Input Selection with Mixed Data Sets: A Simulated Annealing Wrapper Approach

M. Filippone[1], F. Masulli[1], S. Rovetta[1], and S.-P. Constantinescu[2]

[1] *DISI, Dept. Computer and Information Sciences, University of Genova and CNISM*
Via Dodecaneso 35,16146 Genova, ITALY
{filippone,masulli,rovetta}@disi.unige.it
[2] *Faculty of Automatic Control and Computer, University Politehnica of Bucharest*
Splaiul Independentei 313, Bucharest, ROMANIA
constantinescu_sergiu_petre@yahoo.com

Abstract. The input selection problem is approached in a mixed data set framework. Due to the mixed variable types, a stochastic optimization algorithm is required. The proposed technique exploits Simulated Annealing to provide a wrapper-type method, for which experimental evaluation of the quality of variable subsets is a part of the optimization process. Estimation of variable relevance is also provided, based on statistics about the history of each variable in the subsets. The method is demonstrated on a medical diagnosis problem having mixed variables, and comparative experimental results are presented.

1 Introduction

High dimensional data sets often contain redundant subsets of inputs (attributes). The *input selection* problem concerns the search for small relevant subsets of inputs. Input selection procedures (10) can alleviate the *curse of dimensionality* problem (2) arising in the usual situation of data sets characterized by high dimensionality and low cardinality. Moreover, they can improve the knowledge on the problem, e.g., in bioinformatics problems such as gene expression data from DNA microarrays they allow gene selection (i.e. to find the most relevant genes expresses in relation to a pathology), while, with sets of text documents, they can find the most relevant key words.

Input selection algorithms can be broadly divided into two categories (3; 13): *filters* and *wrappers*. Filters evaluate the relevance of each input (subset) using the data set alone, while wrappers invoke a learning algorithm to evaluate the quality of each input (subset). Both approaches, filters and wrappers, usually involve combinatorial searches (often only local) through the space of possible input subsets. Wrappers are usually more computationally demanding, but they can be superior in accuracy when compared with filters.

The strategy for variable selection, and the underlying assumptions about the input variables themselves, is also a design choice. Variables can be selected as a subset with aggregate discriminative power (22; 17), or ranked for their individual relevance (24; 9). In the latter case variables are assumed to be weakly correlated, so that their individual importance can be unambiguously assessed. In the former case, it is assumed that all possible interaction patterns can occur, and this forces a much more complex search space. However, ranks can also be used as an indication to evaluate a subset selection process, in an in-between approach.

The definition of relevance itself can be subject to variations (13), and the goal of the procedure can also be different, with some approaches aiming at comprehensive set (find all significant variables (11; 15; 17)) and others at explanatory sets (this is generally the case with all input selection tasks, where one wants to identify the most important inputs only, as, e.g., in (8)). Again, an in-between approach is possible (24) when an explicit cost function includes both performance and complexity (number of variables) terms. In this case, a continuum of possible balances is provided by the relative weighting of these terms.

To summarize, the input selection problem is here stated as the problem of selecting small subsets of input variables with good explanatory properties improving the knowledge on the problem, and possibly achieving high discriminating power.

These hypotheses form the basis of the method we are presenting in this paper, which is based on optimizing the combination of performance and complexity costs. We propose a wrapper approach to input selection in classification that is able to select small subsets of inputs for data sets containing either numerical and categorical data. The combinatorial search is performed using the Simulated Annealing (SA) method (12) which is a global search method technique derived from Statistical Mechanics and is based on the Metropolis algorithm (16). In principle, any learning algorithms can be chosen depending on the learning task and the type of data base. For dealing with mixed-type variables, we adopt a classification tree.

In the next section, we present the Simulated Annealing technique and describe how we applied it to input selection problem. In Sect. 3, some measures of the input relevance are illustrated. The experimental validation of the proposed input selection method and some heuristics for its speeding-up are shown in Sect. 4. Sect. 5 concerns the discussion and conclusions.

