Machine Learning Tutorial in Hiroshima Winter School

Junchi Liu, Amin Zarshenas, Kenji Suzuki
Illinois Institute of Technology, Chicago, USA

In this tutorial, you are going to learn to apply two fundamental machine-learning techniques, linear discriminant analysis (LDA) [1] and principal component analysis (PCA) [2], on 2 and 4 dimensional (2D and 4D) artificial data and 8 dimensional real colon classification data. You will learn 1) how those 2 machine-learning techniques work, 2) how they are different, 3) what are the advantages and disadvantages of them, and 4) how the characteristics and performance of them differ by using receiver-operating-characteristic analysis (ROC analysis).

I. Applying LDA and PCA on 2D artificial data

When you work with real high-dimensional data, things get messy. So let's build our intuition by first looking at some simpler artificial datasets.

We have created an artificial data having 2 features (2-dimensional data). Each data example is a vector $\boldsymbol{x} = (x_1, x_2)^T$. There are two classes, ω_1 and ω_2.

In real-life problems, we do not know the distributions of the data or their true statistics. But in this case, since we created the data by ourselves, we do know. For each class, the probability density function is a multivariate Gaussian. The mean vectors for the two classes are:

$$\mu_1 = E[\mathbf{x}|\omega_1] = (0,0)^T$$

$$\mu_2 = E[\mathbf{x}|\omega_2] = (2,2)^T$$

The covariance matrices for the two classes are identical:

$$\Sigma = \mathrm{cov}[\mathbf{x}|\omega] = (1,0;0,3)$$

a) First, try to use your own reasoning to find a good linear classifier for this problem. For this part, you may use what we have told you about the true distributions. Under this distribution, the data are ellipsoid full of points. Try to imagine these distributions, remembering that the mean vectors represent the centers of these ellipsoids. Try to guess (using geometrical reasoning) what vector **w** would produce good separation of the two classes, when used in a linear model where $y = f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + w_0$. What would be a logical choice for w_0?

b) Next you will use Octave to compute and study the Fisher LDA and PCA for this problem. For this analysis we are giving you a 2D dataset (in file `2D_artificial_dataset.mat`). This dataset contains 2 classes. Each class has 1,000 samples. Their mean and covariance matrices were described above.

1. Visualize (graph) the data.
   Run program 'Visualize_2Ddata.m' to visualize the 2-D data in scatter plots.
   Looking at the scatter plot, we can see that the two classes have good separation when viewed from certain points of view, but no separation at all in other directions. Relate these graphs to your original thinking in part (a).

   *Note:1. Do not close the plot until the end of this session. You will need to draw the best **w** and the decision boundary for LDA and PCA on this scatter plot. 2. On different*

computers, the display of the samples (dots '.' and stars '') in plot may look different. You may need to adjust the font size of these samples in code. Or simply zoom in and zoom out using the plot tools in the plot toolbar.*

2. Compute the Fisher discriminant vector **w** using the eigenvector equation:

$$(S_W^{-1}S_B)\mathbf{w} = J(\mathbf{w})\mathbf{w}$$

Remember that the best **w** is the eigenvector corresponding to the largest eigenvalue, because the eigenvalue is the Fisher ratio $J(\mathbf{w})$, which is a kind of signal-to-noise ratio.

To help with this, we are providing you with a function (`eigsort.m`) that outputs the eigenvectors in descending order of eigenvalue. Thus, the first eigenvector is the one with the largest eigenvalue. What **w** did you get? Is this what you expected? (Run program 'LDA_2Ddata.m' to get the best **w** and the decision boundary for LDA.)

Note: 1. The sign of the eigenvectors you calculate is not important. For example, (1,0) indicates the same axis as (-1,0). 2. When the code needs to output a value (for example eigen values) in the command window while running, you need to go to the command window and press 'f' key to proceed or press 'q' to run until the end of the code. Otherwise the code won't proceed.

3. Repeat step 2 for PCA. Run program 'PCA_2Ddata.m' to get the best **w** and the decision boundary for PCA. Compare the differences between LDA and PCA in terms of the best **w** and the decision boundary.

II. Applying LDA and PCA on 4D artificial data

Now we are moving to a higher dimensional data. We have created an artificial dataset containing two classes with 4 features (in file `4D_artificial_dataset.mat`). Each class has 1,000 samples. The mean vector for class 1 and class 2 are $(0,0,0,0)^T$ and $(2,2,0,0)^T$ respectively. The covariance matrices for the two classes are identical, $\Sigma = \text{cov}[\mathbf{x}|\omega] = (1,0,0,0; 0,3,0,0; 0,0,1,0; 0,0,0,1)$.

Next you will use Octave to compute and study the Fisher discriminant and PCA for this problem.

