

An Efficient Max-Dependency Algorithm for Gene Selection

Hanchuan Peng

Fuhui Long

Lawrence Berkeley National Laboratory,
Life Sciences Division,
University of California, Berkeley, CA, USA, 94720.
Email: hpeng@lbl.gov

Duke University,
Center for Cognitive Neuroscience,
Durham, NC, USA, 27708.
Email: long@neuro.duke.edu

Brief Abstract

In many bioinformatics problems such as cancer classification using microarray data, it often needs to select the most characterizing genes so that they jointly have high discriminative strength to categorize the cancer type (the target variable). Based on information theory, we prove that for first-order gene selection, the high-dimensional max-dependency criterion which maximizes the dependency between the joint states of the selected features and the target classification variable is equivalent to a combination of low-dimensional max-relevance and min-redundancy criteria to select the least redundant features. Based on this proof, we then develop an efficient gene selection algorithm for max-dependency gene selection. Our comprehensive experimental results on several public-domain data sets (NCI cancer cell lines, Lymphoma, etc.) and different classifiers (naïve Bayes, support vector machine, and linear discriminate analysis, etc.) show that this novel method is very effective in selecting a rather small set of genes from microarray data for gene expression classification. More information can be found at the web site: www.hpeng.net.

Extended Abstract

In many bioinformatics applications, identifying the most characterizing gene features of the observed data, i.e., feature selection (or variable selection, among many other names), is critical to minimize the classification error. Given the input data D tabled as N samples and M features $X = \{x_i, i=1, \dots, M\}$, and the target classification variable c , feature selection problem is to find from the M -dimensional observation space, R^M , a subspace of m features, R^m , to "optimally" characterize c .

Given a condition defining the "optimal characterization", which often means the *minimal classification error*, a good strategy is to select genes that maximize statistical dependency of the target class c on the data distribution in the subspace R^m (and vice versa). This scheme is *maximal dependency* (Max-Dependency). Dependency is usually characterized in term of mutual information. For two random variables x and y , their mutual information is defined in terms of their probabilistic density functions $p(x)$, $p(y)$ and $p(x,y)$:

$$I(x; y) = \iint p(x, y) \log \frac{p(x, y)}{p(x)p(y)} dx dy. \quad (1)$$

One of the most widely used approaches to realize Max-Dependency is *maximal relevance* (Max-Relevance) feature selection: selecting the features with the highest relevance to the target class c . However, it has been well recognized that the combinations of individually good features do not necessarily lead to good classification performance [2][3]. In other words, "the m best

features are not the best m features" [2][3]. The Max-Dependency criterion cannot be simply approximated using the Max-Relevance methods.

In terms of mutual information, the purpose of Max-Dependency feature selection is to find a feature set S with m features $\{x_i\}$, which jointly have the largest dependency on the target class c . This scheme has the following form:

$$\max D(S, c), \quad D = I(\{x_i, i = 1, \dots, m\}; c). \quad (2)$$

Obviously, when m equals 1, the solution is the feature that maximizes $I(x_j; c)$ ($1 \leq j \leq M$). When $m > 1$, a simple incremental search scheme is to add one feature at one time: given the set with $m-1$ features, S_{m-1} , the m th feature can be determined as the one that contributes to the largest increase of $I(S; c)$, which takes the form in Eq. (3).

$$\begin{aligned} I(S_m; c) &= \iint p(S_m, c) \log \frac{p(S_m, c)}{p(S_m)p(c)} dS_m dc \\ &= \iint p(S_{m-1}, x_m, c) \log \frac{p(S_{m-1}, x_m, c)}{p(S_{m-1}, x_m)p(c)} dS_{m-1} dx_m dc \\ &= \int \cdots \int p(x_1, \dots, x_m, c) \log \frac{p(x_1, \dots, x_m, c)}{p(x_1, \dots, x_m)p(c)} dx_1 \cdots dx_m dc. \end{aligned} \quad (3)$$

We note that it is often hard to get an accurate estimation for multivariate density $p(x_1, \dots, x_m)$ and $p(x_1, \dots, x_m, c)$, because of two difficulties in the high-dimensional space: 1) the number of samples is often insufficient, and 2) the multivariate density estimation often involves calculation of the inverse of the high-dimension covariance matrix, which is usually an ill-posed problem. Another drawback of Max-Dependency is the slow computational speed. These problems are most pronounced for continuous feature variables. However, they cannot be entirely avoided even for discrete variables.

