SVM-Based Local Search for Gene Selection and Classification of Microarray Data

Jose Crispin Hernandez Hernandez, Béatrice Duval, and Jin-Kao Hao

LERIA, Université d'Angers,
2 Boulevard Lavoisier, 49045 Angers, France
{josehh,bd,hao}@info.univ-angers.fr

Abstract. This paper presents a SVM-based local search (SVM-LS) approach to the problem of gene selection and classification of microarray data. The proposed approach is highlighted by the use of a SVM classifier both as an essential part of the evaluation function and as a "provider" of useful information for designing effective LS algorithms. The SVM-LS approach is assessed on a set of three well-known data sets and compared with some best algorithms from the literature.

Keywords: Microarray gene expression, Feature selection, Local search, Support vector machines.

1 Introduction

With the fast advances of DNA Microarray technologies, more and more gene expression data are made available for analysis. These data can be used for various purposes, for instance, in classification of tissue samples using gene discriminator between normal and cancer samples [6,2].

Gene expression data are known to be of very high dimensions (thousands of gene expressions at least) with a small number of samples (typically under one hundred). This characteristic, known as the "curse of dimensionality", induces a difficulty for classification and requires special techniques to reduce the data dimensionality (gene selection) in order to obtain reliable predictive results.

Gene selection is a kind of feature selection [10], aiming at identifying a (small) subset of informative genes from the initial data in order to obtain high classification accuracy. In the literature there are two main approaches for feature selection: the filter approach and the wrapper approach.

In the filter approach [5], feature selection is performed without taking into account the classification algorithm that will be applied to the selected features. A filter algorithm generally relies on a relevance measure that evaluates the importance of each feature for the classification task. A typical filter algorithm ranks all the features according to their interestingness for the classification problem and selects the top ranked features. The feature score can be obtained independently for each feature, as it is done in [6] which relies on correlation coefficients between the class and each feature. The drawback of such a method

M. Elloumi et al. (Eds.): BIRD 2008, CCIS 13, pp. 499–508, 2008.

is to score each feature independently and to ignore the relations between the features.

In contrast, the wrapper approach selects a subset of features that is "optimized" for a given classification algorithm. So the classification algorithm, that is considered as a black box, is run many times on different candidate subsets, and each time, the quality of the candidate subset is evaluated by the performance of the classification algorithm trained on this subset. The wrapper approach conducts a search in the space of candidate subsets. For this search problem, genetic algorithms have been used in a number of studies, see e.g. [12,11,8]. Embedded methods, a variant of the wrapper approach, use feature selection as a part of the training process in which the learning algorithm is no more a simple black box. One example of an embedded method is proposed in [7] with recursive feature elimination using support vector machines (SVM-RFE).

In this paper, we present a Local Search approach guided by SVM which can be considered as an embedded method. In this approach, a SVM classifier is used not only to evaluate a candidate gene subset, but also to provide the local search algorithm with useful information for its search operators. As we show in the experimentation section, despite its simplicity, this SVM-based Local Search (SVM-LS) approach allows us to obtain highly competitive results on three well-known data sets when compared with some best algorithms from the literature.

2 SVM Classification and Gene Selection

It is common in wrapper approaches to use a classifier to evaluate the quality of a proposed gene subset. SVM classifiers can be used for such a purpose. In our SVM-based Local Search approach, a SVM classifier is used not only in the evaluation function of gene subsets but also in the design of LS strategies. SVM is thus a key component of our SVM-LS approach. For this reason, this section recalls the main characteristics of SVM and explains how a feature selection process can be guided by useful information provided by a SVM classifier.

2.1 Support Vector Machines

SVMs represent a class of state-of-the-art classifiers [4] that have been successfully used for gene selection and classification [7,13]. SVMs solve a binary classification problem by searching a decision boundary that has the maximum margin with the examples. SVMs handle complex decision boundaries by using linear machines in a high dimensional feature space, implicitly represented by a kernel function. In this work, we only consider linear SVMs because they are known to be well suited to the datasets that we consider.

For a given training set of labeled samples, a linear SVM determines an optimal hyperplane that divides the positively and the negatively labeled samples with the maximum margin of separation. A noteworthy property of SVM is that the hyperplane only depends on a small number of training examples called the support vectors, they are the closest training examples to the decision boundary and they determine the margin.

