Embedding bags of features in hyper-dimensional grids: the counting grid model

Alessandro Perina, Nebojsa Jojic, Umberto Castellani, Manuele Bicego, and Vittorio Murino

Abstract

When the structure of data is difficult to be modelled – unknown of absent – a popular solution consists of representing the objects in terms of unordered Bags of Features (BoF). While models of BoF typically assume that the features in a single bag come from a limited number of components, we show here that many sets of BoF exhibit a very different pattern of variation than the patterns that are efficiently captured by component mixing. In many cases, in fact, from one bag to the next, some features disappear and new ones appear as if the content slowly and smoothly shifts across samples. Examples of latent structures that describe such ordering are present in text processing, computer vision or bioinformatics. This paper takes this spatial metaphor literally and introduces the Counting Grid (CG) model. Counting grids are a multidimensional grid of feature distributions learned in such a way that the features in a bag can be found as the sum of the features generated in some window into the grid. In the experimental section we exploit the embedding provided by CG to improve classification by evaluating different classification strategies on several applicative scenarios.

A.Perina and N.Jojic are with Microsoft Research Redmond, WA

U.Castellani and M.Bicego are with the University of Verona (Italy)

V.Murino is with the Italian Institute of Technology (Italy) and with the University of Verona (Italy)

Embedding bags of features in hyper-dimensional grids: the counting grid model

I. Introduction

In machine learning research, data samples are often represented as bags of words without particular order. This choice is often motivated by the difficulty or the computational inefficiency of modeling the known structure of the data or because the true structure is unknown or absent [1]–[6].

For example, in information retrieval, the word order in a document is often discarded for efficiency reasons. Similarly, in computer vision, the spatial structure of visual features is largely discarded by most object recognition algorithms, also because difficult to model [3], [4]. But there are also examples of data where the structure is truly unknown. A gene expression array can be modeled as a bag of genes with expression levels simply corresponding to counts, because most of the time little is known about the cellular pathways that employ these genes [2]. Moreover, biology is also abundant with situations where the raw data of interest truly has no structure, like the mammalian immune system which sees the virus inside the cell not as a whole but as a set of disordered peptides, sampled from the viral proteins and presented on cellular surface for immune surveillance [7].

To deal with such data, each bag in the collection can be described by an histogram over an unordered set of basic components, called dictionary – the words in a text corpus, the quantized image features, and so on. This is the so called *Bag of Words* paradigm, where objects are represented by histograms over orderless or exchangeable features [3]–[5].

Collections of unorganized bag of words, or features, are often modeled compactly using mixtures or admixture models. In the first case [8], a data corpus is described by a set of sources, called *centers*, each one being a probability distribution over the features. The modeling assumption is that each sample is modeled by one and only one source. For example, in an hypothetical model of all the computer science conference proceedings, the paper "Implicit Cognitive Processes and Multimedia Content Analysis" recently appeared at ACM Multimedia, would be likely to be modeled by a center representing the proceedings of this conference or Multimedia papers in general.

The second approach is to use an admixture model, such as Latent Semantic Analysis [9], [10] and Latent Dirichlet Allocation [11]. These models have been developed in the text analysis domain and they are widely known as topic models. Given a collection of documents, they learn a small number of topics, which correlate semantically-related terms within a collection by establishing associations between those terms that occur in similar context. A document is assumed to have a mix of words from small subset of topics. There are no strong constraints in how topics are mixed and each topic is generally a tight distribution over the words.

