

# How to use the MiPP Package

Mat Soukup, HyungJun Cho, and Jae K. Lee

May 18, 2005

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Misclassification-Penalized Posteriors (MiPP)</b>	<b>1</b>
<b>3</b>	<b>Examples</b>	<b>2</b>
3.1	Acute Leukemia Data: . . . . .	2
3.2	Colon Cancer Data: . . . . .	4

## 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 discriminant 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

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

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<sup>TM</sup> 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)),levels=c("ALL","AML"),
                  labels=c("ALL", "AML"))

#Test set
x.test <- leukemia[,39:72]
y.test <- factor(c(rep("ALL",20),rep("AML",14)),levels=c("ALL","AML"),
                  labels=c("ALL", "AML"))
```

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,
            nfold=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 (*nfold=5*) on the training set. To reduce computing time, it pre-selects the most plausible 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	ErrorRate	MiPP	sMiPP	Select
1	1	571	0.11764706	23.91891	0.7034973	
2	2	436	0.02941176	30.41434	0.8945395	
3	3	366	0.02941176	31.35401	0.9221767	
4	4	457	0.02941176	32.14149	0.9453380	
5	5	413	0.02941176	32.17713	0.9463862	
6	6	635	0.00000000	33.75339	0.9927467	**
7	7	648	0.00000000	33.63446	0.9892489	
8	8	181	0.02941176	31.98469	0.9407261	

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<sup>+</sup> 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 <- colnames(colon)

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

out <- mipp(x=x, y=y, nfold=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 data sets 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	ErrorRate	MiPP	sMiPP	Select
1	1	1	29	0.05263158	16.032732	0.8438280	
2	1	2	177	0.00000000	18.458082	0.9714780	
3	1	3	163	0.00000000	18.832489	0.9911836	**
4	1	4	36	0.00000000	18.978443	0.9988654	*
5	1	5	51	0.00000000	18.972158	0.9985346	
6	1	6	95	0.00000000	18.969822	0.9984117	
7	2	1	29	0.10526316	14.512517	0.7638167	
8	2	2	102	0.10526316	15.420517	0.8116061	
9	2	3	36	0.05263158	16.652730	0.8764595	
10	2	4	185	0.05263158	16.929696	0.8910366	
11	2	5	76	0.00000000	18.562381	0.9769674	**
12	2	6	78	0.05263158	17.446542	0.9182391	
13	2	7	95	0.05263158	17.138486	0.9020256	
14	3	1	28	0.21052632	10.993642	0.5786127	
15	3	2	36	0.10526316	15.323195	0.8064840	
16	3	3	78	0.00000000	18.692086	0.9837940	**
17	3	4	51	0.05263158	17.047799	0.8972526	
18	3	5	29	0.00000000	18.095243	0.9523812	
.							
.							
.							
128	20	1	163	0.10526316	13.724261	0.7223295	
129	20	2	177	0.00000000	18.774879	0.9881515	**
130	20	3	185	0.00000000	18.825061	0.9907927	*
131	20	4	182	0.05263158	17.033708	0.8965109	
132	20	5	29	0.00000000	18.676012	0.9829480	

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

	Split	G1	G2	G3	G4	G5	G6	G7	mean ErrorRate	mean MiPP	mean sMiPP
S1	1	29	177	163	NA	NA	NA	NA	0.0084210526	18.57919	0.9778522
S2	2	29	102	36	185	76	NA	NA	0.0173684211	18.26665	0.9614028
S3	3	28	36	78	NA	NA	NA	NA	0.0005263158	18.74241	0.9864428
S4	4	141	185	49	91	177	36	30	0.0026315789	18.84880	0.9920420
S5	5	163	177	84	185	NA	NA	NA	0.0010526316	18.70606	0.9845295
S6	6	163	177	36	NA	NA	NA	NA	0.0000000000	18.74260	0.9864524
S7	7	30	36	78	185	NA	NA	NA	0.0000000000	18.93579	0.9966204
S8	8	51	185	49	29	36	76	NA	0.0247368421	17.96189	0.9453627
S9	9	30	36	NA	NA	NA	NA	NA	0.0015789474	18.68832	0.9835957
S10	10	29	177	NA	NA	NA	NA	NA	0.0110526316	18.28892	0.9625746
S11	11	29	102	163	36	NA	NA	NA	0.0263157895	17.86323	0.9401701
S12	12	29	177	182	NA	NA	NA	NA	0.0052631579	18.60552	0.9792380
S13	13	29	177	182	NA	NA	NA	NA	0.0052631579	18.60552	0.9792380
S14	14	30	36	NA	NA	NA	NA	NA	0.0015789474	18.68832	0.9835957
S15	15	29	177	185	NA	NA	NA	NA	0.0042105263	18.76306	0.9875297
S16	16	29	177	36	NA	NA	NA	NA	0.0063157895	18.66415	0.9823239
S17	17	163	177	NA	NA	NA	NA	NA	0.0021052632	18.51119	0.9742732
S18	18	163	177	36	NA	NA	NA	NA	0.0000000000	18.74260	0.9864524
S19	19	28	36	185	177	NA	NA	NA	0.0000000000	18.91219	0.9953783
S20	20	163	177	NA	NA	NA	NA	NA	0.0021052632	18.51119	0.9742732

	5% sMiPP	50% sMiPP	95% sMiPP
S1	0.8832269	0.9956378	0.9997555
S2	0.8904381	0.9907046	0.9979650
S3	0.9717611	0.9888683	0.9954501
S4	0.9720076	0.9982314	0.9997744
S5	0.9677334	0.9877863	0.9977993
S6	0.9696978	0.9889706	0.9973368
S7	0.9888911	0.9976407	0.9993538
S8	0.8734358	0.9763289	0.9983271
S9	0.9612196	0.9894887	0.9957796
S10	0.8723262	0.9770533	0.9935208
S11	0.8241824	0.9776791	0.9974065
S12	0.9103882	0.9888216	0.9986135
S13	0.9103882	0.9888216	0.9986135
S14	0.9612196	0.9894887	0.9957796
S15	0.9004550	0.9968640	0.9989926
S16	0.8970961	0.9937537	0.9984018
S17	0.9576879	0.9776923	0.9936058

S18 0.9696978 0.9889706 0.9973368  
S19 0.9871570 0.9970437 0.9992126  
S20 0.9576879 0.9776923 0.9936058

## Reference

Soukup M, Cho H, and Lee JK (2005). Robust classification modeling on microarray data using misclassification penalized posterior, *Bioinformatics* (forthcoming).

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.