We prove that:

- (1) In the case one feature is selected at one time (i.e. "first-order" selection), an optimal approximation scheme to Max-Dependency is to select the feature that has the maximal dependency on the target classification variable and the minimal dependency on each of the already selected features.
- (2) In the case one feature is removed at one time (i.e. "first-order" removal), an optimal approximation scheme to Max-Dependency is to remove the feature that has the minimal dependency on the target classification variable and the maximal dependency on each of the rest features.

In both cases, we only need to consider a bunch of 2-variable mutual information computations, instead of calculating mutual information in the higher-dimensional space as in Eq. (3). This makes the computation much easier and faster. In practice, this also leads to more robust feature-identification.

For the incremental selection case, this work gives theoretical justification of our earlier minimal-redundancy-maximal-relevance (MRMR) selection method [1].

The proofs are partially included in the attached powerpoint/pdf slides of the talk.

The results on several public microarray gene-expression datasets using different classifiers (naïve Bayes (NB), support vector machine (SVM), and linear discriminate analysis (LDA)) are also shown in the talk slides.

We also show results on how to combine this approach with various wrapper selection methods, including forward and backward selections.

More information and related papers can be found at <http://www.hpeng.net>.

References:

- [1] Ding, C., and Peng, H.C., "Minimum redundancy feature selection from microarray gene expression data," Proc. 2nd IEEE Computational Systems Bioinformatics Conf., pp.523-528, Stanford, CA, Aug, 2003. (Also in a coming issue of Journal of Bioinformatics and Computation Biology, 2005)
- [2] Duin, R.P.W., and Tax, D.M.J., Experiments with Classifier Combining Rules, in: J. Kittler, F. Roli (eds.), *Multiple Classifier Systems* (Proc. First Int Workshop, MCS 2000, Cagliari, Italy, June 2000), Lecture Notes in Computer Science, vol. 1857, Springer, Berlin, pp. 16-29, 2000.
- [3] Jain, A.K., and Zongker, D., "Feature selection: evaluation, application, and small sample performance," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 19, no. 2, pp. 153-158, Feb, 1997.

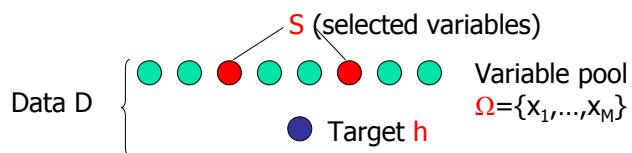
Efficient Maximum Dependency Algorithms for Feature Selection

Hanchuan Peng

Life Science/Genomics Division,
Lawrence Berkeley National Laboratory

1

Feature Selection Problem



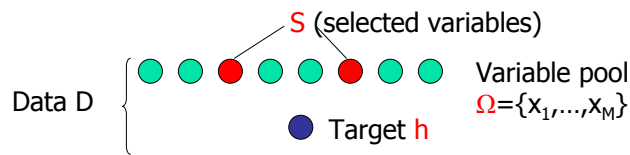
A prediction example

	Ω {Gene x_i }			Cancer type h
	G_1	G_2	G_M	
N Samples	On	On	On	Prostate
	On	Off	Off	Prostate
	Baseline	On	On	Liver
	Off	Off	On	Liver
	Baseline	Off	Off	Lung
	On	On	On	Lung
	Off	On	Off	Lymphoma
	On	On	On	Lymphoma
	On	On	On	Lymphoma