Formally, we consider a training set of n samples belonging to two classes; each sample is noted $\{X_i, y_i\}$ where $\{X_i\}$ is the vector of attribute values describing the sample and y_i the class label.

A soft-margin linear SVM classifier aims at solving the following optimization problem:

$$\min_{w,b,\xi_i} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{n} \xi_i \tag{1}$$

subject to $y_i (w \cdot X_i + b) \geq 1 - \xi_i$ and $\xi_i \geq 0$, $i = 1, ..., n$.

In this formulation, w is the weight vector that determines the separating hyperplane; C is a given penalty term that controls the cost of misclassification errors. To solve this optimization problem, it is convenient to consider the dual formulation [4]:

$$\min_{\alpha_i} \frac{1}{2} \sum_{i=1}^{n} \sum_{l=1}^{n} \alpha_i \alpha_l y_i y_l X_i \cdot X_l - \sum_{i=1}^{n} \alpha_i \tag{2}$$

$$st\text{: } \textstyle\sum_{i=1}^{n} y_i \alpha_i = 0 \text{ and } 0 \leq \alpha_i \leq C$$

The decision function for the linear SVM classifier with input vector X is given by: $\varphi(X) = w \cdot X + b$ with $w = \sum_{i=1}^{n} \alpha_i y_i X_i$ and $b = y_i - w \cdot X_i$.

The weight vector w is a linear combination of training samples. Most weights α_i are zero and the training samples with non-zero weights are the support vectors. The maximum margin M is given by:

$$M = \frac{2}{\|w\|} \tag{3}$$

2.2 Gene Ranking by SVM

As discussed in [7], the weights of a linear discriminant classifier can be used to rank the genes for selection purposes. More precisely, in a backward selection method, one starts with all the genes and removes iteratively the least informative gene. To determine the feature to be removed at each iteration, one considers the gene that has the least influence on the cost function of the classification process. For a linear SVM, the cost function is defined by $\frac{1}{2}||w||^2$. So given a SVM classifier with weight vector w, one can define the ranking coefficient vector c given by:

$$\forall i, c_i = (w_i)^2 \tag{4}$$

Intuitively, in order to select informative genes, the orientation of the separating hyperplane found by a linear SVM can be used. If the plane is orthogonal to a particular gene dimension, then that gene is informative, and vice versa. As we show in the next section, the coefficient vector c contains very useful ranking information that can be used to design a dedicated LS strategy.

3 SVM-LS for Gene Selection and Classification

In this section, we present our SVM-based LS approach for gene selection and classification of Microarray data. We explain the basic ingredients and their underlying rational. Our method begins by a pre-selection step where we use a filter criterion (in our case, the BW ratio introduced in [5]) to obtain a group G_p of p (typically $p \geq 75$) top ranked genes. Then our SVM-LS approach is applied to select, from G_p a gene subset of smaller size (typically less than 20 genes).

3.1 Representation and Search Space

A candidate solution $s = < s^g, s^c >$ is composed of two parts s^g and s^c called respectively *gene subset vector* and *ranking coefficient vector* [8]. The first part, $s^g = (g_1, g_2...g_p)$, is a binary vector of fixed length p. Each $g_i \in \{0, 1\}$ ($i = 1...p$) corresponds to a particular gene and indicates whether or not the gene is selected. The second part, $s^c = (c_1, c_2...c_p)$, is a *positive* real vector of fixed length p and corresponds to the ranking coefficient vector c (Equation 4, Section 2.2) of the linear SVM classifier. s^c indicates thus for each selected gene the interestingness of this gene for the SVM classifier.

Therefore, a solution represents a candidate subset of genes with additional ranking information on each selected gene. The gene subset vector of a solution is evaluated by a linear SVM classifier and the ranking coefficients obtained during this evaluation will be used in our specialized LS strategies.

For the group G_p of p pre-selected genes, the search space is given by the set $\Omega = 2^p$ (i.e. all the possible gene subsets of p genes).

