Feature Selection Using Counting Grids: Application to Microarray Data

Pietro Lovato[1], Manuele Bicego[1], Marco Cristani[1],
Nebojsa Jojic[2], and Alessandro Perina[2]

[1] Computer Science Department, University of Verona (ITALY)
[2] Microsoft Research (US)

Abstract. In this paper a novel feature selection scheme is proposed, which exploits the potentialities of a recent probabilistic generative model, the Counting Grid. This model is able to cluster together similar observations, highlighting the compactness of a class and its underlying structure. The proposed feature selection scheme is applied to the expression microarray scenario, a peculiar context with very few patterns and a huge number of features. Experiments on benchmark datasets show that the proposed approach is effective and stable, assessing state-of-the-art classification accuracies.

Keywords: feature selection, gene selection, generative models.

1 Introduction

Feature selection techniques definitely represent an important class of preprocessing tools in many Pattern Recognition applications: such methods, by eliminating uninformative features, can reduce the dimension of the problem space, thus alleviating the curse of dimensionality issue [1]. Further, there are application fields – like biology, where everyday lab procedures generate enormous amount of data to be processed – where it is inconceivable to devise an analysis procedure which does not comprise a feature selection step. A clear example can be found in the analysis of expression microarray data, where the expression level of thousands of genes is simultaneously measured. A typical classification task implies few dozens of samples, each one characterized by the expression level of thousands of genes (i.e. few points in a huge dimensional space). In this context, feature selection techniques are even more important, since they can help the medical/biological researchers in identifying a stable and informative set of biomarkers for cancer diagnosis, prognosis, and therapeutic targeting [2, 3].

A large amount of approaches have been introduced in the past in the feature selection field. Broadly, they can be divided in three major classes, depending on how they interact with the classification technique. Filter approaches do not interact with the classifier system, and perform selection just by looking at the intrinsic properties of data. Usual examples are ranking of the features according to criteria which spans from simple variance up to complicates statistics [2, 4].

G.L. Gimel' farb et al. (Eds.): SSPR & SPR 2012, LNCS 7626, pp. 629–637, 2012.

Wrapper methods interact with a specific model trained on the subset of features, using metrics such as the classifier performance / error estimate to assess the quality of the selected features. Finally, in embedded techniques, the search for an optimal subset of features is built into the classifier construction. In the popular SVM-RFE algorithm [5], the weight given to each feature by the SVM classifier is used as a score to rank features, from the most important to the less important. In the specific field of expression microarray – where the feature selection is called gene selection – a common problem of most methods proposed in the past is the stability of the extracted features/genes: actually, datasets which differ by a few samples can lead to complete different sets of genes selected by the feature selection algorithm, still guaranteeing good classification performances [6]. This issue has been often disregarded and has been addressed only recently [7, 8].

This paper presents a novel feature selection scheme, which is based on the Counting Grid (CG) model [9] – a probabilistic model which clusters together similar observations, highlighting the compactness of a class and its underlying structure. The proposed approach is specifically thought for the microarray scenario, which is characterized by the presence of few points in a very high dimensional space. In fact, in [9] it has been show that CGs provide a rich and powerful description of a microarray dataset: samples and gene expressions can be placed on an N-dimensional grid; samples coming from the same class are placed close together in this grid, allowing easy and interpretable visualization of the transition from one class to the other, which turns out to be smooth in most of the cases. In this paper we make one step ahead along this direction, proposing a method which starts from the embedding of the data into the grid and permits to gain insights into which genes characterize a particular class. In fact, starting from the dense embedding of the data provided by the CG, *i)* we embed the class label on the grid, *ii)* we highlight the directions of maximum variation between classes by means of directional derivatives, and finally *iii)* we rank the genes based on how much they vary along these directions. Eventually the ranking is used to extract a stable set of genes for classification or biomarker identification. A further important note concerns the assumption made by most of the gene selection techniques about the independence between genes (actually the typical approach is to rank individually the genes): actually this assumption oversimplifies the complex relationship between genes – which are well known to interact with each other through gene regulative networks. Therefore, models like Counting Grid which can measure and consider the relation and the influence between genes should be preferred.

