How to use the MiPP Package

Mat Soukup, HyungJun Cho, and Jae K. Lee

September 9, 2005

Contents

1 Introduction

The *MiPP* package is designed to sequentially add genes to a classification gene model based upon the Misclassification-Penalized Posteriors (MiPP) as discussed in Section 2. The construction of the model is based upon a training data set and the estimated actual performance of the model is based upon an independent data set. When no clear distinction between the training and independent data sets exists, the cross-validation technique is used to estimate actual performance. For the detailed algorithms, see Soukup, Cho, and Lee (2005) and Soukup and Lee (2004). The *MiPP* package employs libraries *MASS* for LDA/QDA (linear/quadratic discrimant analysis) and *e1071* for SVM (support vector machine). Users should install the *e1071* package from the main web page of R (http://www.r-project.org/).

2 Misclassification-Penalized Posteriors (MiPP)

In the above section, estimated actual performance is mentioned a number of times. Classically, the accuracy of a classification model is done by reporting its estimated actual error rate. However, error rate fails to take into account how likely a particular sample belongs to a given class and dichotomizes the data into yes the sample was

correctly classified or no the sample was NOT correctly classified. Although error rate, plays a key role in how well a classification model performs, it fails to take into account all the information that is available from a classification rule.

The Misclassification-Penalized Posteriors (MiPP) takes into account how likely a sample belongs to a given class by using a posterior probability of correct classification. MiPP also adjusts its definition any time a sample is misclassified by subtracting a 1 from the posterior probability of correct classification resulting in a negative value of MiPP. If we define the posterior probability of correct classification using genes $\mathbf{x}$ as $\hat{f}(\mathbf{x})$, MiPP can be calculated as

$$\psi_p = \sum_{correct} \hat{f}(\mathbf{x}) + \sum_{wrong} \left(\hat{f}(\mathbf{x}) - 1 \right). \tag{1}$$

Here, *correct* refers to the subset of samples that are correctly classified and *wrong* refers to the subset of samples that are misclassified. By introducing a random variable that takes into account whether a sample is misclassified or not MiPP can be shown to be the sum of posterior probabilities of correct classification minus the number of misclassified samples. As a result, MiPP increases whenever the sum of posterior probabilities of correction classification increase, the number of misclassified samples decreases, or both of these occur.

We standardize the MiPP score divided by the number of samples in each data set, denoted as sMiPP. Thus, the range of sMiPP is from -1 to 1. Note that as accuracy increases, sMiPP converges to 1.

Some basic properties of MiPP are that the maximum value it can take is equal to the sample size (or $sMiPP = 1$), and on the flip side, the minimum value is equal to the negation of the sample size (or $sMiPP = -1$). Under a pure random model, the expected value of MiPP is equal to zero (or $sMiPP = 0$). The variance is derived and is available from the first author for the two class case, however an explicit value for more than two classes can not be derived analytically. Thus, a bootstrapped estimate is the preferred method of estimating the variance.

3 Examples

3.1 Acute Leukemia Data

This data set has been frequently used for testing various methods in classification and prediction of cancer sub-types. Two distinct subsets of array data for AML and ALL leukemia patients are available: a training set of 27 ALL and 11 AML samples and a test set of 20 ALL and 14 AML samples. The independent set was from adult bone marrow samples, whereas the independent set was from 24 bone marrow samples, 10 from peripheral blood samples, and 4 of the AML samples from adults. Gene expression levels contain probes for 6817 human genes from Affymetrix™oligonucleotide microarrays. Note that a subset of genes (713 probe sets) was stored into the *MiPP* package.