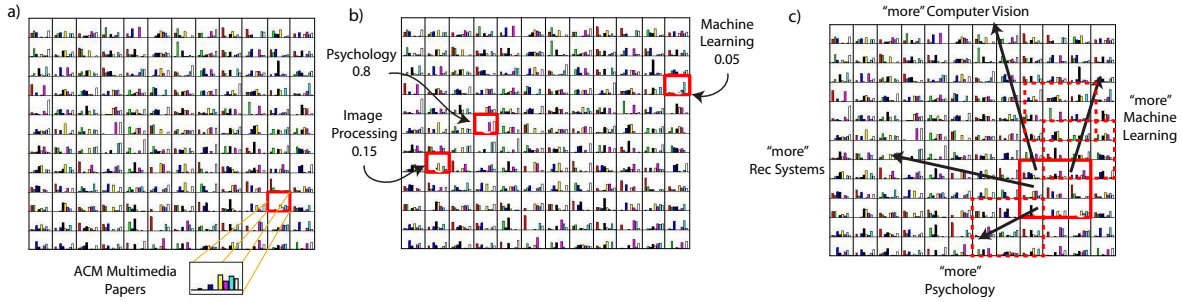

Fig. 1. Different modeling of the paper "Implicit Cognitive Processes and Multimedia Content Analysis". a) *Mixture model*. A document is modeled by a single component. b) *Admixture model*. Samples are modeled by mixing a restricted number of components. c) *Counting Grids*. Samples are modeled by windows in a grid. Sources are arranged on a 2 dimensional grid but a) and b) are not aware of the topology.

Going back to our previous example, in the hypothetical topic model of all the computer science conference proceedings, our previous ACM Multimedia paper, which relates users' favorite images to personality traits, would be modeled by a combination of topics related to psychology, image processing and machine learning. Topic models had a huge success in machine learning and they have been successfully employed in very different applicative domains like computer vision, bioinformatics or medicine [2], [12]–[14].

However, more recently, a new, orthogonal approach emerged: it has been observed that in several cases data samples evolve into one another in a smooth way, with some features dropping and new ones being introduced [15].

The Counting Grid model (CG), took this spatial metaphor – of moving through topics and dropping and picking new words or features – literally. Counting grids are a multidimensional grid of word distributions learned in such a way that each sample's own bag of features can be modeled as the sum of the histograms found in some (hyper)-window into the grid, as illustrated in Fig. 1c for the 2 dimensional case.

Differently from previous models, CGs considers that much of the variability in many interesting datasets is better modeled in terms of multidimensional *thematic shifts*, rather than outright mixing. In fact, in many cases from one bag of words to the next certain words/features are dropped and others added as if the theme slowly and smoothly shifted across subsequent and related documents.

In a corpus like the NIPS proceeding database, for example, a spectrum of papers on the interface between neural science and computer vision can be found, as well as many papers on building neural networks based on the insights from neural science. We can imagine that researchers starting as neural scientists may move in one of these directions and the vocabulary in their papers will smoothly vary. Another example are new stories, where due to the advancement of the date, time-closed news stories are characterized by a smooth change over the theme of the day as certain evolving news stories fall out of favor and new events create new stories. In computer vision this accounts for misalignment of scenes as a change of location in the counting grid. The smooth thematic motion here corresponds to the motion of the camera: few features are lost on one side of the scene, while others appear. This

can be generalized to scenes and visual words [16]. Moreover, very recently, this paradigm also turned out to work very well in some other Computational Biology scenarios [7], [17].

From the machine learning perspective, Counting Grids estimate a possible latent structure that introduces a spatial ordering among bags. This leads to a multidimensional space where the set of tight distributions is embedded. Learning the CG model creates a spatial embedding of the collection of samples such that the closer two samples are on the grid, the larger the overlap between their content is. This is substantially different from other embedding methods based on distances/similarities, such as CODE [18], Local Linear Embedding [19], or Isomap [20], which produce a sparse embedding of bags in an Euclidean space.

Finally, we point out how Counting Grids have some similarity with the Epitome model [21]. Epitomes have originally appeared in computer vision and are based on overlapping patches in the latent space, and also reaped benefits from shift-invariance. However, they relied on the data samples already being ordered into an array (raw image patches [21] or sub-sequences [22], for example), while the Counting Grid model opens this modeling strategy to a much wider set of data types as features can be orderless.

This paper presents the multidimensional counting grid model and extends the conference versions previously appeared in [15], [23]. The first paper, [15] was only concerned with unsupervised learning: in every test, a single model was learned and a Nearest Neighbor classifier in the embedding space was then used to evaluate the model in terms of clustering purity. Counting grids were compared with similar usage of topic models [2], [11], [24]. The second paper [23] was more concerned on classification, proposing a kernel explicitly exploiting the CGs' geometry, with promising performances.