The experimental evaluation, performed on well-known datasets and compared with state-of-the-art methodologies, shows the suitability of the proposed approach in terms of classification accuracy. Furthermore, to assess the stability of the selected genes, we show that slight alterations in the composition of the training set do not change the selected features, giving confidence that the genes may be somehow involved in the pathology of interest.

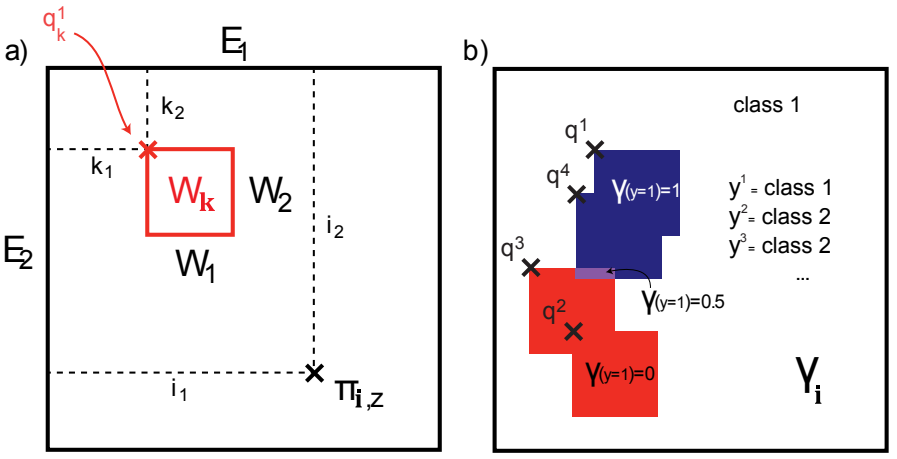

Fig. 1. a) An example of a counting grid geometry. b) Label Embedding $\gamma_{\mathbf{i}}$.

2 Background: Counting Grid Model

In Pattern Recognition, data samples are often represented as bags of features without particular order; each t-th observation is characterized by a vector – often called count vector $\{c_z^t\}$ – containing the number of occurrences of each feature z [10, 11]. For example, a text document may be described by the number of occurrences of the different words it contains (or an image with the number of occurrences of different visual features it contains). This choice is often motivated by the difficulty or computational efficiency of modeling the known structure of the data. Concerning microarray, it has been shown in [12–14] that the bag-of-features representation is well-suited also for microarray data, providing interpretable and descriptive signatures. Each sample can be seen as an independent observation; the gene expression value is then interpreted as the "count" of that gene in the sample: the higher the expression level, the "more present" the gene is in such experiment.

The counting grid model, recently introduced in [9], is a generative model for such representations. Formally, the basic counting grid $\pi_{\mathbf{i},z}$ is a set of normalized counts of features indexed by z on the 2-dimensional[1] discrete grid indexed by $\mathbf{i} = (i, j)$ where $i \in [1 \dots E_1]$, $j \in [1 \dots E_2]$ and $\mathbf{E} = [E_1, E_2]$ describes the extent of the counting grid. Since π is a grid of distributions, $\sum_z \pi_{\mathbf{i},z} = 1$ everywhere on the grid (see Fig.1a for an illustration).

A given bag of features, represented by counts $\{c_z\}$ is assumed to follow a count distribution found in a patch of the counting grid. In particular, using a window of dimensions $\mathbf{W} = [W_1, W_2]$, each bag can be generated by first selecting a position $\mathbf{k}$ on the grid and then by placing the window in the grid such that $\mathbf{k}$ is its upper left corner. Then, all counts in this patch are averaged

[1] N-dimensional in general, here we focus on 2 dimensions.

to form the histogram $h_{\mathbf{k},z} = \frac{1}{W_1 \cdot W_2} \sum_{\mathbf{i} \in W_\mathbf{k}} \pi_{\mathbf{i},z}$, and finally a set of features in the bag is generated. In other words, the position of the window $\mathbf{k}$ in the grid is a latent variable given which the probability of the bag of features $\{c_z\}$ is

$$p(\{c_z\}|\mathbf{k}) = \prod_z (h_{\mathbf{k},z})^{c_z} = \frac{1}{W_1 \cdot W_2} \prod_z (\sum_{\mathbf{i} \in W_\mathbf{k}} \pi_{\mathbf{i},z})^{c_z}$$

where with $W_\mathbf{k}$ we indicate the particular window placed at location $\mathbf{k}$ (see Fig. 1a). We will also often refer to the ratio of the CG area and the window area $\kappa = \frac{E_1 \cdot E_2}{W_1 \cdot W_2}$, as the capacity of the model.