1. Visualize (graph) the data. When data are more than 3-dimensions, you cannot view them directly. One method of displaying high-dimensional data is the "trellis plot" – a matrix of scatter plots, with each scatter plot in the matrix graphing one of the variables against another. For example, the scatter plot in the (2,1) position in the matrix plots x_2 against x_1. You can think of each plot as the projection of the 4D scatter plot onto a different plane, as if we were viewing the clouds of points from different directions (along different coordinate axes). Run program 'Visualize_4Ddata.m' to visualize the dataset. Looking at the scatter plots, we can see that the two classes have good separation when viewed from certain points of view, but no separation at all in other directions. In which plot(s) do you see the best separation?

2. Compute the Fisher discriminant vector **w** using the eigenvector equation we learned in the previous section. What **w** did you get? What Fisher ratio did you find? (Run program 'LDA_4Ddata.m' to get the best **w** and the decision boundary for LDA based on the first two features.)

3. Generating histograms:
   a) Compute $\mathbf{w}^T\mathbf{x}$ for each example data point.
   b) Plot two histograms of $\mathbf{w}^T\mathbf{x}$ (one histogram for each class) on a single graph. The result should be two overlapping Gaussian-shaped functions (but they will be noisy, not smooth like a Gaussian).
   Run program 'LDA_4Ddata_histogram.m' to get the histogram.

4. Repeat step 2 and 3 for PCA. Run program 'PCA_4Ddata.m' to get the best **w** and the decision boundary for PCA based on the first two features. Run program 'PCA_4Ddata_histogram.m' to get the histogram.

5. ROC analysis:
   a) Let's apply ROC analysis to measure the performance of LDA and PCA. Assume x1 in the dataset "negative/No" class and x2 in the dataset "positive/Yes" class. Which method produces the higher ROC curve? Is this what you expect?
   b) It is sometimes convenient to summarize the performance by one number, instead of an ROC curve. To do this, it is common to use the area under the ROC curve (AUC), which is often denoted as A_z. Calculate AUC for each of your ROC curves from part (a). You can use the simple rectangle method [3] to perform the numerical integration. Which method produces the higher AUC value? Is this what you expect? What is the maximum possible AUC value that can occur?
   (Run program 'LDA_ROC_4D.m' and 'PCA_ROC_4D.m' to draw the ROC curve and calculate the AUC value for LDA and PCA respectively.)

III. Applying LDA and PCA on colon classification data

Now it is time to move on to real clinical data [4,5]. We prepared an 8-feature colon classification dataset (file '8D_colon_dataset.mat'). This dataset contains two classes. Class 1 is polyp class consisting of 46 samples. Class 2 is non-polyp class consisting of 500 samples. The 8 features, listed in table 1, are selected from the 79 features in the original dataset with the goal of maximizing the AUC. A sophisticated feature selection algorithm called Binary Coordinate Ascent (BCA) [6] was applied for this task. An alternative feature selection is the Max-AUC feature selection [4].

1. Visualize (graph) the data. Use "trellis plot" to visualize the high dimensional data. In this case there will be 64 plots in the matrix. Run program 'Visualize_8Dcolondata.m' to visualize the dataset. Looking at the scatter plots. In which plot(s) do you see the best separation?

2. Compute the Fisher discriminant vector **w** using the eigenvector equation we learned in the previous section. What **w** did you get? What Fisher ratio did you find? (Run program 'LDA_8Dcolondata.m' to get the best **w** and the decision boundary for LDA based on the first and the eighth feature.)

3. Generating normalized histograms:

a) Compute $\mathbf{w}^T\mathbf{x}$ for each example data point.
b) Plot two histograms of $\mathbf{w}^T\mathbf{x}$ (one histogram for each class) on a single graph. The result should be two overlapping Gaussian-shaped functions (but they will be noisy, not smooth like a Gaussian).
Run program 'LDA_8Dcolon_histogram.m' to get the normalized histogram.

4. Repeat step 2 and 3 for PCA. Run program 'PCA_8Dcolondata.m' to get the best **w** and the decision boundary for PCA based on the first and the eighth feature. Run program 'PCA_8Dcolon_histogram.m' to get the histogram.

5. ROC analysis:
   a) Let's apply ROC analysis to measure the performance of LDA and PCA on the colon data. Assume x1 in the dataset "positive/Yes" class and x2 in the dataset "negative/No" class. Which method produces the higher ROC curve?
   b) Calculate AUC for each of your ROC curves from part (a). Which method produces the higher AUC value? Is this what you expect?
   (Run program 'LDA_ROC_8Dcolon.m' and 'PCA_ROC_8Dcolon.m' to draw the ROC curve and calculate the AUC value for LDA and PCA.)