To run *MiPP*, the data can be prepared as follows.

```
data(leukemia)

#IQR normalization
leukemia <- cbind(leuk1, leuk2)
leukemia <- mipp.preproc(leukemia, data.type="MAS4")

#Train set
x.train <- leukemia[,1:38]
y.train <- factor(c(rep("ALL",27),rep("AML",11)))
#Test set
x.test <- leukemia[,39:72]
y.test <- factor(c(rep("ALL",20),rep("AML",14)))
```

Since two distinct data sets exist, the model is constructed on the training data and evaluated on the test data set as follows.

```
out <- mipp(x=x.train, y=y.train, x.test=x.test, y.test=y.test,
            n.fold=5, percent.cut=0.05, rule="lda")
```

This sequentially selects genes one gene at a time with the LDA rule (*rule="lda"*) and 5-fold cross-validation (*n.fold=5*) on the training set. To reduce computing time, it pre-selects the most plausuable 5% out of 713 genes by the two-sample t-test (*percent.cut=0.05*), and then performs gene selection. To utilize all genes without pre-selection, set the argument *percent.cut=1*. The above command generates the following output.

```
out$model
```

	Order	Gene	Tr.ER	Tr.MiPP	Tr.sMiPP	Te.ER	Te.MiPP	Te.sMiPP	Select
1	1	571	0.0526	30.86	0.8122	0.1176	23.92	0.7035	
2	2	436	0.0000	36.89	0.9707	0.0294	30.41	0.8945	
3	3	366	0.0000	37.95	0.9988	0.0294	31.35	0.9222	
4	4	457	0.0000	38.00	0.9999	0.0294	32.14	0.9453	
5	5	413	0.0000	38.00	1.0000	0.0294	32.18	0.9464	
6	6	635	0.0000	38.00	1.0000	0.0000	33.75	0.9927	**
7	7	648	0.0000	38.00	1.0000	0.0000	33.57	0.9874	

The gene models are evaluated by both train (denoted by **Tr**) and test (denoted by **Te**) sets; however, we select the final model based on the test set independent of the train

set used for gene selection. The gene model with the maximum sMiPP is indicated by one star (*) and the parsimonious model (indicated by **) contains the fewest number of genes with sMiPP greater than or equal to (max sMiPP - 0.01). In this example, the maximum and parsimonious models (indicated by **) are the same. Thus, the final model with sMiPP 0.993 contains genes 571, 436, 366, 457, 413, and 635. Note that genes listed in the output correspond to the column number of the matrices.

3.2 Colon Cancer Data

The colon cancer data set consists of the 2000 genes with the highest minimal intensity across the 62 tissue samples out of the original $6,500^{+}$ genes. The data set is filtered using the procedures described at the author's web site. The 62 samples consist of 40 colon tumor tissue samples and 22 normal colon tissue samples (Alon *et al.*, 1999). Li *et al.* (2001) identified 5 samples (N34, N36, T30, T33, and T36) which were likely to have been contaminated. As a result, these five samples are excluded from any future analysis; our error rate would be higher if they were included.

Since we are working with a small data set (57 samples), we will be implementing cross-validation techniques. With the lack of a 'true' independent test set, we randomly create a training data set with 38 samples (25 tumor and 13 normal) and an independent data set with 19 samples (12 tumor and 7 normal). Since this is a random creation of the data set, it would be of interest to see what model is selected based upon a different random split of the data. Note that the choice of the sizes of the training and independent test set is somewhat arbitrary, but consistent results were found using a training and test set of sizes 29 (19 tumor and 10 normal) and 28 (18 tumor and 10 normal), respectively. The colon data set of the *MiPP* package contains only 200 genes as an example. For the colon data with no independent test set, *MiPP* can be run as follows.

```
data(colon)
x <- mipp.preproc(colon)
y <- factor(c("T", "N", "T", "N", "T", "N", "T", "N", "T", "N",
       "T", "N", "T", "N", "T", "N", "T", "N", "T", "N",
       "T", "N", "T", "N", "T", "T", "T", "T", "T", "T",
       "T", "T", "T", "T", "T", "T", "T", "T", "N", "T",
       "T", "N", "N", "T", "T", "T", "T", "N", "T", "N",
       "N", "T", "T", "N", "N", "T", "T", "T", "T", "N",
       "T", "N"))



#Deleting comtaminated chips
x <- x[,-c(51,55,45,49,56)]
y <- y[ -c(51,55,45,49,56)]
```

```
out <- mipp(x=x, y=y, n.fold=5, p.test=1/3, n.split=20, n.split.eval=100,
            percent.cut = 0.1 , rule="lda")
```

