ECE 5258 - Pattern Recognition (Fall 2016)
Mini-Project #2

Dr. Georgios C. Anagnostopoulos*

October 2016

1 Objectives

The objective of this Mini-Project is to expose the students to (i) Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA) and Multi-Nomial Regression (MNR) classification and (ii) formal hypothesis testing using the pair-wise Sign Test to compare classification performances. Additionally, the students will be exposed to preliminary notions of model regularization.

As usual, standard preparation guidelines (Section 4) and submission instructions (Section 5) are provided, which the students are expected to strictly adhere to. Finally, at the end of this document, a few, possibly helpful, references can be found.

2 Problem Setting

This assignment deals with classification of three genuses of *Iris* plants (*e.g.* see [6]), namely *Setosa* (class 1), *Versicolour* (class 2) and *Virginica* (class 3). The classification is to be made based on 4 features that correspond to 4 physical characteristics of their flowers: sepal length, sepal width, petal length and petal width, which are all measured in centimeters.

The relevant data set to be used is the *Iris Dataset* obtained from UCI Machine Learning Repository (UCIMLR) [3]. R.A. Fisher, the founding father of what we know as modern statistics, was the first one to popularize it in [5] and since then it has become, perhaps, the most famous data set in Machine Learning. The data set consists of 150 samples with each class being represented equally by 50 samples. The data are stored in the file `iris.data.shuffled.mat` as follows: the 150×4 `Pattern` matrix contains the real-valued feature values in the order just mentioned and the 150×1 vector `Label` contains the class labels $(1 - 3)$ for each sample. This data set is a shuffled version (the order of samples has been randomly permuted) of the original one found at UCIMLR.

The *Iris* population will be modeled as a 3-class Gaussian Mixture Model (GMM). 3 pertinent types of classifiers will be used to classify *Iris* samples, namely, LDA, QDA and MNR.

Note(s): Scalars are depicted in normal font, vectors in lower-case bold face and matrices in upper-case bold face. All vectors are considered column vectors. If $\mathbf{A}$ is a matrix, then $\mathbf{A}^T$ denotes its transpose. MATLAB keywords and/or code are depicted in `orange` font.

*georgio@fit.edu

3 Assignments

Task 1. [25 total points]
Before you attempt this part, please understand well the material presented in [1]. Implement the different variations of QDA and LDA in `MATLAB` according to the following suggestions:

- Author the function `da_train( )` to implement the training phase of all LDA and QDA variants. Use the following function signature `function [Means, Covariances, Priors] = da_train(Xtrain, Ltrain, classifier_type)`, where `Xtrain` $\in \mathbb{R}^{N_{train} \times D}$ contains N_{train} training samples arranged in rows (each sample is D-dimensional) and `Ltrain` $\in \mathbb{R}^{N_{train} \times 1}$ contains the corresponding class labels in a single column. If C is the number of classes represented in `Xtrain`, you can safely assume that the labels in `Ltrain` and `Ltest` will belong to $\{1, 2, \ldots, C\}$. You should also assume that `Xtrain` always contains at least one sample from each one of the C classes. Finally, `classifier_type` equal to $1, 2, 3$ specifies the use of a LDA model (1 for the general, 2 for the Naive Bayes and 3 for the isotropic variants) and `classifier_type` equal to $4, 5, 6$ specifies the use of a QDA model (4 for the general, 5 for the Naive Bayes and 3 for the isotropic variants). Regarding the output variables `Means` $\in \mathbb{R}^{C \times D}$ should contain the estimated means in rows (row k contains the estimate for class k), `Covariances` $\in \mathbb{R}^{D \times D \times C}$ should contain the covariance matrix estimates of the classes and `Priors` $\in \mathbb{R}^{C}$ should contain the estimated class priors.
- Author the function `da_classify( )` to implement the testing/performance phase of all LDA and QDA variants. Use the function signature `function [Lpred, Scores] = da_classify(Xtest, Means, Covariances, Priors)`, where `Xtest` $\in \mathbb{R}^{N_{test} \times D}$ contains N_{test} test samples arranged in rows. The return arguments are: `Lpred` $\in \mathbb{R}^{N_{test} \times 1}$ are the labels of the test samples as predicted by the LDA/acQDA classifier, while `Scores` $\in \mathbb{R}^{N_{test} \times C}$ contains the classifier's predicted posterior class probabilities $\hat{P}(\omega_i|\mathbf{x})$ for $i = 1, \ldots, C$ arranged in a row for each test sample.