Here we cover several aspects still not investigated and complete the discussion of the model. In particular *i)* we evaluate counting grids as generative classifier in supervised scenarios: this proved to outperform both [15] and [23]. Then *ii)*, despite the "multidimensional" in the title, [15] only considered 2- and 3- dimensional counting grids, and tested few (4-5) complexities[1] for each dimension. Here we better evaluated the effect of the topology of the space, considering dimensionality from 1- to 5-, testing systematically up to 40 complexities per dimension. We refuted some of the findings of [15] and we reached the surprising conclusion that dimensionality doesn't affect classification, and only 1-dimensional counting grids are slightly worse. As further experiment, *iii)* we evaluated the impact of unbalanced windows – when one (few) dimension is significantly smaller than the others – and the impact of models of different grid and window sizes still keeping the same ratio κ – which we dubbed the *capacity* of the model (e.g., 20×20 / 5×5 and 40×40 / 10×10). Finally, as technical novelty, *iv)* we equipped the model with priors and we modified the learning algorithm presented in [15] to achieve faster convergence. This has proven to be useful to analyze large datasets.

The rest of the paper is organized as following; in Section II we introduce counting grids and we derive two

[1] in terms of counting grid and window size

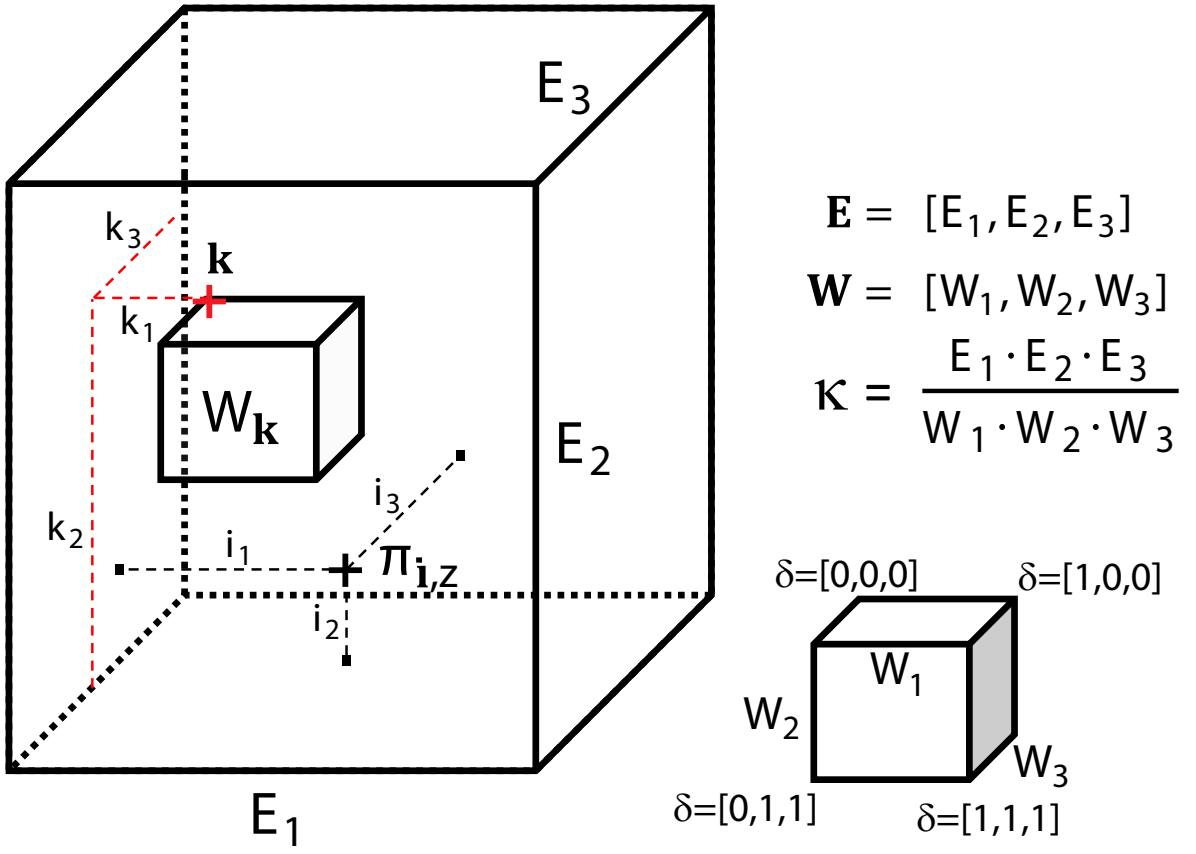

Fig. 2. An example of a counting grid geometry. In general, the data is embedded into a hypercube which is wrapped around along each dimension to avoid local minima that would be caused by abrupt cuts along any dimension.

algorithms to learn them. Then, in Section III, we present three classification strategies based on different learning procedure and different use of the estimated CG. An extensive experimental section is introduced in Section IV which presents applications in computer vision, information retrieval, computational biology and medicine. Moreover it discusses the sensitivity of the classification framework to dimensionality and complexity of the model and refuses some of the hypothesis of [15]. Finally, conclusions are drawn in Section V.

