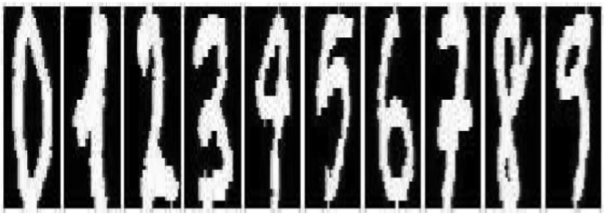
ST758, Homework 6

Due Nov 6, 2014

In this homework, we work on a model-based method for handwritten digit recognition. Following figure shows example bitmaps of handwritten digits from U.S. postal envelopes.



Each digit is represented by a 32×32 bitmap in which each element indicates one pixel with a value of white or black. Each 32×32 bitmap is divided into blocks of 4×4, and the number of white pixels are counted in each block. Therefore each handwritten digit is summarized by a vector $\boldsymbol{x} = (x_1, \ldots, x_{64})$ of length 64 where each element is a count between 0 and 16.

By a model-based method, we mean to impose a distribution on the count vector and carry out classification using probabilities. A common distribution for count vectors is the multinomial distribution. However as you will see in Q11, it is not a good model for handwritten digits. Let's work on a more flexible model for count vectors. In the Dirichlet-multinomial model, we assume the multinomial probabilities $\boldsymbol{p} = (p_1, \ldots, p_d)$ follow a Dirichlet distribution with parameter vector $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_d)$, $\alpha_j > 0$, and density

$$\pi(\boldsymbol{p}) = \frac{\Gamma(|\boldsymbol{\alpha}|)}{\prod_{j=1}^d \Gamma(\alpha_j)} \prod_{j=1}^d p_j^{\alpha_j - 1},$$

where $|\boldsymbol{\alpha}| = \sum_{j=1}^d \alpha_j$.

1. For a multivariate count vector $\boldsymbol{x} = (x_1, \ldots, x_d)$ with batch size $|\boldsymbol{x}| = \sum_{j=1}^d x_j$, show that the probability mass function for Dirichlet-multinomial distribution is

$$f(\boldsymbol{x} \mid \boldsymbol{\alpha}) = \int_{\Delta_d} \binom{|\boldsymbol{x}|}{\boldsymbol{x}} \prod_{j=1}^d p_j^{x_j} \pi(\boldsymbol{p})\, d\boldsymbol{p} \quad = \quad \binom{|\boldsymbol{x}|}{\boldsymbol{x}} \frac{\prod_{j=1}^d (\alpha_j)_{x_j}}{(|\boldsymbol{\alpha}|)_{|\boldsymbol{x}|}},$$

   where Δ_d is the unit simplex in d dimensions, $|\boldsymbol{\alpha}|$ equals $\sum_{j=1}^d \alpha_j$, and $(a)_k = \prod_{i=0}^{k-1}(a+i)$ denotes a rising factorial. (Hint: $\Gamma(a+k)/\Gamma(a) = (a)_k$.)

2. Given independent data points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$, show that the log-likelihood is

$$\begin{aligned} L(\boldsymbol{\alpha}) &= \sum_{i=1}^n \ln \binom{|\boldsymbol{x}_i|}{\boldsymbol{x}_i} + \sum_{i=1}^n \sum_{j=1}^d \sum_{k=0}^{x_{ij}-1} \ln(\alpha_j + k) - \sum_{i=1}^n \sum_{k=0}^{|\boldsymbol{x}_i|-1} \ln(|\boldsymbol{\alpha}| + k) \\ &= \sum_{i=1}^n \ln \binom{|\boldsymbol{x}_i|}{\boldsymbol{x}_i} + \sum_{i=1}^n \sum_{j=1}^d [\ln \Gamma(\alpha_j + x_{ij}) - \ln \Gamma(\alpha_j)] - \sum_{i=1}^n [\ln \Gamma(|\boldsymbol{\alpha}| + |\boldsymbol{x}_i|) - \ln \Gamma(|\boldsymbol{\alpha}|)]. \end{aligned}$$

Is the log-likelihood a concave function?