2 SA for gene selection

Simulated Annealing (SA) technique (12) is a global search method technique derived from Statistical Mechanics. SA is based on the Metropolis algorithm (16) proposed to simulate the behavior and small fluctuations of a system of atoms starting from an initial configuration, by the generation of a sequence of iterations. In the Metropolis algorithm each iteration is composed by a random perturbation of the actual configuration and the computation of the corresponding energy variation (ΔE). If $\Delta E < 0$ the transition is unconditionally accepted, otherwise the transition is accepted with probability given by the Boltzmann distribution:

$$P(\Delta E) = e^{-\Delta E/KT} \tag{1}$$

where K is the Boltzmann constant and T the temperature.

In SA this approach is generalized to the solution of general optimization problems (12; 18) by using an *ad hoc* selected cost function (*generalized energy*) instead of the physical energy. SA works as a probabilistic hill-climbing procedure searching for the global optimum of the cost function (21). The temperature T takes the role of a control parameter of the search area (while K is usually set to 1), and is gradually lowered until no further improvements of the cost function are noticed. SA can work in very high-dimensional searches, given enough computational resources.

In this paper we apply SA to the input selection problem with the aim of aggregating an ideally minimal subset of inputs with strong discriminative power. The approach we adopted is to constrain the search space to subsets of variables, and to evaluate a compound cost function combining performance and complexity scores, as previously indicated. The method is described in the following, whereas in Tab. 1 a step-by-step outline of the proposed Simulated Annealing Input Selection (SAIS) algorithm is presented.

Table 1. Simulated Annealing Input Selection (SAIS) algorithm.

```
1. Initialize the parameters;
2. Initialize the binary mask **g** and the temperature $T$;
3. Train and test the classifier and evaluate the generalized system energy $E$;
4. do
5. Initialize $f = 0$ (number of iterations), $h$=0 (number of success);
   (a) do
   (b) Increment the number of iterations $f$;
   (c) Perturb the binary mask **g**;
   (d) Train and test the classifier and evaluate the generalized system energy $E$;
   (e) Generate a random number rnd in the interval [0,1];
   (f) if $rnd < P(\Delta E)$ then
         i. Accept the new binary mask **g**;
        ii. Increment the number of success $h$;
   (g) endif
   (h) loop while $h \leq h_{min}$ and $f \leq f_{max}$;
6. update $T = \alpha T$;
7. loop while $h > 0$;
8. end.
```

Let d be the dimensionality of the input space and $\mathbf{g} = (g_1, g_2, \ldots, g_d)$ be a binary mask representing the system state (configuration), where each bit g_i (with $i = 1, \ldots, d$) corresponds to either selection ($g_i = 1$) or deselection ($g_i = 0$) of an input. The number of bits of **g** set to 1 is denoted by s (i.e., $s \equiv |\mathbf{g}| = \sum_{i=1}^{d} g_i$) .

At Steps 1, 3, and 5, the classifier is trained in the sub-space of selected inputs as defined by the vector mask **g** and the *Classification Error* ε is evaluated using a cross-validation technique (*k-fold cross-validation* (23)). The generalized energy E is defined as a linear combination of ε and of the number of selected inputs s:

$$E = \varepsilon + \lambda\, s \tag{2}$$

Note that the introduction of the number of selected inputs s in the computation of E penalizes situations in which the number of selected inputs is too high. The trade-off between size of input space and accuracy is controlled by the parameter λ (*penalization coefficient*). If $\lambda = 0$ we favor solutions with low classification error ε, without taking the dimensionality of the input space into account, while high values of λ can led to solutions with few input variables, but with a large classification error.

Moreover, due to the redundancy of groups of input variables (14) and to curse of dimensionality problem (2), often a good tuning of the penalization coefficient can allow us to find a very small input space supporting a small classification error.

As suggested in (18), at Step 2 the initial temperature T is obtained as the mean variation of generalized energy (ΔE) over an assigned number p of random initializations of **g**. The vector mask **g** is initialized by randomly setting s_0 bits to 1 and leaving the remaining $d - s_0$ to 0.

A perturbation or move (Step 5c) is obtained in the following way: w bits of **g** set to 1 are switched to 0, and v bits of **g** set to 0 are switched to 1, where the values of w and v are extracted with uniform distributions, respectively in the (integer) intervals $[w_{\min}, w_{\max}]$ and $[v_{\min}, v_{\max}]$.