3.2 Evaluation Function

Given a candidate solution $s = < s^g, s^c >$, the quality of s (more precisely, of the gene subset part s^g) is assessed by an evaluation function f according to two criteria: the ability of s to obtain a good classification with this gene subset (C) and the maximum margin (M) given by the SVM classifier (Equation 3). More formally, the evaluation function can be written as follows:

$$f(s) =< f_C(s), f_M(s) > \tag{5}$$

where

- $f_C(s)$ is the classification accuracy of the SVM classifier using the set of genes and applied to the given training data,
- $f_M(s)$ is simply the maximum margin of the SVM classifier, given by Equation 3 (Section 2.1).

Now given two candidate solutions s and s', it is possible to compare them: $f(s)$ is better than $f(s')$, denoted by $f(s) > f(s')$, if the following condition is satisfied: $f(s) > f(s') \Leftrightarrow f_C(s) > f_C(s')$ or $f_C(s) = f_C(s') \wedge f_M(s) > f_K(s')$.

So the dominating criterion is the classification accuracy, ties are broken by comparing the maximum margins, with a preference for a larger value (a larger margin indicates a better discrimination between the two classes).

3.3 Move and Neighborhood

One of the most important features of a local search algorithm is its neighborhood. In a local search algorithm, applying a move operator mv to a candidate solution s leads to a new solution s', denoted by $s' = s \oplus mv$. Let $\Gamma(s)$ be the set of all possible moves which can be applied to s, then the neighborhood $N(s)$ of s is defined by: $N(s) = \{s \oplus mv | mv \in \Gamma(s)\}$.

In our case, the move is based on the drop/add operation which removes a gene g_i from the solution s and add another gene g_j. Moreover, the move operator is defined in such a way that it integrates semantic knowledges of the gene selection and classification problem. More formally, let $s =< s^g, s^c >$ with $s^g = (g_1, g_2 ... g_p)$ and $s^c = (c_1, c_2 ... c_p)$, define:

- $i = ArgMin_j\{c_j | c_j \in s^c \wedge c_j \neq 0\}$, i.e. i identifies the gene g_i which has the smallest ranking coefficient c_i and thus is the least relevant gene,
- $O = \{j | g_j \in s^g \wedge g_j = 0\}$, i.e. O is the set of non selected genes in the current solution s

Then our move operator drops, from the current solution, g_i (identified by the above index i) which is the least informative gene among the selected genes and adds a non selected gene g_j ($j \in O$). This can be formally written as: $mv(i, j) = (g_i : 1 \rightarrow 0; g_j : 0 \rightarrow 1)$.

Clearly, for two neighbor solutions $s =< s^g, s^c >$ and $s' =< s'^g, s'^c >$, the hamming distance between s^g and s'^g is exactly two. Moreover, one sees that the size of this neighborhood is equal to $|O|$ and bounded by p, the length of s.

3.4 Local Search Algorithms

Local search (LS) is a class of general and powerful heuristics methods [9]. For our SVM-LS approach, we implemented three LS algorithms: steepest descent (SD), Tabu Search (TS) and Iterative Local Search (ILS).

Steepest Descent (SD)**:** Given the current solution s, the steepest descent moves at each iteration to the *best improving* neighboring solution $s' \in N(s)$ such that $f(s') > f(s)$ and $\forall s" \in N(s), f(s") \leq f(s')$. Notice that SD needs no parameter and stops when no improving neighbor can be found in the neighborhood, at which point the last solution is the best solution found and corresponds to a local optimum.

Tabu Search (TS)**:** From the steepest descent SD, one can obtain a basic TS algorithm by adding a tabu list (see below). At each iteration, the current solution s is replaced by the best neighboring solution s' that is not forbidden by tabu list, i.e. $s' \in N(s)$ such that $\forall s" \in N(s), f(s") \leq f(s')$ and $s' \notin \bar{S}$ where $\bar{S}$ is the set of solutions currently forbidden by tabu list. Notice that contrary to the SD algorithm, the selected neighbor s' may or may not be better than s. The TS algorithm stops when a fixed maximum number of iterations is reached or when all the moves become tabu.

The main role of a tabu list is to prevent the search from cycling. In our case, the tabu list is implemented as follows. Each time a move $mv(i, j)$ is carried out, i.e. gene g_i is dropped and gene g_j is selected, g_i is recorded in the tabu list for the next k iterations. Consequently, g_i cannot be reselected during this period. The value of k is determined experimentally and varies typically from k_{min} to k_{max}. Notice that such a tabu list does not forbid a newly selected gene g_j to be removed soon after its selection if its ranking coefficient is very weak.