Computing and maximizing the log likelihood of the data turns to be an intractable problem; therefore it is necessary to employ an iterative EM algorithm. The E step aligns all bags of features to grid windows, to match the bags' histograms, inferring $q^t_\mathbf{k} \propto \exp \sum_z c^t_z \cdot \log h_{\mathbf{i},z}$, i.e., where each bag maps on the grid. In the M-step the model parameter, i.e. the counting grid π, is re-estimated. To avoid severe local minima it is important to consider the Counting Grid as a torus, and perform all windowing operation accordingly. For details on the learning algorithm and on its efficiency see [9].

3 The Proposed Approach

Once a Counting Grid is learned, each sample can be mapped on it through $q^t_\mathbf{k}$, which represents a map telling which part of the CG has more likely generated the pattern t. As a first step of our procedure, we can map all samples belonging to the same class to the CG, trying to obtain a class-related averaged map. This step in [9] has been called class labels embedding, where the goal was to embed the samples' class labels $y^t = l, l = [1, \ldots, L]$ to obtain a posterior probability of each class $p(l|\mathbf{i}) = \gamma_l(\mathbf{i})$ in each position $\mathbf{i}$: this indicates which positions of the CG better "explain" that class. This is achieved using the posterior probabilities $q^t_\mathbf{k}$ already inferred like illustrated in Fig.1b and described by Eq.1

$$\gamma_l(\mathbf{i}) = \frac{\sum_t \sum_{\mathbf{k}|\mathbf{i} \in W_\mathbf{k}} q^t_\mathbf{k} \cdot [y^t = l]}{\sum_t \sum_{\mathbf{k}|\mathbf{i} \in W_\mathbf{k}} q^t_\mathbf{k}} \tag{1}$$

where $[\cdot]$ is the indicator function, which returns 1 if sample t belongs to class l and 0 otherwise. Roughly speaking, the main idea is to "average" all the mappings $q^t_\mathbf{k}$ of the training samples belonging to a given class. If the CG is able to capture the underlying behaviour of a specific class, then all the mappings will be more or less coherent, and only a part of this averaged map will be different than zero, possibly in a spatially coherent small region – the region which more likely "explains" the training patterns of that class. In order to clarify this concept, in Fig. 2a we show the label embedding for the prostate cancer dataset [15], which comprises two classes. In the figure the tumoral class is embedded. Please observe that the active (non zero) locations are all grouped in spatially coherent zones of the averaged map. Therefore, even if the labels are not used

