Project: DATA620013

Jan 14th, 2019

1 Guideline

The files digits.RData contain the same training and test data of handwritten digit images for R. There are 10 digits classes. Each digit image $x \in \{0,1\}^{20\times20}$ is 20×20 pixels in size and consists of binary values 0's and 1's only. There are 500 training images and 1000 test images per digit class. The whole training set is stored in a $10 \times 500 \times 20 \times 20$ array. The goal of this project is to experiment with different classifiers for this data set: 1) LDA, 2) Mixture of Bernoulli's. For each classifier you will report the error rate on the test set.

1.1 Tuning parameter selection

For all classifiers, there will be a free parameter that needs to be adjusted using a held out part of the training set, i.e., cross-validation. The tuning parameters are:

- LDA: λ - smoothing parameter for covariance matrix
- Mixture: number of components M for the mixture model
- Logistic regression: regularization parameter λ

For these methods, run the algorithm FIVE times on a random subset of 400 training examples per class, and evaluate the error on the remaining 100 examples per class, for a range of values of the free parameter, to obtain the value that minimizes the average error rate over the five runs. Once you choose this value, check the resulting algorithm on the test set

1.2 Writing up the report

Please turn in your code for each problem with detailed in-line documentation, and for every run of the EM algorithm, show the value of the log-likelihood at each iteration.
Here is a template:

- Answer the questions associated with the method.
- Fix a choice of the tuning parameter, provide the code for fitting the method on the training set.
- Show the cross-validation error for every value of the tuning parameter on a grid you have chosen. (The code for cross-validation can be skipped, but you have to describe in text how you did it).

- Select the tuning parameter, present the final classifier (in equations and illustrating figures, see details below); if it is an iterative algorithm, show the output for a few iterations.
- Report the error rate on the test set.

1.3 Useful R code

Below is some R code to load the dataset and visualize one digit:

```
load('digits.RData')
windows()
image(t(1-training.data[3,1,,])[,20:1],
        col=gray(seq(0, 1, length.out=256)),
        axes=FALSE, asp=1)
```

Below is some R code to set up the data for use in the R classification packages

```
# Number of classes
num.class <- dim(training.data)[1]
# Number of training data per class
num.training <- dim(training.data)[2]
# Dimension of each training image (rowsxcolumns)
d <- prod(dim(training.data)[3:4])
# Number of test data
num.test <- dim(test.data)[2]
# Reshape training data to 2-dim matrix
dim(training.data) <- c(num.class * num.training, d)
# Same for test.
dim(test.data) <- c(num.class * num.test, d)
# Labels of training data.
training.label <- rep(0:9, num.training)
# Labels of test data
test.label <- rep(0:9, num.test)
```

2 Problems

Below, a feature vector X_i is a vector in dimension 400 by converting the 20×20 image to a long vector.

2.1 LDA

Consider a generative model: the observations $(X_i, y_i) \in \mathbb{R}^p \times \{1, ..., K\}$, $1 \leqslant i \leqslant n$ are i.i.d. samples from a Gaussian mixture $\sum_{k=1}^{K} \pi_k \mathcal{N}(\mu_k, \Sigma)$, i.e., $P(y_i = k) = \pi_k$ and $X_i \mid (y_i = k) \sim \mathcal{N}(\mu_k, \Sigma)$

- (a) Show that the Bayes classifier has the form that $h(x) = argmax_{1 \leqslant k \leqslant K} \{x^\top \beta_k + \beta_{0k}$. Give the explicit expression of $\{\beta_{0k}, \beta_k\}_{k=1}^K$.

- (b) Write down the estimators for μ_k and Σ

- Let $\hat{\Sigma}$ be the pooled sample covariance matrix. If it is (close to) singular, then consider using $\hat{\Sigma}_\lambda = (1-\lambda)\hat{\Sigma} + (\lambda/4)I_p$ for some small $\lambda \in (0,1)$. Why must $\hat{\Sigma}_\lambda$ be invertible even if $\hat{\Sigma}$ is singular?

- (d) Implement LDA.

2.2 Mixture of Independent Bernoullis

We fit a mixture model using the training data for each class. The mixture model has M components and each component a product of $D = 400$ independent Bernoulli variables

$$p(X_i) = \sum_{m=1}^{M} \pi_m \cdot p(X_i \mid \mu_m) = \sum_{m=1}^{M} \pi_m \left(\prod_{j=1}^{D} \mu_{mi}^{X_{ij}} (1-\mu_{mi})^{1-X_{ij}} \right)$$

where $\mu_m = (\mu_{m1}, ..., \mu_{mD})$ contains the Bernoulli parameters for each component m. We train each digit class separately, and the M can be different.

In the general EM framework, we let X be the observed variables, Z be the latent variables and θ be the parameters to maximize. Before, we used the EM algorithm to maximize $\log p(X \mid \theta)$ over θ, where in the M-step, we maximize

$$Q(\theta, \theta^{old}) = \int p(Z \mid X, \theta^{old}) \log p(X, Z \mid \theta) dZ$$

Now we put a prior $p(\theta)$ and want to find the MAP (maximum posterior) solution of θ, i.e., to maximize $\log p(\theta \mid X)$ over θ. This can be done by modifying the M-step to maximize

$$Q(\theta, \theta^{old}) + \log p(\theta)$$

- (a) Use a $Beta(2,2)$ prior for the parameters μ_{mj}, and $Dirichlet(2, ..., 2)$ for the parameter π_m. Derive the EM algorithm for the MAP of μ_{mj} and π_m.

- (b) Initialize the EM as follows. Assign each example at random to one of the M components. Now you have complete data and use them to estimate μ_{mj} and π_m. Use these estimates as initial values. Provide details about how you do this.

- (c) When computing $\gamma(z_{im}) = \frac{\pi_m p(X_i \mid \mu_m)}{\sum_{\alpha=1}^{M} \pi_\alpha p(X_i \mid \mu_\alpha)}$, first compute the log of each of the terms in the denominator sum. Call them $l_m(X_i)$, $m = 1, ..., M$. Find the largest one l^*. Then compute $\gamma(z_{im})$ in this way: $\gamma(z_{im}) = \frac{\exp(l_m - l^*)}{\sum_{\alpha=1}^{M} \exp(l_\alpha - l^*)}$. Can you explain why this is important?

- (d) Select one digit class and fit the mixture model for $M = 2, 3, 5$. For each M, after you get the MAP of $\{\mu_{mj} : 1 \leqslant m \leqslant M, 1 \leqslant j \leqslant D\}$ and $\{\pi_m : 1 \leqslant m \leqslant M\}$, plot the image for each of the M components by reshaping $(\mu_{m1}, ..., \mu_{mD})$, with $D = 400$, to a 20×20 array. What do you see?

- (e) Take the training data of two digit classes, and fit a mixture model for each of them (you can pick one M heuristically without cross-validation). How are you going to use them for classification? Compute the test error of your classifier on the testing data of the two selected digit classes.