`, that is organized such that each row contains the true class (either 0 or 1) followed by the associated (2-dimensional) feature vector. When you visualize the data set, you should see this:



(15) 1. Apply a Support Vector Machine (SVM) classifier with a linear kernel[4] (no kernel) to this dataset.

   (a) Visualize the decision statistic surface with the support vectors superimposed on top. The true class of each support vector should be easily discernible (*e.g.*, encode class using color and/or symbol).

   (b) Comment on the locations and relative sparsity of the support vectors.

   (c) Implementing an SVM with a linear kernel results in a linear boundary. Comment on how the location and slope of the linear boundary produced by an SVM with a linear kernel compares to the nearly linear boundary produced by the KNN classifier with very large k ($k \to 399$). Which linear boundary do you find to be more "trustworthy"?

[2] `http://www.miketipping.com/downloads.htm`
[3] `https://github.com/JamesRitchie/scikit-rvm`
[4] The linear kernel is also referred to as a direct kernel.

(15) 2. Apply a Relevance Vector Machine (RVM) classifier with a linear kernel (no kernel) to this dataset.

(a) Visualize the decision statistic surface with the relevance vectors superimposed on top. The true class of each relevance vector should be easily discernible (*e.g.*, encode class using color and/or symbol).

(b) Comment on the locations and relative sparsity of the relevance vectors.

(c) Implementing an RVM with a linear kernel results in a linear boundary. Comment on how the location and slope of the linear boundary produced by an RVM with a linear kernel compares to the nearly linear boundary produced by the KNN classifier with very large k ($k \to 399$). Which linear boundary do you find to be more "trustworthy"?

(5) 3. Comment on the relative strengths and weaknesses of the linear kernel SVM and linear kernel RVM when applied to this 'Horseshoe' data set. (Be sure to compare and comment on both the relative sparsity and the locations of the support/relevance vectors.)

(5) 4. Submit a PDF print-out of your code for this section (Exploring Support Vector and Relevance Vector Machines). (Submitting a URL for a cloud-based repository is insufficient.)

Exploring Kernels in Support Vector and Relevance Vector Machines

Kernels allow SVMs and RVMs to provide *nonlinear* decision boundaries even though they are linear classifiers, *i.e.*, their decision statistics are the *linear* combination of (potentially nonlinear) basis functions.

(15) 5. Apply a Support Vector Machine (SVM) classifier with a radial basis function (RBF) kernel[5] to this dataset.

(a) Visualize the decision statistic surface with the support vectors superimposed on top. The true class of each support vector should be easily discernible (*e.g.*, encode class using color and/or symbol).

(b) Comment on the locations and relative sparsity of the support vectors. What are the support vectors characterizing in the data?

(c) Comment on the shape (contours) of the decision statistic surface. Does the shape of the decision statistic surface make intuitive sense?

(d) Implementing an SVM with an RBF kernel results in a nonlinear boundary. Comment on how the location and shape of the nonlinear boundary produced by an SVM with an RBF kernel compares to the boundary produced by the KNN classifier with small k ($k \to 1$). Which nonlinear boundary do you find to be more "trustworthy"?

[5] You may wish to experiment with the RBF parameter.

(15) 6. Apply a Relevance Vector Machine (RVM) classifier with a radial basis function (RBF) kernel to this dataset.

   (a) Visualize the decision statistic surface with the relevance vectors superimposed on top. The true class of each relevance vector should be easily discernible (*e.g.*, encode class using color and/or symbol).

   (b) Comment on the locations and relative sparsity of the relevance vectors. What are the relevance vectors characterizing in the data?

   (c) Comment on the shape (contours) of the decision statistic surface. Does the shape of the decision statistic surface make intuitive sense?

   (d) Implementing an RVM with an RBF kernel results in a nonlinear boundary. Comment on how the location and shape of the nonlinear boundary produced by an RVM with an RBF kernel compares to the boundary produced by the KNN classifier with small k ($k \to 1$). Which nonlinear boundary do you find to be more "trustworthy"?

(5) 7. Comment on the relative strengths and weaknesses of the RBF kernel SVM and RBF kernel RVM when applied to this 'Horseshoe' data set. (Be sure to compare and comment on both the relative sparsity and the locations of the support/relevance vectors.)

(10) 8. Describe the factors you would consider when choosing between SVM and RVM classifiers, and why you consider these factors important considerations.

(10) 9. Describe the factors you would consider when choosing a kernel or kernel parameters (such as the RBF radius) for an SVM or RVM classifier, and how you expect your choice of kernel and/or kernel parameters to affect the resulting classifier.

(5) 10. Submit a PDF print-out of your code for this section (Exploring Kernels in Support Vector and Relevance Vector Machines). (Submitting a URL for a cloud-based repository is insufficient.)