In the experiment described in this paper, the w bits of **g** set to 1 to be switched to 0 and the v bits of **g** set to 0 to be switched to 1, at the move step of SAIS (Step5c), are selected using a uniform distribution in the (integer) interval $[1, s]$, i.e., with probability

$$p_i = \frac{1}{s} \quad \textit{(Uniform Selection)}, \tag{3}$$

Table 2. Parameters of SAIS algorithm and their values selected for the experiments presented in Sect.s 4 and 5.

Symbol	**Meaning**	**Values**
s_0	Number of inputs initially selected	5
p	Number of initializations of **g** for estimating the initial value of T	10000
$[w_{\min}, w_{\max}]$	Interval for w	$[1, s]$
$[v_{\min}, v_{\max}]$	Interval for v	$[1, d-s]$
λ	Penalization coefficient	10^{-4}, 10^{-2}
f_{max}	Maximum number of iterations for each T	100
h_{min}	Minimum number of successes for each T	30
α	Cooling parameter	0.9
γ	Aging constant	0.98

and, similarly, the v bits of **g** set to 0 to be switched to 1, are selected using an uniform distribution in the interval $[1, d-s]$, i.e., with probability

$$p_i = \frac{1}{d-s} \quad \textit{(Uniform Selection).} \tag{4}$$

In Tab. 2, the list of the parameters to be initialized at Step 1 is presented together with the values selected for the experiment that we will describe in Sect. 4. Namely, s_0 and p are used at Step 2 for the initialization of **g** and T; $[w_{\min}, w_{\max}]$ and $[v_{\min}, v_{\max}]$ constrain the size of move (Step 5c); λ is used for computing the Generalization Energy E (Steps 3 and 5d); f_{max}, h_{min}, α are used for the SA scheduling (Steps 5 and 6); γ is used for estimating the input relevances (see Sect. 3).

As already stated, the SAIS algorithm aims to find a small subset of variables, with high discriminant capability, by exploiting the redundancy of subsets of variables and penalizing solutions with high input dimensionality. To this aim s_0 should be selected of the order of the estimated input dimensionality, while intervals $[w_{\min}, w_{\max}]$ and $[v_{\min}, v_{\max}]$, regulating the variability of perturbation, can be changed during the algorithm to favor solutions with small input space.

SAIS is a computationally intensive algorithm, but it is able to work with both numerical and categorical inputs. It is also worth noting that, due to the random nature of SA, each time we run the SAIS algorithm we can find a new subset of s inputs from the original d.

3 Evaluating input relevance

As noticed in Sect. 1, many definitions of input relevance have been presented in the literature, with different meaning, motivation, and depending on the numerical or categorical type of inputs (13; 11; 15; 17; 8; 24). Here we present some new definitions that apply both to numerical and categorical variables.

SA is an algorithm implementing a stochastic time-varying dynamical system where the state vector evolves in the direction of the minima of the generalized energy function. In our case during the evolution of the SAIS algorithm, the bits set in the state vector **g** will be related to the more relevant inputs with increasing probability.

The inputs which are more relevant for classification should appear soon in the set of bits of **g** set to 1 and will be as more frequent as the temperature decreases. In order to estimate the relevance of inputs, we can include in SAIS an aging algorithm. To this end, we can define a vector $\mathbf{r} = (r_1, r_2, \ldots, r_d)$. At Step 1 of the SAIS algorithm, we set $r_i = 0\ \forall i$. Every time a

Table 3. Inputs selected in each run of the SAIS algorithm on Cleve data set using as penalization parameter $\lambda = 10^{-2}$ in the generalized energy. For each run we show also ε (classification error) and m (number of misclassified instances).

	Run 1	**Run 2**	**Run 3**	**Run 4**	**Run 5**	**Run 6**	**Run 7**	**Run 8**	**Run 9**	**Run 10**
ε**;** m	0.18; 55	0.18; 55	0.18; 54	0.18; 55	0.18; 55	0.15; 45	0.18; 55	0.18; 54	0.18; 55	0.18; 55
Input 1	ncol	ncol	ncol	ncol	ncol	ncol	ncol	ncol	ncol	thal
Input 2	thal	chest	ches	thal	thal	thal	thal	chest	chest	ncol
Input 3	chest	thal	induced	chest	chest	induced	chest	induced	thal	chest
Input 4	–	–	oldpeak	–	–	Sex	–	oldpeak	–	–
Input 5	–	–	–	–	–	Age	–	–	–	–
Input 6	–	–	–	–	–	bps	–	–	–	–

Table 4. Inputs selected in each run of the SAIS algorithm on Cleve data set using as penalization parameter $\lambda = 10^{-4}$ in the generalized energy. For each run we show also ε (classification error) and m (number of misclassified instances).