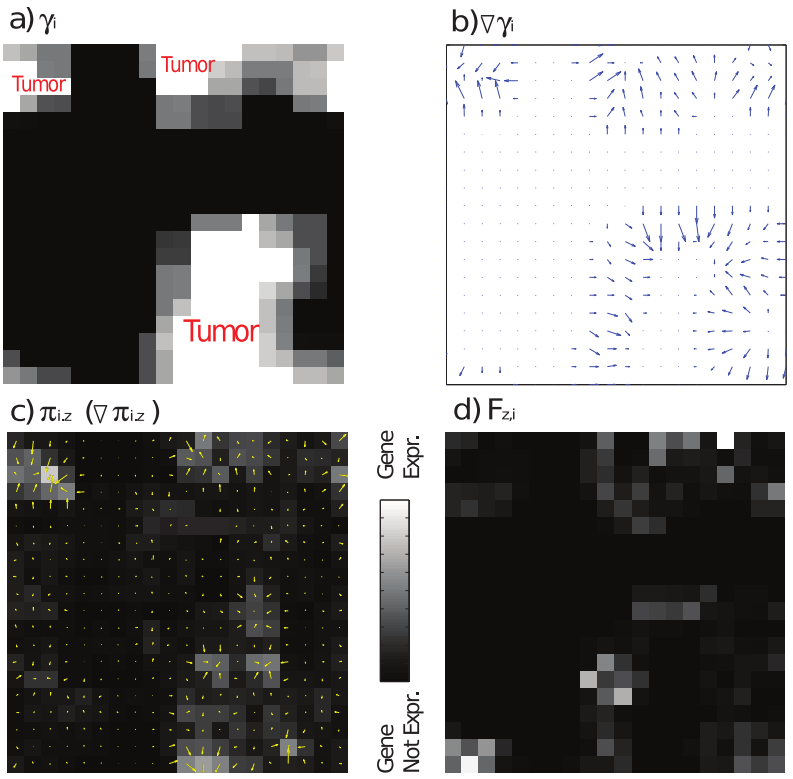

Fig. 2. a) Label embedding $\gamma_{\mathbf{i}}$. b) Gradient of the embedding. c) Counting grid for a particular gene (π_z) and its gradient. d) $F_{z,\mathbf{i}}$.

during the learning of the CG, tumoral and non-tumoral samples are naturally separated (since we are in a two class problem, the embedding of non tumoral clas is simply obtained by reversing this image); this suggests that indeed CGs are suitable to describe the latent structure which generates the data.

As a second step, we compute the gradient of the embedding, $\nabla\gamma_{\mathbf{i}}$, which returns information about where and how the classes separates (see Fig.2b). In this case the idea is to find which are the regions in the CG where the first class "translates" to the second class or vice versa. Please note that, in the two class case, we only need to compute the gradient on one map, since the map of the second class is just the complementary of the first. Even if the generalization to the multiclass case is somehow straightforward (for example 1 versus all embeddings, or others), for simplicity here we present the two class case.

As a final step, to get the feature score F_z, upon which we will base our feature selection strategy, we rank the genes depending on how much their expression vary along the borders between the classes. The idea is straightforward: to discriminate between the two classes the most useful features are the ones which vary most where we have the class transition. For example in Fig.2c we show

for a particular gene $\hat{z}$ the map $\pi_{\hat{z},\mathbf{i}}$, which represents where that gene is more expressed in the grid. We also show its gradient in each position (yellow arrows). After a quick glance at Fig.2b one can convince himself that the expression of $\hat{z}$ is mostly expressed in tumoral samples and often varies where a transition between tumoral and non-tumoral samples is present; that suggets that the gene is important for classification and related to the disease.

To capture this idea mathematically we compute the directional derivatives of the $\pi_{z,\mathbf{i}}$ in the direction $\boldsymbol{v}$ of the gradient of the class embedding $\boldsymbol{v} = \nabla\gamma_{\mathbf{i}}$ and we sum over all the locations $\mathbf{i}$ in the grid. To reward more the variation in expression where we have a high variation between classes, we also multiply by the module of v.

In formulae we have that the feature score is equal to:

$$F_z = \sum_{\mathbf{i}} \left| |\boldsymbol{v}| \cdot \frac{\boldsymbol{v}}{|\boldsymbol{v}|} \cdot \nabla\pi_{z,\mathbf{i}} \right| = \sum_{\mathbf{i}} \left| \boldsymbol{v} \cdot \nabla\pi_{z,\mathbf{i}} \right| \tag{2}$$

In Fig.2d we show that $F_{\hat{z},\mathbf{i}} \neq 0$ only along the borders between the 2 classes. F_z represents the rank score of every feature, which permits to order the genes from the most prominent (i.e. the one which varies the most in the direction of "transition" of the classes) to the least.

Summarizing, the proposed approach consists in the following steps:

1. Training of the Counting Grid on the whole dataset (generative step, labels are not used)
2. Label embedding of the training samples of one of the two classes
3. Computation of the gradient of the map, which estimates the regions of the maps where there is the transition from one class to the other
4. Computation in such zones of the gradient of the genes
5. As a final score, each gene is ranked by its averaged variation in the direction where the two classes vary most.