After implementing these functions, address the following parts:

(a) **[5 points]** Plot all 2-dimensional scatter plots of the *Iris* data set by taking different pair combinations of the 4 original features and make comments about the distributions of the 3 classes. Eventually, retain the petal length (feature #3) & petal width (feature #4) features to form the *IrisReduced* data set, so that the sample distributions and the effect of the classifiers can be visualized in the next parts.

(b) **[5 points]** Form a training set S_{train} out of the first 50 samples in *IrisReduced*. Train all 3 cases of LDA (general case, independent features and isotropic for each classifier) and pick out a single champion model through Leave-One-Out Cross-Validation (LOOCV) using S_{train}. Report the LOOCV errors for each of the 3 models in a small table and point out the champion model. Regarding LOOCV, please refer to your textbook [4] or to online resources. What is LOOCV used for and under what circumstances is it useful?

(c) **[5 points]** Repeat the process delineated in Part (b) for the 3 QDA variants. Finally, compare the two resulting tables. Were these results expected (why or why not), in light of the data distribution?

(d) **[10 points]** Show the decision regions of your LDA and QDA champion classifiers obtained in Part (b) and Part (c) and comment on them. Are the decision boundaries as expected?

Task 2. [25 total points]
Before attempting this part, it is probably a good idea for you to revisit the material presented in [1]. A variant of QDA, which we will call Regularized Quadratic Discriminant Analysis (RQDA), subtracts a particular penalty term $r(\boldsymbol{\theta})$ from the GMM's log-likelihood function $\ell(\boldsymbol{\theta})$ giving rise to the *penalized log-likelihood* $\ell_r(\boldsymbol{\theta})$. In specific, if C is the number of classes involved in the mixture, the penalty term r in the case of RQDA is given as

$$r(\boldsymbol{\theta}) = -\frac{1}{2} \sum_{i=1}^{C} \rho_k N_K \text{trace}\left\{\mathbf{C}_k^{-1} \mathbf{R}_k\right\} \tag{1}$$

where the ρ_k's are non-negative penalty parameters and the matrices $\mathbf{R}_k$'s are symmetric and positive-definite. Both of these kinds of parameters are user-specified depending on what the user is trying to achieve or address. For example, a sometimes useful choice is $\mathbf{R}_k = \hat{\mathbf{S}}^{pooled}$ for all k, where $\hat{\mathbf{S}}^{pooled}$ is the maximum likelihood estimate of the common covariance matrix used in LDA.

(a) **[5 points]** State good reasons why $r(\boldsymbol{\theta})$ is referred to as a "penalty term". What is being penalized and in what way? What are the roles of the ρ_k's? Provide detailed answers with appropriate arguments.

(b) **[5 points]** Argue concretely and with enough detail that the the values of the class priors π_k^* and class means $\boldsymbol{\mu}_k^*$ for $k = 1, 2, \ldots, C$ that maximize ℓ_r coincide with the maximum likelihood estimates utilized in QDA (respectively, $\hat{\pi}_k$ and $\hat{\boldsymbol{\mu}}_k$).

(c) **[5 points]** Show that the values $\mathbf{C}_k^*$ of the class covariance matrices that maximize ℓ_r are given as

$$\mathbf{C}_k^* = \hat{\mathbf{C}}_k + \rho_k \mathbf{R}_k \qquad\qquad k = 1, 2, \ldots, C \tag{2}$$

where $\hat{\mathbf{C}}_k$ are the maximum likelihood estimates utilized by QDA.

(d) **[5 points]** Why do we require that $\rho_k \geq 0$ and $\mathbf{R}_k^T = \mathbf{R}_k \succeq 0$ for $k = 1, 2, \ldots, C$? Justify each one of these 3 constraints.