	Run 11	**Run 12**	**Run 13**	**Run 14**	**Run 15**	**Run 16**	**Run 17**	**Run 18**	**Run 19**	**Run 20**
ε**;** m	0.16; 48	0.15; 44	0.14; 42	0.17; 50	0.16; 49	0.14; 42	0.15; 44	0.15; 45	0.14; 42	0.15; 44
Input 1	ncol	ncol	ncol	ncol	ncol	ncol	ncol	ncol	ncol	ncol
Input 2	induced	thal	Age	induced	induced	induced	thal	thal	induced	thal
Input 3	Sex	Sex	induced	Sex	thal	Sex	induced	induced	thal	induced
Input 4	bps	induced	thal	bps	ecg	thal	Sex	ecg	Sex	Sex
Input 5	thal	Age	Sex	slope	Sex	Age	ecg	fbs	Age	ecg
Input 6	Age	bps	bps	–	oldpeak	bps	Age	Sex	bps	bps
Input 7	chol	ecg	–	–	Age	–	bps	bps	–	Age
Input 8	–	–	–	–	bps	–	–	Age	–	–

perturbation is accepted according to the Boltzmann distribution (Step 5.f), we update $\mathbf{r}$ as follows:

$$\mathbf{r} = \gamma \mathbf{r} + \mathbf{g} \tag{5}$$

where γ is the aging constant chosen in the (real) interval [0,1].

At the end of the SAIS the vector $\mathbf{r}$ measures how long each input has been selected in the last few successful moves of the algorithm. We give to vector $\mathbf{r}$ an interpretation as vector of input relevances (*aged relevance*).

We also introduce the concepts of *voted relevance* and of *soft-voted relevance* defined as the sums of the input masks $\mathbf{g}$ and of vectors $\mathbf{r}$ obtained at the ends of an assigned number m of SAIS runs, i.e., as, respectively, $\sum_j \mathbf{g}_j$ and $\sum_j \mathbf{r}_j$ (with $j = 1, 2, ..., m$).

It is worth noting that none of those relevance notions are exploited for making decisions in the SAIS algorithm.

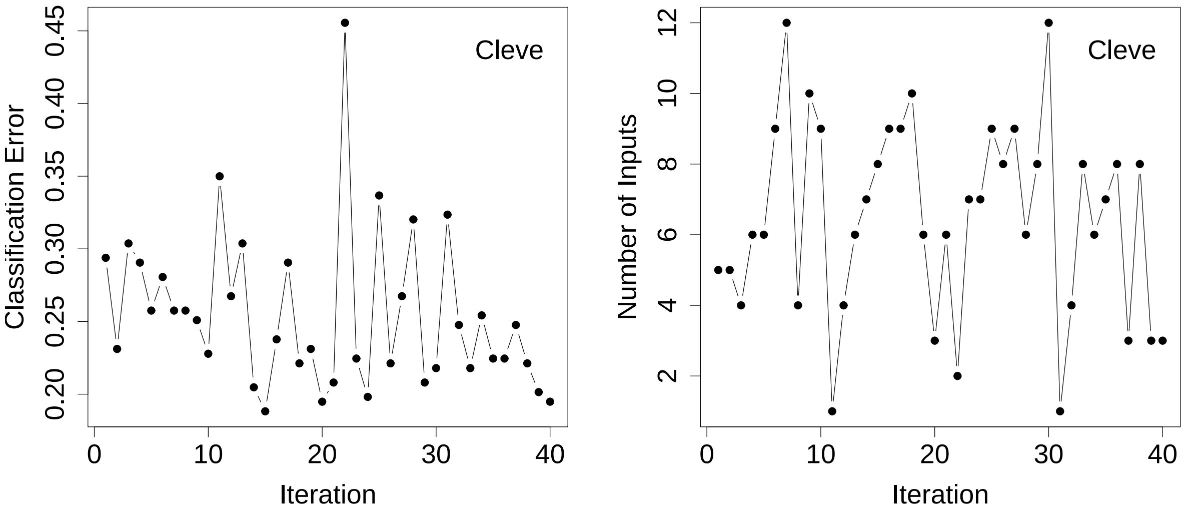
4 Experimental validation of SAIS

We report here an experimental validation of SAIS performed on the *Cleve* mixed data set which is modified from the Detrano's heart disease data set which contains 76 clinic attributes for each patient. Cleve data set is composed by 303 instances (patients) with six categorical and eight numerical inputs[3]. SAIS algorithm has been implemented in R (20) under the Linux operating system.