Iterate Local Search (ILS)**:** ILS uses a local search strategy (e.g. Descent or TS) to reach a local optimum s^*, at which point the search applies a perturbation operator to the local optimum solution to allow further search progress. ILS can be combined with any local search algorithm. Here, we consider the combination with TS, denoted by ILS^{TS} because this is the best combination we have found. More precisely, ILS^{TS} iterates two phases: a TS phase to reach a local optimum s^* and a perturbation to diversify the search. Our perturbation operator changes the best local optimum s^* in a controlled way and is based on the evaluation function; the second to the fifth best neighbors are successively tried in order to continue the search process. Otherwise the search stops.

3.5 Initial Solution

The initial candidate solution can be randomly created with a risk of being of bad quality. For this reason, we devise a simple way to obtain a "not-too-bad" initial solution as follows. We generate randomly l solutions such that the number of genes in each solution varies between $p * 0.9$ and $p * 0.6$ (p being the number of pre-selected genes by a filter, see the beginning of Section 3), from which the best solution according to the evaluation function (see Equation 5) is taken.

3.6 The General SVM-LS Procedure

The general SVM-LS procedure is shown in Algorithm 1. It is composed of two repeated main phases: SVM-LS phase for gene selection (Line 7) and gene reduction phase (Line 8). At line 7, a SVM-LS algorithm (with any of the above LS algorithms) is used to search for the best gene subset of a given size. After each LS phase, gene reduction is achieved by deleting the least relevant gene (i.e., the gene with the least ranking coefficient) from the best gene subset given by the SVM-LS phase, from which point a new SVM-LS search is re-applied. This two-stage process stops when removing the least interesting gene worsens the classification accuracy on the training data.

4 Experimental Results

In this section we present two comparative studies. The first compares the different LS algorithms presented in Section 3: SD, TS, ITS^{TS}. In the second study, we compare the results of our SVM-LS approach with SVM-RFE as well as three other state-of-the-art algorithms from the literature.

Algorithm 1. General SVM-LS Procedure

1: **Input**: G_p, i.e. a group of p pre-selected genes with a filter
2: **Output**: s^g, the set of selected (most informative) genes
3: Generate an initial set of genes s^g (section 3.5)
4: **repeat**
5: Evaluate s^g using the SVM classifier on the training data (section 2) and fill s^c
6: $s = (s^g, s^c)$ /* s is the current solution */
7: s= SVM-LS(s) /* LS phase: apply SVM-based local search to improve current solution $s = (s^g, s^c)$ */
8: $s^g = s^g - \{g_i\}$ /* Gene reduction phase: remove the least informative gene from the best solution found by SVM-LS phase */
9: **until** (stop condition is verified)

4.1 Data Sets

We applied our approach on three well-known datasets that concern colon cancer, leukemia and lymphoma. These data sets have largely been used for benchmarking feature selection algorithms, for instance in [14,13,11].

The colon cancer data set, first studied in [2], contains 62 tissue samples (22 normal and 40 anomal), each with 2000 gene expression values. The data set is available at http://www.molbio.princeton.edu/colondata

The leukemia data set, first studied in [6], consists of 72 tissue samples, each with 7129 gene expression values. The samples include 47 acute lymphoblastic leukemia (ALL) and 25 acute myeloid leukemia (AML). The original data are divided into a training set of 38 samples and a test set of 34 samples. The data set is available at http://www-genome.wi.mit.edu/cancer/

The lymphoma data set, first analyzed in [1], is based on 4026 variables describing 96 observations (62 and 34 of which are respectively considered as abnormal and normal). The data set is available at http://www.kyb.tuebingen.mpg.de/-bs/people/weston/l0

Notice that prior to running our method, we apply a linear normalization procedure to each data set to transform the gene expressions to mean value 0 and standard deviation 1.