This divides the whole data into two groups for training (two-third) and testing (one-third) (*p.test = 1/3*) and performs the forward gene selection as done with the acute leukemia data. Splitting of the data set into training and independent dat seta and then selecting a model for a given split are repeated 20 times (*n.split=20*). This generates the following output.

```
out$model
```

	Split	Order	Gene	Tr.ER	Tr.MiPP	Tr.sMiPP	Te.ER	Te.MiPP	Te.sMiPP	Select
1	1	1	29	0.0526	32.37	0.8517	0.0526	16.73	0.8806	
2	1	2	177	0.0000	36.72	0.9664	0.0000	18.84	0.9917	**
3	1	3	163	0.0000	37.79	0.9945	0.0000	18.95	0.9974	
4	1	4	36	0.0000	37.97	0.9992	0.0000	18.99	0.9994	*
5	1	5	148	0.0000	37.99	0.9997	0.0000	18.97	0.9987	
6	1	6	78	0.0000	38.00	0.9999	0.0000	18.90	0.9947	
7	1	7	84	0.0000	38.00	0.9999	0.0000	18.93	0.9961	
8	1	8	18	0.0000	38.00	0.9999	0.0000	18.99	0.9994	
9	1	9	141	0.0000	38.00	1.0000	0.0000	18.99	0.9994	
10	2	1	29	0.0526	34.22	0.9005	0.1579	13.46	0.7082	
11	2	2	102	0.0000	37.82	0.9952	0.1053	15.35	0.8079	
12	2	3	36	0.0000	37.89	0.9971	0.0526	16.56	0.8718	
13	2	4	18	0.0000	37.94	0.9983	0.1053	15.31	0.8060	
14	2	5	49	0.0000	37.99	0.9998	0.0526	17.09	0.8993	**
15	2	6	78	0.0000	38.00	1.0000	0.0526	16.79	0.8835	
16	2	7	148	0.0000	38.00	1.0000	0.0526	16.58	0.8726	
17	2	8	65	0.0000	38.00	1.0000	0.0526	16.97	0.8929	
18	2	9	95	0.0000	38.00	1.0000	0.0526	17.08	0.8990	
.										
.										
.										
172	20	1	30	0.0263	34.15	0.8987	0.1579	13.11	0.6902	
173	20	2	102	0.0000	37.58	0.9890	0.1053	15.32	0.8065	
174	20	3	36	0.0000	37.85	0.9961	0.0000	18.43	0.9698	**
175	20	4	18	0.0000	37.88	0.9969	0.0000	18.40	0.9685	

```
176    20     5  177 0.0000   37.94   0.9984 0.0526   16.73   0.8807
177    20     6   76 0.0000   37.98   0.9995 0.1579   13.59   0.7152
178    20     7   29 0.0000   37.98   0.9996 0.0526   16.90   0.8892
179    20     8  182 0.0000   37.98   0.9996 0.0526   17.16   0.9029
180    20     9   95 0.0000   37.98   0.9995 0.0000   18.21   0.9585
```