Table 1 – List of 8 features used in this experiment

Feature 1	Standard deviation of gray levels inside the 2D candidate
Feature 2	Radial gradient index (RGI) outside the 2D candidate
Feature 3	Tangential gradient index (TGI) outside the 2D candidate
Feature 4	Matsushita distance of normalized histograms in the gray scale image
Feature 5	Mode of the histogram inside the lesion candidate in the gray scale image
Feature 6	Maximum of the histogram inside the lesion candidate in the gray scale image
Feature 7	Maximum of the histogram outside the lesion candidate in the gray scale image
Feature 8	Mean voxel intensity in the Sobel image

III. Summary

In this big data era, machine learning is indispensable in any applications from medical to robotics to the Internet to financial, to automobiles. This tutorial covers only the fundamentals of machine learning and hands-on experience on basic classification data. Once you lean the fundamentals, however, you can apply your fundamental knowledge and experience to solve more complicated problems with more advanced machine-learning techniques.

For those who are interested in such advanced machine learning, we recommend reading the following textbooks and papers from my group. Examples of recommended textbooks include Haykin's textbook [7] that is considered a bible of artificial neural networks (ANNs) that comprehensively covers ANNs, Bishop's textbook [8] that is a good textbook for ANNs in a pattern recognition aspect, and Vapnik's textbook [9] that convers the theory of support vector

machines. In addition, Suzuki's reference books [10-11] cover the recent advances in machine learning and computational intelligence in the medical imaging area. Advanced machine-learning models that can learn pixels in images directly from our group [12-18], now people call similar approaches deep learning or deep convolutional neural networks, would be very useful to learn, as such machine-learning models are revolutionizing many fields including computer vision and medical imaging.

Reference

[1] https://en.wikipedia.org/wiki/Linear_discriminant_analysis
[2] https://en.wikipedia.org/wiki/Principal_component_analysis
[3] http://en.wikipedia.org/wiki/Rectangle_method
[4] Xu J., and Suzuki K.: Max-AUC Feature Selection in Computer-aided Detection of Polyps in CT Colonography. IEEE Journal of Biomedical and Health Informatics 18: 585-593, 2014
[5] Suzuki K., Zhang J., and Xu J.: Massive-training artificial neural network coupled with Laplacian-eigenfunction-based dimensionality reduction for computer-aided detection of polyps in CT colonography. IEEE Transactions on Medical Imaging 29: 1907-1917, 2010.
[6] Zarshenas A, Suzuki K (2016) Binary coordinate ascent: An efficient optimization technique for feature subset selection for machine learning. Knowledge-Based Syst 110:191–201.
[7] Haykin S: Neural Networks a comprehensive foundation. Prentice Hall, NJ (1999).
[8] Bishop CM: Neural Networks for Pattern Recognition. Oxford University Press, Oxford (1995).
[9] Vapnik V: The Nature of Statistical Learning Theory. Springer Verlag, New York (1995).
[10] Suzuki K.: Editor. Computational Intelligence in Biomedical Imaging, Springer (New York, NY), 411pp., 2014. (ISBN 978-1-4614-7244-5)
[11] Suzuki K.: Editor. Machine Learning in Computer-Aided Diagnosis: Medical Imaging Intelligence and Analysis, IGI Global (Hershey, PA), 524 pp., 2012. (ISBN 9781466600591)
[12] Suzuki K., Armato III S. G., Li F., Sone S., and Doi K.: Massive training artificial neural network (MTANN) for reduction of false positives in computerized detection of lung nodules in low-dose CT. Medical Physics 30: 1602-1617, 2003.
[13] Suzuki K., Horiba I., and Sugie N.: Neural edge enhancer for supervised edge enhancement from noisy images. IEEE Transactions on Pattern Analysis and Machine Intelligence 25: 1582-1596, 2003.
[14] Suzuki K., Horiba I., Sugie N., and Nanki M.: Extraction of left ventricular contours from left ventriculograms by means of a neural edge detector. IEEE Transactions on Medical Imaging 23: 330-339, 2004.
[15] Suzuki K., Li F., Sone S., and Doi K.: Computer-aided diagnostic scheme for distinction between benign and malignant nodules in thoracic low-dose CT by use of massive training artificial neural network. IEEE Transactions on Medical Imaging 24: 1138-1150, 2005.
[16] Suzuki K., Abe H., MacMahon H., and Doi K.: Image-processing technique for suppressing ribs in chest radiographs by means of massive training artificial neural network (MTANN). IEEE Transactions on Medical Imaging 25: 406-416, 2006.
[17] Suzuki K.: Supervised "lesion-enhancement" filter by use of a massive-training artificial neural network (MTANN) in computer-aided diagnosis (CAD). Physics in Medicine and Biology 54: S31-S45, 2009.

[18] Suzuki K., Zhang J., and Xu J.: Massive-training artificial neural network coupled with Laplacian-eigenfunction-based dimensionality reduction for computer-aided detection of polyps in CT colonography. IEEE Transactions on Medical Imaging 29: 1907-1917, 2010.