4.2 Protocol for Experimentations and Comparison Criteria

To avoid the problem of selection bias which leads to over-optimistic estimations, we adopt the experimental protocol suggested in [3]. For each SVM-LS algorithm and each data set, 50 independent experiments are carried out. For each of these experiments, the data set samples are first randomly partitioned into a training set L and a testing set T ((L, T) respectively fixed at (50,12), (38,34) and (60,36) for "Colon", "Leukemia" and "Lymphoma"). The training set L is then used by the SVM-LS algorithm to determine the best gene subset G (smallest size and highest classification accuracy on the samples of L). Finally, the selected gene subset G is evaluated on the testing samples of T using the SVM classifier. The

resulting classification accuracy and the size of G are used for calculating the averaged statistics.

For comparison, we use two criteria: *averaged classification accuracy (Acc)* on the testing samples and the *averaged number of selected genes (NG)* over these 50 independent experiments. Computing time is not reported, but let us mention that one experiment on one data set takes about 20 minutes on a typical PC (Pentium Centrino Duo, 1.2MB).

4.3 Results and Comparisons

Comparison of the Three LS Algorithms. Table 1 shows the results of our SVM-LS approach using the three different LS algorithms: SD, TS and ILS^{TS}. One can rank these LS algorithms as follows: ILS^{TS} > TS > SD (> means "better than"). Indeed, ILS^{TS} performs globally the best even if for Leukemia, SD obtains a better prediction accuracy (92.52% against 91.94%), but requires more genes (6.04 against 3.14). The results of TS are also globally good, followed by the simple descent. Comparing these results with those showed in the next two tables will allow us to better assess the interest of the SVM-LS approach.

Table 1. Comparison of SVM-LS algorithms based on the classification accuracy on test set (Acc) with standard deviation and the number of selected genes (NG) with standard deviation

	SD		TS		ILS^{TS}	
Dataset	*Acc*	*NG*	*Acc*	*NG*	*Acc*	*NG*
Colon	84.84%±9.17%	15.32±1.83	85.50%±8.21%	11.16±2.81	**87.00%**±7.36%	**08.20**±2.09
Leukemia	**92.52%**±3.42%	06.04±1.38	92.47%±3.36%	04.74±1.32	91.94%±4.06%	**3.14**±1.08
Lymphome	92.11%±2.20%	17.04±2.44	92.44%±1.86%	14.32±2.21	**95.44%**±2.15%	**12.46**±1.58

Table 2. Results of SVM-RFE algorithm

	Colon		Leukemia		Lymphoma	
	Acc	*NG*	*Acc*	*NG*	*Acc*	*NG*
$SVM-RFE$	85.16%±8.11%	18.32±6.07	92.35%±3.25%	4.82±2.39	92.33%±3.96%	16.40±2.51

Comparison with SVM-RFE. The proposed approach is somewhat related to the well-known SVM-RFE approach [7]. With SVM-RFE, one starts with all features and remove iteratively the "least relevant" feature (according to the SVM classifier). Notice that SVM-RFE is fully greedy; a wrongly eliminated gene can never be reselected afterwards. Table 2 shows the results of SVM-RFE obtained under the same experimental conditions. Comparing Tables 2 and 1, one observes that SVM-RFE performs better than the pure decent algorithm, but is outperformed by TS and ILS^{TS}. This confirms the interest of using LS to explore the search space of a fixed size before gene elimination.

Comparison with State-of-the-art Approaches. Table 3 shows the results of three other best performing selection algorithms [14,13,11]. We have chosen these references because they use the same or similar experimental protocol to

Table 3. Comparison with three other SVM-based based selection methods (the symbol - indicates that the paper gives no information for the concerned dataset)

	[14]		[13]		[11]	
Dataset	*Acc*	*NG*	*Acc*	*NG*	*Acc*	*NG*
Colon	85.83%±2.0%	20	82.33%±9%	20	81.00%±8.00%	4.44±1.74
Leukemia	-	-	-	-	90.00%±6.00%	3.16±1.00
Lymphome	91.57%±0.9%	20	92.28%±4%	20	93.00%±4.00%	4.42±2.46

avoid selection bias. Once again, one observes that the SVM-LS approach (in particular with TS and ILS^{TS}) is very competitive since its results often dominate these reference methods with a higher classification accuracy and smaller set of selected genes.

5 Conclusion

In this paper, we have presented a SVM-based Local Search approach for gene subset selection and classification with two distinguished and original features. First, the evaluation function of our LS algorithms is based not only on the classification accuracy given by the SVM classifier, but also on the its maximum margin. Second, the ranking information provided by the SVM classifier is explicitly exploited in the LS strategies. These two features ensure that the SVM-LS approach is fully dedicated to the targeted problem and constitute its basic foundation.