4 Experimental Evaluation

We tested the proposed approach on two well-known microarray benchmark datasets for two-class problems; a brief description can be found on table 1.

Table 1. Summary of the datasets used

Name	N. Features (genes)	N. Samples	Reference
Colon	2000	62 (40-22)	[16]
Prostate	6033	102 (50-22)	[15]

Since, as a base level, we are mostly interested in the quality of unsupervised learning of the distributions over the microarray samples, the whole dataset

Table 2. Classification results (AUC) for the dataset used

Colon dataset

	Gene Signature Size				
Sel. Method	10	50	100	150	200
SVM-RFE [8]	76.4	77.5	79.2	79.4	80.1
Ens.SVM-RFE [8]	80.3	79.4	78.6	78.6	79.4
SW SVM-RFE [8]	79.5	81.2	78.4	76.2	76.2
ReliefF [8]	78.8	80.1	78.5	77.5	76.1
Ens. ReliefF [8]	78.9	80.2	79.1	77.3	76.1
SW ReliefF [8]	78.3	79.6	78.1	76.4	75.4
[7]	**85.0**	86.0	87.0	87.5	86.5
Our method	81.38	**89.53**	**89.64**	**89.25**	**88.97**

Prostate dataset

	Gene Signature Size				
Sel. Method	10	50	100	150	200
SVM-RFE [8]	89.8	91.3	92.1	92.1	92.2
Ens.SVM-RFE [8]	92.9	92.0	92.0	92.6	92.7
SW SVM-RFE [8]	93.4	91.3	90.0	90.7	91.2
ReliefF [8]	93.3	93.0	91.4	91.4	91.7
Ens. ReliefF [8]	93.4	92.4	91.4	91.0	91.9
SW ReliefF [8]	93.3	92.7	91.4	91.3	91.4
[7]	**95.5**	**96.0**	**95.0**	94.0	94.0
Our method	78.21	88.30	92.45	**94.99**	**95.73**

has been used to train a CG (of course labels are ignored in this phase), in a transductive way [14, 17]. Then, in order to have a fair comparison with the state-of-the-art, we adopted the testing protocol of [8]: the data set was randomly split 2:3/1:3 (training/testing). Labels have been embedded in the Counting Grid, the score F_z has been calculated for each gene z and the top-ranked genes have been extracted, ranging in the values [10 50 100 200]. In order to have a fair evaluation, the gene ranking has been calculated using only the training samples, and applied to the testing samples. The classification is performed using a linear SVM with the parameter C = 1, using the area under the ROC curve (AUC) as an estimate for the classification performance. The test has been repeated 100 times, and the mean of the computed AUCs is shown in table 2, along with comparative state-of-the-art results (see the references between brackets). As for the Counting Grid size, we varied its dimensions by selecting κ between 5 and 40, reporting in the table the mean of the obtained AUCs.

From table 2 it is evident that the proposed approach produces results comparable, and in many cases superior, with state-of-the-art techniques. Furthermore, we assessed the stability of the selected features using the Kuncheva index [18]. The idea is to compare the subsets of genes extracted while varying the training/testing splitting. Given two sets of features $\mathbf{f}_1$ and $\mathbf{f}_2$, the stability index is defined as follows:

Table 3. Stability of the proposed approach

Colon dataset

	Gene Signature Size				
Feat. Sel.	10	50	100	150	200
Best [8]	0.78	0.75	0.70	0.69	0.67
[7]	0.65	0.59	0.58	0.61	0.62
Our method	**0.94**	**0.92**	**0.92**	**0.91**	**0.91**

Prostate dataset

	Gene Signature Size				
Feat. Sel.	10	50	100	150	200
Best [8]	0.68	0.65	0.68	0.68	0.69
[7]	0.72	0.72	0.73	0.72	0.71
Our method	**0.90**	**0.94**	**0.96**	**0.96**	**0.96**

$$KI(\mathbf{f}_1, \mathbf{f}_2) = \frac{r - (s^2/N)}{s - (s^2/N)} \tag{3}$$

where s denotes the signature size, $r = |\mathbf{f}_1 \cap \mathbf{f}_2|$ and N is the total number of genes in the dataset. The Kuncheva index takes values in [-1, 1], and the higher its value, the larger the number of commonly selected genes in both signatures. The index is shown in Table 3, for our approach and other methods. Since the proposed approach is aimed at explaining the data through a generative model, and labels are used later on, the stability index is very high: for both datasets and all different signature sizes, it is always above 0.9, while the best result found in the references we used for comparison is 0.78.