2

Maximum Dependency Criterion

- Statistical association
- Definition
 - Mutual information $I(S,h)$



3

Mutual Information

- For two univariate variables x and y

$$I(x; y) = \iint p(x, y) \log \frac{p(x, y)}{p(x)p(y)} dx dy$$

- For multivariate variable S_m and the target h

$$I(S_m; h) = \iint p(S_m, h) \log \frac{p(S_m, h)}{p(S_m)p(h)} dS_m dh$$

4

High-Dimensional Mutual Information

- For multivariate variable S_m and the target h

$$\begin{aligned}
 I(S_m; h) &= \iint p(S_m, h) \log \frac{p(S_m, h)}{p(S_m)p(h)} dS_m dh \\
 &= \iint p(S_{m-1}, x_m, h) \log \frac{p(S_{m-1}, x_m, h)}{p(S_{m-1}, x_m)p(h)} dS_m dh \\
 &= \int \cdots \int p(x_1, \dots, x_m, h) \log \frac{p(x_1, \dots, x_m, h)}{p(x_1, \dots, x_m)p(h)} dx_1 \cdots dx_m dh.
 \end{aligned}$$

- Estimating high-dimensional $I(S_m, h)$ is difficult
 - An ill-posed problem to find inverse of large co-variance matrix
 - Insufficient number of samples

5

Maximum Dependency Feature Selection is Combinatorial !

- Number of subspaces

- Total $2^{|\Omega|}$ or 2^M
- For given number of features $|S|$ $\binom{|\Omega|}{|S|}$

Example:

$ \Omega $	$ S $	#configurations of selected variables
1000	1	10^3
1000	2	0.5×10^6
1000	3	$\sim 1.66 \times 10^8$
5000	3	$\sim 2.08 \times 10^{10}$

Heuristic search algorithms are necessary.
Simplest case: the incremental search.

6

Factorize the Mutual Information

Mutual information for multivariate variable S_m
and the target h

$$I(S_m; h) = \iint p(S_m, h) \log \frac{p(S_m, h)}{p(S_m)p(h)} dS_m dh$$

Define:

$$J(x_1, x_2, \dots, x_m) = \int \dots \int p(x_1, \dots, x_m) \log \frac{p(x_1, x_2, \dots, x_m)}{p(x_1) \dots p(x_m)} dx_1 \dots dx_m$$

It can be proved:

$$I(S_m, h) = J(h, S_m) - J(S_m)$$

7

Upper & Lower Bounds of J(.)

Lower bound

$$J(x_1, x_2, \dots, x_m) \geq 0.$$

when variables are maximally **independent**

Upper bound

$$J(x_1, x_2, \dots, x_n) \leq \min \left\{ \sum_{i=2}^n H(x_i), \sum_{i=1, i \neq 2}^n H(x_i), \dots, \sum_{i=1, i \neq n-1}^n H(x_i), \sum_{i=1}^{n-1} H(x_i) \right\}$$

when variables are maximally **dependent**

8

Factorize $I(S_m, h)$

- **Relevance** of $S = \{x_1, x_2, \dots\}$ and h , or $R_L(S, h)$
- **Redundancy** among variables $\{x_1, x_2, \dots\}$, or $R_D(S)$

$$R_L = \frac{1}{|S|} \sum_{x_i \in S} I(x_i, h) \quad R_D = \frac{1}{|S|^2} \sum_{x_i, x_j \in S} I(x_i, x_j)$$
$$I(S_m, h) = J(S_{m-1}, x_m, h) - J(S_{m-1}, x_m).$$

- For incremental search, $\max I(S, h)$ is equivalent to $\max [R_L(S, h) - R_D(S)]$, i.e. combination of **min-Redundancy-Max-Relevance (mRMR)**.