Using an experimental protocol that avoids the selection bias problem, the SVM-LS approach is experimentally assessed on three well-known data sets (Colon, Leukemia and Lymphoma) and compared with four state-of-the-art gene selection algorithms. The experimental results clearly show that the proposed approach competes very well with the reference methods in terms of the classification accuracy and the number of selected genes. The proposed approach has an additional and important advantage over the filter methods and SVM-RFE. Indeed, SVM-LS allows us to generate multiple gene subsets of high quality, which can be used for further analysis and data mining purpose.

This study shows that local search constitutes a simple, yet powerful approach for gene selection and classification of microarray data. Its effectiveness depends strongly on how semantic information of the given problem is integrated in its basic operators such as neighborhood and evaluation function. Finally, it is clear that the proposed approach can easily be combined with other ranking and classification methods.

Acknowledgments. We acknowledge that the work is partially supported by the French Ouest Genopole® and the Region "Pays de La Loire" via the BIL (Bioinformatique Ligérienne) Project. The first author of the paper is supported by a Mexican PROMEP scholarship. We thank the reviewers of the paper for their useful comments.

References

1. Alizadeh, A., Eisen, M.B., Davis, E., et al.: Distinct types of diffuse large B–cell lymphoma identified by gene expression profiling. Nature 403, 503–511 (2000)
2. Alon, U., Barkai, N., Notterman, D.A., Gish, K., Ybarra, S., Mack, D., Levine, A.J.: Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. Proc. Natl. Acad. Sci. USA 96, 6745–6750 (1999)
3. Ambroise, C., McLachlan, G.J.: Selection bias in gene extraction on the basis of microarray gene-expression data. Proc. Natl. Acad. Sci. USA 99(10), 6562–6566 (2002)
4. Boser, B.E., Guyon, I., Vapnik, V.: A training algorithm for optimal margin classifiers. In: Proceedings of the Fifth Annual Workshop on Computational Learning Theory, pp. 144–152. ACM Press, New York (1992)
5. Dudoit, S., Fridlyand, J., Speed, T.P.: Comparison of discrimination methods for the classification of tumors using gene expression data. Journal of the American Statistical Association 97(457), 77–87 (2002)
6. Golub, T.R., Slonim, D.K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J.P., Coller, H., Loh, M.L., Downing, J.R., Caligiuri, M.A., Bloomfield, C.D., Lander, E.S.: Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. Science 286, 531–537 (1999)
7. Guyon, I., Weston, J., Barnhill, S., Vapnik, V.: Gene selection for cancer classification using support vector machines. Machine Learning 46(1-3), 389–422 (2002)
8. Hernandez Hernandez, J.C., Duval, B., Hao, J.K.: A genetic embedded approach for selection and SVM classification of microarray data. In: Marchiori, E., Moore, J.H., Rajapakse, J.C. (eds.) EvoBIO 2007. LNCS, vol. 4447, pp. 90–101. Springer, Heidelberg (2007)
9. Hoos, H., Stutzle, T.: Stochastic Local Search: Foundations and Applications. Morgan Kaufmann Publishers Inc., San Francisco (2004)
10. Kohavi, R., John, G.H.: Wrappers for feature subset selection. Artificial Intelligence 97(1-2), 273–324 (1997)
11. Paul, T.K., Iba, H.: Selection of the most useful subset of genes for gene expression-based classification. In: Proceedings of the 2004 Congress on Evolutionary Computation, pp. 2076–2083. IEEE Press, Los Alamitos (2004)
12. Peng, S., Xu, Q., Ling, X.B., Peng, X., Du, W., Chen, L.: Molecular classification of cancer types from microarray data using the combination of genetic algorithms and support vector machines. FEBS Letters 555(2), 358–362 (2003)
13. Rakotomamonjy, A.: Variable selection using svm-based criteria. Journal of Machine Learning Research 3, 1357–1370 (2003)
14. Weston, J., Elisseeff, A., Scholkopf, B., Tipping, M.: The use of zero-norm with linear models and kernel methods. Journal of Machine Learning Research 3(7-8), 1439–1461 (2003)