(e) **[5 points]** Under which circumstances would one want to use RQDA in lieu of QDA? Elaborate on such a such a scenario. Also, describe how would go about choosing the best value of the ρ_k's and why?
Hint(s): When is $\hat{\mathbf{C}}_k$ singular?

● Task 3. [30 total points]

Before you attempt this part, please understand well the MNR materials presented in class. Implement the MNR in `MATLAB` according to the following suggestions:

- Author the function `mnr_train( )` to implement the training phase of the MNR classifier. Use the following function signature `function [Weights, CEvalues, status] = mnr_train(Xtrain, Ltrain, maxIter, lambdaMax, alpha, tol)`, where `Xtrain` $\in \mathbb{R}^{N_{train} \times D}$ contains N_{train} training samples arranged in rows (each sample is D-dimensional) and `Ltrain` $\in \mathbb{R}^{N_{train} \times 1}$ contains the corresponding class labels in a single column. If C is the number of classes represented in `Xtrain`, you can safely assume that the labels in `Ltrain` and `Ltest` will belong to $\{1, 2, \ldots, C\}$. You should also assume that `Xtrain` always contains at least one sample from each one of the C classes. Training of the MNR should be performed via Gradient Descent (GD) using backtracking as a line search approach; if unsuccessful, bactracking should terminate after 100 failed attempts. Regarding the rest of the input arguments, `maxIter` $\in \mathbb{N}$ should specify the maximum number of GD iterations allowed, `lambdaMax` > 0 should specify the maximum step length (learning rate), `alpha` $\in (0, 1)$ should determine the ratio, by which the step length is reduced for each failed backtracking step and `tol` > 0 should specify the base-10 log value of the threshold determining if the gradient's L_∞ norm is low enough to declare convergence. Here, by gradient we mean the vector $\mathbf{g}$ containing the gradients of the cross-entropy with respect to all weights concatenated into a single column. Regarding the output arguments, `Weights` $\in \mathbb{R}^{C \times D}$ should contain the MNR's weight parameters for each class arranged in rows, `CEvalues` $\in \mathbb{R}^T$ should contain the cross-entropy values of the model for each GD iteration, assuming that the training stopped after T steps, and `status` should return 0, if training converged, or 1, if training just stopped due to performing the maximum number of GD iterations or due to backtracking not being able to further reduce the cross-entropy value of the model.

- Author the function `mnr_classify( )` to implement the testing/performance phase of the MNR classifier. Use the function signature `function [Lpred, Scores] = mnr_classify(Xtest, Weights)`, where `Xtest` $\in \mathbb{R}^{N_{test} \times D}$ contains N_{test} test samples arranged in rows. The return arguments are:

`Lpred` $\in \mathbb{R}^{N_{test}\times 1}$ are the labels of the test samples as predicted by the MNR classifier, while `Scores` $\in \mathbb{R}^{N_{test}\times C}$ contains the classifier's predicted posterior class probabilities $\hat{P}(\omega_i|\mathbf{x})$ for $i = 1, \ldots, C$ arranged in a row for each test sample.

A few comments and suggestions are in order, specifically pertaining to your MNR training implementation `mnr_classify( )`. It is strongly recommended that for each GD iteration you have MATLAB print out (i) the iteration number, (ii) the cross-entropy value at this iteration, (iii) the value of $\log_{10}\|\mathbf{g}\|_\infty$, (iv) the number of failed backtracking attempts. All this information will help you in troubleshooting your implementation and in choosing good values for `lambdaMax` and `alpha`. Item (ii) is useful, because you can affirm that the cross-entropy is non-increasing; if it occasionally increases, you are probably calculating the gradients incorrectly. Item (iii) is useful, because you can tell how close you are to converge given the `tol` value that you have specified. Finally, item (iv) is useful, because, if backtracking stops your training too early or makes too many attempts early on, this means that `lambdaMax` and/or `alpha` is too large. If backtracking consistently fails after 100 attempts at the very beginning of training, this may also point to an incorrectly calculation of gradients. After implementing these functions, address the following parts:

(a) **[15 points]** Use the training set S_{train} stemming from *IrisReduced* to train a MNR classifier and report back its LOOCV error. Also provide graphs showing the cross-entropy values versus the GD iteration numbers for two choices of maximum step length.