9

Advantages of mRMR

- Both relevance and redundancy estimation are low-dimensional problems (i.e. involving only 2 variables). This is much easier than directly estimating multivariate density or mutual information in the high-dimensional space!
- Faster speed
- More reliable estimation

10

Search Algorithm

■ Greedy search algorithm

- In the pool Ω find the variable x^1 that has the largest $I(.,h)$. Exclude x^1 from Ω .
- Search x^2 so that it **maximizes** $I(.,h) - \Sigma I(.,x^1)/|\Omega|$.
- Iterate this process until an expected number of variables have been obtained, or other constraints are satisfied.

■ Complexity $O(|S|^*|\Omega|)$

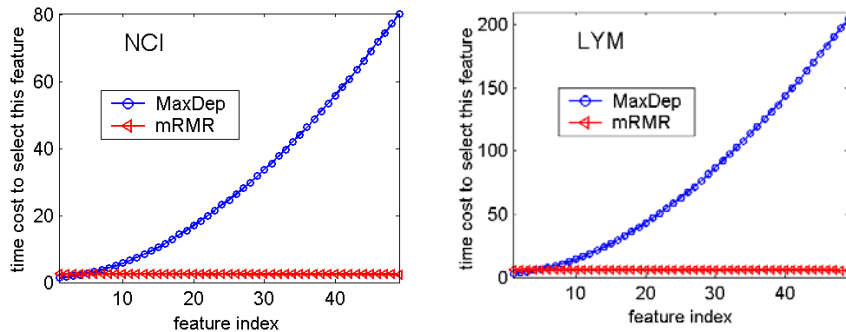
11

Data sets

DATASET	NCI		LYMPHOMA	
SOURCE	Ross et al (2000) Scherf et al (2000)		Alizadeh et al (2000)	
# GENE	9703		4026	
# S	60		96	
# CLASS	9		9	
CLASS	CLASS NAME	# S	CLASS NAME	# S
C1	NSCLC	9	Diffuse large B cell lymphoma	46
C2	Renal	9	Chronic Lympho. leukemia	11
C3	Breast	8	Activated blood B	10
C4	Melanoma	8	Follicular lymphoma	9
C5	Colon	7	Resting/ activated T	6
C6	Leukemia	6	Transformed cell lines	6
C7	Ovarian	6	Resting blood B	4
C8	CNS	5	Germinal center B	2
C9	Prostate	2	Lymph node/tonsil	2

12

Comparing Max-Dep and mRMR: Complexity of Feature Selection

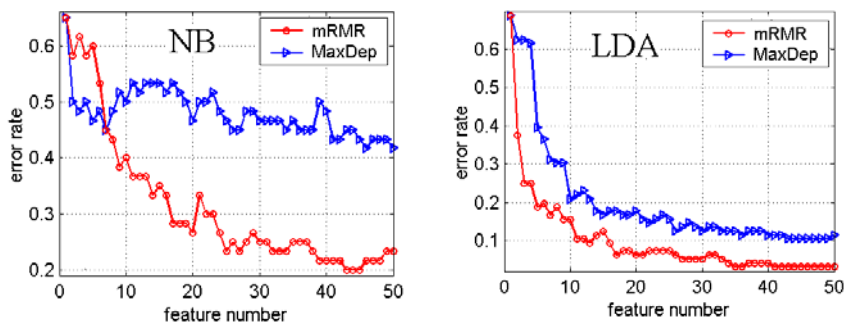


Time cost (seconds) for selecting individual features based on mutual information estimation for continuous feature variables.

(Parallel experiments on a cluster of eight 3.06G Xeon CPUs running Redhat Linux 9, with the Matlab implementation)

13

Comparing Max-Dep and mRMR: Accuracy of Feature Selected in Classification



Leave-One-Out cross validation of feature classification accuracies of mRMR and MaxDep

14

Comparing mRMR with the Standard Max-Relevance Method

- **Standard methods:**

- Select top-ranking variables based on mutual information, t-test, F-test, etc.