For each split, the parsimonious model identified (denoted as **) is evaluated by an independent 100 splits (n.split.eval=100) generating the following output.

```
out$model.eval
```

```
     G1  G2  G3  G4 G5  G6 G7 G8 mean ER mean MiPP mean sMiPP 5% sMiPP
S1   29 177  NA  NA NA  NA NA NA  0.0111     18.25     0.9607   0.8829
S2   29 102  36  18 49  NA NA NA  0.0211     18.15     0.9554   0.8819
S3   29 177 185  NA NA  NA NA NA  0.0042     18.76     0.9873   0.9106
S4   29 177 185  NA NA  NA NA NA  0.0042     18.76     0.9873   0.9106
S5   30  36 185  NA NA  NA NA NA  0.0000     18.89     0.9942   0.9824
S6   36  30  NA  NA NA  NA NA NA  0.0016     18.72     0.9855   0.9601
S7   29 102 185 177 NA  NA NA NA  0.0058     18.67     0.9824   0.8936
S8   29  49 185  36 18 177 NA NA  0.0026     18.86     0.9927   0.9765
S9   28  65 148  NA NA  NA NA NA  0.0779     15.77     0.8299   0.6800
S10  49  95 177  29 91  65 84 36  0.0158     18.35     0.9657   0.8140
S11  29 102  65 185 95  NA NA NA  0.0163     18.33     0.9646   0.8911
S12  29 102 177 185 NA  NA NA NA  0.0058     18.67     0.9824   0.8936
S13  29 177 185  NA NA  NA NA NA  0.0042     18.76     0.9873   0.9106
S14  49  95  36  29 84 177 NA NA  0.0053     18.65     0.9814   0.9035
S15  30  36 177 185 NA  NA NA NA  0.0000     18.97     0.9984   0.9960
S16  30  36 185  NA NA  NA NA NA  0.0000     18.89     0.9942   0.9824
S17  28  36 177  91 29  NA NA NA  0.0026     18.79     0.9890   0.9735
S18 163 177 185  29 NA  NA NA NA  0.0037     18.78     0.9885   0.9108
S19  29  49 185  36 95  65 NA NA  0.0074     18.68     0.9832   0.9056
S20  30 102  36  NA NA  NA NA NA  0.0005     18.80     0.9896   0.9739
```

```
    50% sMiPP 95% sMiPP
S1     0.9759    0.9901
S2     0.9890    0.9986
S3     0.9962    0.9988
S4     0.9962    0.9988
S5     0.9964    0.9990
S6     0.9907    0.9958
S7     0.9954    0.9990
```

```
S8     0.9989    0.9999
S9     0.8523    0.9745
S10    0.9971    0.9999
S11    0.9910    0.9990
S12    0.9954    0.9990
S13    0.9962    0.9988
S14    0.9942    0.9994
S15    0.9989    0.9997
S16    0.9964    0.9990
S17    0.9962    0.9995
S18    0.9980    0.9997
S19    0.9975    0.9998
S20    0.9924    0.9972
```

3.3 Sequential selection

Good classifying genes may be masked by other better classifying genes, so the genes may be discovered if the other genes are not present. Therefore, it is worth selecting gene models after removing genes selected in the first run. The *MiPP* package also enables to sequentailly select gene models after removing genes selected in the previous runs.

For the acute leukemia data, the sequntial analysis can be performed by the following arguments in the `mipp.seq` function: The argument *n.seq=3* means that the sequentaik analysis is performed 3 times.

```
out <- mipp.seq(x=x.train, y=y.train, x.test=x.test, y.test=y.test,
            n.fold=5, percent.cut=0.05, rule="lda", n.seq=3)
```

For the colon cancer data, the sequntial analysis can be performed by the following arguments:

```
out <- mipp.seq(x=x, y=y, n.fold=5, p.test=1/3, n.split=20, n.split.eval=100,
            percent.cut = 0.1 , rule="lda", n.seq=3)
```

Reference

Soukup M, Cho H, and Lee JK (2005). Robust classification modeling on microarray data using misclassification penalized posterior, *Bioinformatics*, 21 (Suppl): i423-i430.

Soukup M and Lee JK (2004). Developing optimal prediction models for cancer classification using gene expression data, *Journal of Bioinformatics and Computational Biology*, 1(4) 681-694.