(b) **[15 points]** Show the decision regions of your MNR classifier obtained in Part (a) and compare them with the ones showcased in Part 1(d). Make appropriate comments.

Task 4. [20 total points]

(a) **[10 points]** Probability Density Functions (PDFs) or Probability Mass Functions (PMFs) of the form $f(\mathbf{x};\boldsymbol{\eta}) = g(\boldsymbol{\eta})h(\mathbf{x})\exp\left\{\boldsymbol{\eta}^T u(\mathbf{x})\right\}$, where h and u are some functions of $\mathbf{x}$ (h must be positive for all $\mathbf{x}$) and $g(\boldsymbol{\eta}) \triangleq 1/\int_{\mathbb{R}^D} h(\mathbf{x})\exp\left\{\boldsymbol{\eta}^T u(\mathbf{x})\right\} d\mathbf{x}$ is a normalizing constant, are said to belong to the *exponential family* of distributions. Assume now a classification problem involving C classes, where each class conditional distribution is the same exponential family PDF or PMF, but with different parameters $\boldsymbol{\eta}_k$'s and $u(\mathbf{x}) = \mathbf{x}$. Show that the Probabilistic Discriminative Model (PDM) arising from this setting is the MNR model.

(b) **[10 points]** Assume that, for the training of a MNR classifier, you add a penalty term of the form $r(\mathbf{W}) \triangleq \rho\sum_{k=1}^{C}\|\mathbf{w}_k\|_2^2$ to the cross-entropy, where $\rho \geq 0$ is a penalty parameter. Derive the GD-based update equation for the $\mathbf{w}_k$'s that stems from the new cost function. By adding this term, what happens to the weights? What is the point of considering this penalty term?

4 Preparation Guidelines

Below are some general guidelines that should be followed, when compiling a Mini-Project report. I strongly encourage you to stick to them, so that you receive full credit for your correct responses.

- **Task Statements:** Before attempting to address a particular task, ensure that you completely understand what is asked from you to perform and/or to produce. When in doubt, come to ask me for clarifications! Also, make sure you did not omit your response to any of the parts that you have attempted. Finally, make sure that it is crystal clear, which response corresponds to which task/part.

- **Material Presentation:** The material you generate for each task should be presented in your report in proper sequence by task and part number. If you have not attempted or completed a part, you need to indicate so at the appropriate location of your report.

- **Derivations & Proofs:** If you provide handwritten derivations and/or proofs, make sure you use your best handwriting. Each derivation should have a logical and organized flow, so that it is easy to follow and verify.

- **Code & Data:** The code that you author should be as well organized as possible and amply commented. This is very useful for assessing your work, as well as for you, while you are debugging/or modifying it, or if you have to go back to it in the near future. Driver scripts (scripts that may call other scripts or functions to accomplish a main task) should be named according to the part, for which they generate their numerical and/or graphical results; for example, the driver program for Task 1, Part (b) should be named **task1b.m**. Regarding the data you generate, keep them organize and document somehow (*e.g.*, in a text file) the specifics of how they were generated. **Caution:** You are not allowed to use any code and/or data that you have not produced without my explicit prior permission, in which case the sources you have obtained these from must be clearly indicated in your code or data description as well in your report. You are deemed to be plagiarizing, if you fail to do so, which may have dire consequences to your academic tenure here at Florida Tech!

- **Figures, Plots & Tables:** Plots should have their axes labeled and, if featuring several graphs, an appropriate legend should be used. Whether figures, plots or tables, each one of these elements should feature a caption with sufficient information on what is being displayed and how were these results obtained (*e.g.*, under what experimental conditions or settings, etc.). You should ask yourself the question: if someone comes across it, will they understand about what is being depicted? Apart from a concise description, major, relevant conclusions stemming from the display should also be included in the caption text.