- **What is different in mRMR, which also considers redundancy in features :**

- Selected variables are less correlated
 - Selected features cover broader regions in the feature space
 - An optimal approximation method for Max-Dependency

15

NCI Cell-line Data (60 samples, 9-class)

Continuous Data (error rate)

Classifier	Method	<i>m</i>											
		1	5	10	15	20	25	30	35	40	45	50	
NB	MaxRel	65.00	51.67	51.67	45.00	46.67	43.33	41.67	38.33	36.67	33.33	36.67	
	mRMR	65.00	60.00	40.00	35.00	26.67	23.33	25.00	25.00	21.67	20.00	23.33	
SVM	MaxRel	98.33	46.67	55.00	50.00	45.00	55.00	41.67	35.00	38.33	35.00	36.67	
	mRMR	98.33	70.00	58.33	48.33	40.00	31.67	31.67	31.67	26.67	23.33	23.33	
LDA	MaxRel	73.33	60.00	60.00	50.00	46.67	46.67	41.67	36.67	38.33	41.67	40.00	
	mRMR	73.33	66.67	50.00	53.33	45.00	33.33	35.00	35.00	33.33	30.00	30.00	

Discretized Data (error number)

Classifier	Method	<i>M</i>														
		3	6	9	12	15	18	21	24	27	30	36	42	48	54	60
NB	Baseline	29	26	20	17	14	15	12	11	11	13	13	14	14	15	13
	MID	28	15	13	13	6	7	8	7	7	5	8	9	9	8	10
LDA	Baseline	35	25	23	20	21	18	19	19	16	19	17	19	17	16	17
	MID	31	20	21	19	16	16	16	16	14	14	10	9	9	8	8
SVM	Baseline	34	29	27	25	21	19	19	19	20	18	17	18	18	18	16
	MID	33	20	19	20	18	17	17	16	13	13	9	8	7	7	8

16

Lymphoma Data (96 samples, 9-class)

Continuous Data (error rate)

Classifier	Method	m										
		1	5	10	15	20	25	30	35	40	45	50
NB	MaxRel	72.92	25.00	15.63	13.54	13.54	12.50	13.54	12.50	11.46	11.46	10.42
	MRMR	72.92	17.71	16.67	10.42	11.46	9.38	10.42	9.38	9.38	7.29	8.33
SVM	MaxRel	42.71	27.08	21.88	21.88	18.75	16.67	14.58	14.58	15.63	11.46	12.50
	MRMR	42.71	11.46	10.42	7.29	5.21	7.29	7.29	5.21	5.21	5.21	4.17
LDA	MaxRel	68.75	32.29	22.92	23.96	23.96	21.88	22.92	17.71	16.67	16.67	15.63
	MRMR	68.75	18.75	15.63	12.50	6.25	7.29	5.21	3.13	4.17	3.13	3.13

Discretized Data (error number)

Classifier	Method	M													
		3	6	9	12	15	18	21	24	27	30	36	42	48	54
NB	MaxRel	38	39	25	29	23	22	22	19	20	17	19	18	17	17
	mRMR	31	15	10	9	9	8	6	7	7	7	4	7	5	8
LDA	MaxRel	40	42	28	26	20	21	21	20	18	19	14	15	13	14
	mRMR	32	15	14	10	7	5	4	5	4	6	5	3	3	4
SVM	MaxRel	32	29	25	23	20	22	18	13	14	15	11	10	10	8
	mRMR	24	10	7	4	2	3	3	3	3	3	3	3	3	3