We focus on the problem of finding a set of features able to distinguish between health and disease.

We chose this data base, as it has been well studied in the literature and can allow us to make a deeply comparison on a mixed data set of the performances of SAIS with those of previously studied models.

[3] http://www.ics.uci.edu/ mlearn/databases/heart-disease/cleve.mod.



(a)

(b)

Figure 1. SAIS on the Cleve data set. Classification error ε and Number of selected inputs s versus the iteration number for Run 1.

In all the runs of SAIS on the Cleve data set, the classification error ε has been evaluated using 10-fold cross-validation. In Tab.s 3 and 4 we show the results of 20 independent runs of SAIS using the assumptions in Tab. 2 and a RPART classification tree (4; 19) as a classifier. The classification tree has been employed here as it is able to handle both numerical and categorical inputs.

In the runs of SAIS shown in Tab. 3 we chose a penalization coefficient $\lambda = 10^{-2}$, while in those of Tab. 4 we used $\lambda = 10^{-4}$. In both tables the inputs selected are ranked following the valuation of their aged relevance.

Using the value of the penalization coefficient $\lambda = 10^{-2}$, we obtained solutions with higher classification error ε and few selected inputs. With $\lambda = 10^{-4}$, we obtained solutions with better accuracy and higher dimension of the selected input space. In the former case the first inputs, ranked using both soft voted and voted relevances, were: *ncol* (number of vessels colored), *thal* (norm, fixed, rever), *chest* (chest pain type, namely: angina, abnang, notang, asympt), and *induced* (exercise induced angina). In the latter case the first inputs, ranked using voted relevance, were: *sex*, *induced*, *ncol*, and *ecg* (resting ecg, namely: norm, abn, hyper); while, ranking according to soft voted relevance, the first inputs were: *ncol*, *chest*, *induced*, and *thal*.

In Fig. 1(a) and Fig. 1(b) the behavior of the classification error ε and the number of selected inputs are plotted versus the iteration number of the algorithm in run number 11. Each iteration corresponds to a different value of temperature T (i.e. Step 5 and Step 6 in Tab. 1).

5 Discussion and Conclusions

In this paper we have described SAIS, a variable subset selection method of the wrapper type, based on Simulated Annealing (12), and therefore implementing a global search strategy. SAIS can work either with numerical or categorical inputs depending on the associated learning machine. The output of the SAIS algorithm is a relevant subset of inputs, where relevance is experimentally assessed by using a supervised classifier during the evaluation phase of the algorithm. In addition we provided some estimates of individual relevances of inputs within the output subset.

On the Cleve mixed data set, the proposed input selection method finds minimal subsets of features with good accuracy in classification and giving new insights on the medical problem. Moreover, in (7) SAIS has been tested on two bioinformatics data sets, namely on the 7129-dimensional Leukemia data set by Golub et al. (8) and on the 2000-dimensional Colon data set by Alon et al. (1). On both data sets it finds solutions characterized by a small number of selected genes and by good discriminant capability (null classification error on Leukemia data set).

The results can be obtained semi-automatically with few parameter selection iterations, and the method performs classification and input selection and ranking jointly.

SAIS can employ each kind of learning machine, included clustering algorithms. In (6), e.g., we run SAIS method associated to Fuzzy c-means on a Leukemia data base obtaining minimal sets of inputs able to perform unsupervised clustering with null representation error.

Acknowledgment

Work funded by the Italian Ministry of Education, University and Research (2004 “Research Projects of Major National Interest”, code 2004062740). We thanks Giorgio Valentini for useful discussions.

Bibliography

[1] U. Alon, N. Barkai, D. A. Notterman, K. Gish, S. Ybarra, D. Mack, and A. J. Levine, Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays, *Proc. Natl. Acad. Sci.*, USA, vol.96 no. 12 pp. 6745–6750, 1999.