3. Write an `R` function to compute the density and/or log-density of the Dirichlet-multinomial distribution. The interface should be `ddirmult(x, alpha, log = FALSE)`. Vectorize your code. The input `x` and `alpha` are allowed to be n-by-d matrices. In this case, the function should return a vector of Dirichlet-multinomial probabilities $f(\boldsymbol{x}_i|\boldsymbol{\alpha}_i)$ (if `log = FALSE`) or log-probabilities $\ln f(\boldsymbol{x}_i|\boldsymbol{\alpha}_i)$ (if `log = TRUE`) for $i = 1, \ldots, n$. Feel free to use `lgamma()`, `digamma()`, `trigamma()` functions in R. They are vectorized and efficient.

4. Read in `optdigits.tra`, the training set of 3823 handwritten digits. Each row contains the 64 counts of a digit and the last element (65th element) indicates what digit it is. For grading purpose, evaluate the total log-likelihood of this data at parameter values $\boldsymbol{\alpha} = (1, \ldots, 1)$ using your function `ddirmult()`.

5. Derive the score function $\nabla L(\boldsymbol{\alpha})$, observed information matrix $-d^2 L(\boldsymbol{\alpha})$, and expected Fisher information matrix $\mathbf{E}[-d^2 L(\boldsymbol{\alpha})]$ for the Dirichlet-multinomial distribution.

6. Comment on why Fisher scoring method is inefficient for computing MLE in this example.

7. What structure does the observed information matrix possess that can facilitate the evaluation of the Newton direction? Is the observed information matrix always positive definite? What remedy can we take if it fails to be positive definite?

8. Discuss how to choose a good starting point. Implement this as the default starting value in your function below. (Hint: Method of moment estimator may furnish a good starting point.)

9. Write a function for finding MLE of Dirichlet-multinomial distribution given iid observations $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$, using safeguarded Newton's method. The interface should be `dirmultfit(x, alpha0 = NULL, maxiters = 100, tolfun = 1e-6)`. The arguments are: `x` a n-by-d matrix of counts, `alpha0` a d vector of starting point (optional), `maxiters` the maximum allowable Newton iterations (default 100), `tolfun` the tolerance for relative change in objective values (default 1e-6). The return value should be a list containing: `maximum` the log-likelihood at MLE, `estimate` the MLE, `gradient` the gradient at MLE, `hessian` the Hessian at MLE, `se` a d vector of standard errors, `iterations` the number of iterations performed.

10. Read in `optdigits.tra`, the training set of 3823 handwritten digits. Find the MLE for the subset of digit 0, digit 1, ..., and digit 9 separately using your function. We denote these MLEs $\hat{\boldsymbol{\alpha}}_0, \ldots, \hat{\boldsymbol{\alpha}}_9$.

11. As $\boldsymbol{\alpha}/|\boldsymbol{\alpha}| \to \boldsymbol{p}$, the Dirichlet-multinomial distribution converges to a multinomial with parameter $\boldsymbol{p}$. Therefore multinomial can be considered as a special Dirichlet-multinomial with $|\boldsymbol{\alpha}| = \infty$. Perform a likelihood ratio test (LRT) whether Dirichlet-multinomial offers a better fit than multinomial for digits 0, 1, ..., 9 respectively.

12. Now we can construct a simple Bayesian rule for handwritten digits recognition:

$$\boldsymbol{x} \mapsto \arg\max_k \hat{\pi}_k f(x|\hat{\boldsymbol{\alpha}}_k).$$

Here we can use the proportion of digit k in the training set as the prior probability $\hat{\pi}_k$. Report the performance of your classifier on the test set of 1797 digits in `optdigits.tes`.