- **Observations, Comments & Conclusions:** When stating observations about a particular result, do not stop at the obvious that anyone can notice (*e.g.*, "*... we see that the curve is increasing.*"). Instead, assess whether the result is expected, either by theory or intuition (*e.g.*, "*... This is as expected, because X is the integral of ...*"), or, if it is unexpected, offer a convincing reasoning behind it (*e.g.*, "*... We expected a decreasing curve ... All points to that I must have not been calculating X correctly ...*"). The latter is more preferable (*i.e.*, expect partial credit) than stopping at the obvious, which happens to be wrong (*i.e.*, do not expect partial credit). Next, descriptions and comments on results should be sufficient. Be concise, but complete. Finally, conclusions that you draw must be well-justified; vacuous conclusions will be swiftly discounted.

5 Submission Instructions

Kindly adhere to the conventions and submission instructions outlined below. Deviations from what is described here may cause unnecessary delays, costly oversights and immense frustrations related to the assessment of your hard work.

First, store all your Mini-Project deliverables in a folder named **lastname_mpX**, where "lastname" should be your last name and X should be the number of the Mini-Project, like 1, 2, etc. The folder name should be all lower case. For example, my folder for Mini-Project 1 would be named *anagnostopoulos_mp1*.

Secondly, your **lastname_mpX** folder should have the following contents:

- An Adobe PDF document named **lastname_report.pdf**, where, again, "lastname" should be replaced by your last name in all lower case, *e.g.*, *anagnostopoulos_report.pdf*. This document should contain your entire Mini-Project report as a single document. This will be the document that will be graded. Also, here are some important things to keep in mind:
  - The report must include a signed & dated copy of the Work Origination Certification page. You can either scan such a page and include it in your document, or sign and date it electronically, as long as your signature is not typed. If this page is missing from your report, or it does not comply with the aforementioned conditions, I reserve the right not to accept the report and assign a score of 0/100 for the relevant Mini-Project.
  - The Mini-Project may ask you to produce a variety of derivations, proofs, etc. You are not obliged to type such parts; it would be nice, but I realize that such effort would be quite time-consuming. Instead, you can import scanned images (or whole pages) of your handwritten work, as long as they are legible and well organized, so that the report has a clear logical flow. For example, it has to be clear where this hand-written work corresponds to (*e.g.*, which assignment it addresses).
  - Having said all this, you may want to consider to print out your typed work, appropriately merge it with any handwritten pages (don't forget the signed Work Origination Certification page!) and then scan the whole compilation into a single PDF report, say, in the Library. **Caution:** when scanning, use a relatively low-resolution (DPI) setting, so your resulting PDF document does not become too big in size, which may prevent you from uploading your work to Canvas.
- A folder named **src**, which should contain all your MATLAB scripts that you authored and used for producing your results and the data sets that you created for this Mini-Project, if applicable.
- An optional folder named **docs**, in which you can include a MS Word version of your report and other ancillary material connected in one way or another to your Mini-Project report.

Next, compress your **lastname_mpX** folder into a single ZIP archive named **lastname_mpX.zip**; *e.g.*, mine would be called *anagnostopoulos_mp1.zip*.

Finally, upload your ZIP archive to Canvas by the specified deadline using the appropriate drop box. You are done!

References

[1] Georgios C. Anagnostopoulos. LDA & QDA Classifires (Lecture Notes). http://courses.fit.edu, 2010. Accessed: 2016-10-22.

[2] K. Bache and M. Lichman. UCI machine learning repository, 2013. URL: http://archive.ics.uci.edu/ml.

[3] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification*. Wiley-Interscience, 2^{nd} edition, 2000.

[4] R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annual Eugenics*, 7:179–188, 1936. doi:10.1111/j.1469-1809.1936.tb02137.x.

[5] Wikipedia. Iris (plant). http://en.wikipedia.org/wiki/Iris_(plant). Accessed: 2016-10-22.

Work Origination Certification

By submitting this document, I, the author of this deliverable, certify that

1. I have reviewed and understood the Academic Honesty section of the current version of FITs Student Handbook available at http://www.fit.edu/studenthandbook/, which discusses academic dishonesty (plagiarism, cheating, miscellaneous misconduct, etc.)
2. The content of this report reflects my personal work and, in cases it is not, the source(s) of the relevant material has/have been appropriately acknowledged after it has been first approved by the courses instructor.
3. In preparing and compiling all this report material, I have not collaborated with anyone and I have not received any type of help from anyone but from the courses instructor.

Full Name (please PRINT)

Signature Date