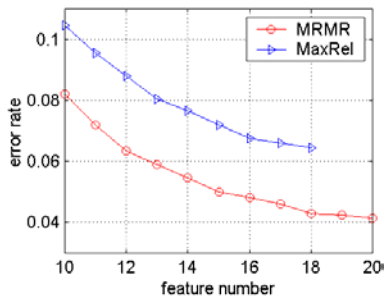
17

Use Wrappers to Refine Features

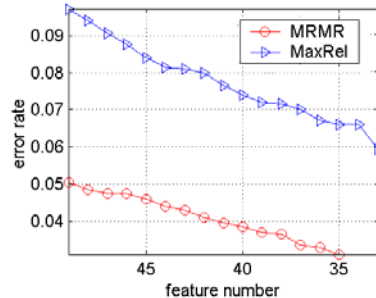
- mRMR is a filter approach
 - Fast
 - Features might be redundant
 - Independent of the classifier
- Wrappers seek to minimize the number of errors directly
 - Slow
 - Features are less robust
 - Dependent on classifier
 - Better prediction accuracy
- Use mRMR first to generate a short feature pool and use wrappers to get a least redundant feature set with better accuracy

18

Use Wrappers to Refine Features



Forward wrappers
(incremental selection)



Backward wrappers
(Decremental selection)

NCI data

19

Conclusions

- The Max-Dependency feature selection can be efficiently implemented as the mRMR algorithm.
- Significantly outperforms the currently widely used max-relevance selection method: mRMR features cover a broader feature space with less features.
- mRMR is very efficient and useful in gene selection.

More information: <http://hpeng.net>

20

Upper & Lower Bounds of J(.)

$$\begin{aligned}
 -J(x_1, x_2, \dots, x_m) &= \int \cdots \int p(x_1, \dots, x_m) \log \frac{p(x_1) \cdots p(x_m)}{p(x_1, \dots, x_m)} dx_1 \cdots dx_m \\
 &\leq \int \cdots \int p(x_1, \dots, x_m) \left[\frac{p(x_1) \cdots p(x_m)}{p(x_1, \dots, x_m)} - 1 \right] dx_1 \cdots dx_m \\
 &= \int \cdots \int p(x_1) \cdots p(x_m) dx_1 \cdots dx_m - \int \cdots \int p(x_1, \dots, x_m) dx_1 \cdots dx_m \\
 &= 1 - 1 = 0.
 \end{aligned}$$

$$\begin{aligned}
 J(y_1, y_2, \dots, y_n) &= \int \cdots \int p(y_1, \dots, y_n) \log \frac{p(y_1, \dots, y_n)}{p(y_1) \cdots p(y_n)} dy_1 \cdots dy_n \\
 &= \int \cdots \int p(y_1, \dots, y_n) \log \frac{p(y_1 | y_2, \dots, y_n) p(y_2 | y_3, \dots, y_n) \cdots p(y_{n-1} | y_n) p(y_n)}{p(y_1) \cdots p(y_{n-1}) p(y_n)} dy_1 \cdots dy_n \\
 &= \sum_{i=1}^{n-1} H(y_i) - H(y_1 | y_2, \dots, y_n) - H(y_2 | y_3, \dots, y_n) - \cdots - H(y_{n-1} | y_n) \\
 &\leq \sum_{i=1}^{n-1} H(y_i).
 \end{aligned}$$

21

Jensen Inequality

Definition 6 A function $f(x)$ is said to be **convex** over an interval (a, b) if for every $x_1, x_2 \in (a, b)$ and $0 \leq \lambda \leq 1$,

$$f(\lambda x_1 + (1 - \lambda)x_2) \leq \lambda f(x_1) + (1 - \lambda)f(x_2)$$

A function is **strictly convex** if equality holds only if $\lambda = 0$ or $\lambda = 1$. \square

Theorem 6 If f is a convex function and X is a r.v. then

$$Ef(X) \geq f(EX).$$

Put another way,

$$\sum_x p(x) f(x) \geq f\left(\sum_x p(x) x\right)$$

If f is strictly convex then equality in the theorem implies that $X = EX$ w.p. 1.

If f is concave then

$$Ef(X) \leq f(EX).$$