5 Conclusions

In this paper we presented a filter algorithm to perform feature selection, which is based on the recently proposed Counting Grid generative model. The representation given by this model in terms of patterns placed on a 2-dimensional grid has been tailored to derive a new feature selection algorithm. We applied the proposed approach to expression microarray data validating through a series of experiments on benchmark microarray datasets found in the literature. Obtained results were satisfactory.

References

1. Duda, R., Hart, P., Stork, D.: Pattern Classification, 2nd edn. John Wiley & Sons (2001)
2. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. Journal of Machine Learning Research 3, 1157–1182 (2003)

3. Saeys, Y., Inza, I., Larraaga, P.: A review of feature selection techniques in bioinformatics. Bioinformatics 23(19), 2507–2517 (2007)
4. Thomas, J., Olson, J., Tapscott, S., Zhao, L.: An efficient and robust statistical modeling approach to discover differentially expressed genes using genomic expression profiles. Genome Research 11, 1227–1236 (2001)
5. Guyon, I., Weston, J., Barnhill, S., Vapnik, V.: Gene selection for cancer classification using support vector machines. Machine Learning 46, 389–422 (2002)
6. Li, T., Zhang, C., Ogihara, M.: A comprehensive study on feature selection and multiclass classification methods for tissue classifcation based on gene expression. Bioinformatics 20, 2429–2437 (2004)
7. Abeel, T., Helleputte, T., de Peer, Y.V., Dupont, P., Saeys, Y.: Robust biomarker identification for cancer diagnosis with ensemble feature selection methods. Bioinformatics 26, 392–398 (2010)
8. Yu, L., Han, Y., Berens, M.: Stable gene selection from microarray data via sample weighting. IEEE Transaction on Computational Biology and Bionformatics 9, 262–272 (2012)
9. Jojic, N., Perina, A.: Multidimensional counting grids: Inferring word order from disordered bags of words. In: Uncertainty in Artificial Intelligence (2011)
10. Salton, G., McGill, M.: Introduction to Modern Information Retrieval. McGraw-Hill, New York (1983)
11. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. J. Mach. Learn. Res. 3, 993–1022 (2003)
12. Rogers, S., Girolami, M., Campbell, C., Breitling, R.: The latent process decomposition of cdna microarray datasets. IEEE/ACM Transactions on Computational Biology and Bioinformatics (2005)
13. Bicego, M., Lovato, P., Oliboni, B., Perina, A.: Expression microarray classification using topic models. In: SAC, pp. 1516–1520 (2010)
14. Perina, A., Lovato, P., Cristani, M., Bicego, M.: A Comparison on Score Spaces for Expression Microarray Data Classification. In: Loog, M., Wessels, L., Reinders, M.J.T., de Ridder, D. (eds.) PRIB 2011. LNCS, vol. 7036, pp. 202–213. Springer, Heidelberg (2011)
15. Singh, D., Febbo, P., Ross, K., Jackson, D., Manola, J., Ladd, C., Tamayo, P., Renshaw, A., D'Amico, A., et al.: Gene expression correlates of clinical prostate cancer behavior. Cancer Cell 98, 203–209 (2002)
16. Alon, U., Barkai, N., Notterman, D., Gish, K., Ybarra, S., Mack, D., Levine, A.: Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. Proc. Natl. Acad. Sci. 96, 6745–6750 (1999)
17. Vapnik, V.: Statistical Learning Theory. Wiley, New York (1998)
18. Kuncheva, L.: A stability index for feature selection. In: IASTED International Multi-Conference Artificial Intelligence and Applications, pp. 390–395 (2007)