[2] R. Bellman, *Adaptive Control Processes: A Guided Tour*, Princeton University Pressv, 1961.

[3] A. Blum and P. Langley, Selection of Relevant Features and Examples in Machine Learning, *Artificial Intelligence*, vol. 97, nos. 1–2, pp. 245–271, 1997.

[4] L. Breiman, J. Friedman, R. Olshen, and C. Stone, *Classification and Regression Trees*, Wadsworth & Brooks, Pacific Grove, CA., 1984.

[5] C. Cortes and V. Vapnik, Support vector networks, *Machine Learning*, vol. 20 pp. 273 – 297, 1995.

[6] M. Filippone, F. Masulli, and S. Rovetta, Unsupervised gene selection and clustering using simulated annealing, In I. Bloch, A. Petrosino, and A. Tettamanzi, ed.s, *WILF*, volume 3849 of *Lecture Notes in Computer Science*, pp. 229–235, Springer, 2005.

[7] M. Filippone, F. Masulli, S. Rovetta, Supervised Classification and Gene Selection Using Simulated Annealing, *Proceedings of the International Joint Conference on Neural Networks 2006, Vancouver - Canada*, IEEE Computational Intelligence Society, Piscataway, NJ, USA.

[8] T. Golub, D. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. Mesirov, H. Coller, M. Loh, J. Downing, M. Caligiuri, C. Bloomfield, E Lander, Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring, *Science* vol. 286, pp. 531–537, 1999.

[9] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, Gene selection for cancer classification using support vector machines, *Machine Learning*, vol. 46 no.s 1–3, pp. 389–422, 2002.

[10] I. Guyon, A. Elisseeff, *An introduction to variable and feature selection*, *Journal of Machine Learning Research*, vol. 3 pp. 1157–1182, 2003.

[11] K. Kira and L. Rendell, The Feature Selection Problem: Traditional Methods and a New Algorithm, Proc. 10th Nat l Conf. Artificial Intelligence (AAAI-92), pp. 129–134, 1992.

[12] S. Kirkpatrick, C.D. Gelatt, and M.P. Vecchi. Optimization by simulated annealing. *Science*, vol. 220, pp.661–680, 1983.

[13] R. Kohavi and G. John, Wrappers for Feature Subset Selection, *Artificial Intelligence*, vol. 97, nos. 1–2, pp. 273–324, 1997.

[14] D. Koller and M. Sahami, Toward Optimal Feature Selection, *Proceedings of the Thirteenth International Conference (ICML '96)*, Lorenza Saitta, Ed., pp.284–292, Morgan Kaufmann, 1996.

[15] I. Kononenko, *Estimating Attributes: Analysis and Extensions of RELIEF*, Proc. Seventh European Conf. Machine Learning, pp. 171–182, 1994.

[16] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller. Equation of state calculations for fast computing machines. *Journal of Chemical Physics*, vol. 21, pp. 1087–1092, 1953.

[17] C. Moneta, G.C. Parodi, S. Rovetta, and R. Zunino, Automated diagnosis and disease characterization using neural network analysis, in *Proceedings of the 1992 IEEE International Conference on Systems, Man and Cybernetics - Chicago, IL, USA*, pp. 123-128, 1992.

[18] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling, *Numerical Recipes in C*, (2nd Edition), Cambridge University Press, 1992.

[19] B. D. Ripley, *Pattern recognition and neural networks*. Cambridge university press, Cambridge, 1996.

[20] R Development Core Team. *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria, 2005. ISBN 3-900051-07-0. http://www.R-project.org.

[21] F. Romeo and A. Sangiovanni-Vincentelli. *Probabilistic Hill-Climbing Algorithms: Properties and Applications*. Computer Science Press, Chapell Hill, NC, 1985.

[22] N. Slonim and N. Tishby, Agglomerative information bottleneck, in *Advances in Neural Information Processing Systems*, pages 617-623, 2000.

[23] M. Stone, Cross-validatory choice and assessment of statistical predictions, *Journal of the Royal Statistical Society*, B, vol. 36, no. 1, pp.111–147, 1974.

[24] J.Weston, A. Elisseff, B. Schoelkopf, and M. Tipping, Use of the zero norm with linear models and kernel methods, *Journal of Machine Learning Research*, vol. 3, pp. 1439-1461, 2003.