II. Epitomes for bags of features: The counting grid model

Formally, the basic counting grid $\pi_{\mathbf{i},z}$ is a set of normalized counts of words/features indexed by z on the D-dimensional discrete grid indexed by $\mathbf{i} = (i_1, \ldots, i_D)$ where each $i_d \in [1 \ldots E_d]$ and $\mathbf{E} = (E_1, \ldots, E_D)$ 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.

A given bag of words/features, represented by counts $\{c_z\}$ is assumed to follow a count distribution found in a window somewhere in the counting grid. In particular, using windows of dimensions $\mathbf{W} = [W_1, \ldots, W_D]$,

each bag can be generated by first averaging all counts in the hypercube window $W_{\mathbf{k}} = [\mathbf{k} \dots \mathbf{k} + \mathbf{W}]$ starting at D-dimensional grid location $\mathbf{k}$ and extending in each direction d by W_d grid positions to form the histogram $h_{\mathbf{k},z} = \frac{1}{\prod_d W_d} \sum_{\mathbf{i} \in W_{\mathbf{k}}} \pi_{\mathbf{i},z}$, and then generating the bag of features from such averaged histogram. 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

$$p(\{c_z\}|\mathbf{k}) = \prod_z (h_{\mathbf{k},z})^{c_z} = \frac{1}{\prod_d W_d} \prod_z \big(\sum_{\mathbf{i} \in W_{\mathbf{k}}} \pi_{\mathbf{i},z} \big)^{c_z}, \tag{1}$$

Relaxing the terminology, we will refer to $\mathbf{E}$ and $\mathbf{W}$ respectively as the counting grid size and the window size, indicating with $W_{\mathbf{k}}$ the particular window placed at location $\mathbf{k}$. We will refer to the ratio of the window volumes, κ, as a capacity of the model in terms of an *equivalent number of topics*, as this is how many nonoverlapping windows can be fit onto the grid. We will see in the experimental section how the model is sensitive to this parameter more than the individual choices of $\mathbf{W}$ and $\mathbf{E}$. Fine variation achievable by moving the windows in between any two close by but nonoverlapping windows is useful if we expect such smooth changes to occur in the data, and we illustrate in our experiments that indeed it does.

A. Inference and learning

To compute the log likelihood of the data, $\log P$, we need to sum over the latent variables $\mathbf{k}$ before computing the logarithm, which, as in mixture models, or as in epitomes [21] – much more similar to the counting grids –, makes it difficult to perform assignment of the latent variables (in our case positions in the counting grid) while also estimating the model parameters. This makes an iterative exact or a variational EM algorithm necessary. Bounding (variationally) the non-constant part of $\log P$, we get

$$\begin{aligned} \log P \geq B = \quad & - \quad \sum_t \sum_{\mathbf{k}} q^t_{\mathbf{k}} \log q^t_{\mathbf{k}} + \\ & + \quad \sum_t \sum_{\mathbf{k}} q^t_{\mathbf{k}} \sum_z c^t_z \log \sum_{\mathbf{i} \in W_{\mathbf{k}}} \pi_{\mathbf{i},z} + \\ & + \quad \sum_t \sum_{\mathbf{k}} q^t_{\mathbf{k}} \log p_{\mathbf{k}}, \end{aligned} \tag{2}$$

where $q^t_{\mathbf{k}}$ is the variational distribution over the latent mapping onto the counting grid of the t-th bag and $p_{\mathbf{k}}$ is the overall prior on the location.

Each of these variational distributions can be varied to maximize the bound. In fact, for a given counting grid π, the bound is maximized when each distribution q^t is equal to the exact posterior distribution. This is a standard variational derivation of the exact E step, which leads to

$$q^t_{\mathbf{k}} \propto p_{\mathbf{k}} \cdot \exp \sum_z c^t_z \cdot \log h_{\mathbf{k},z}, \tag{3}$$

which simply establishes that the choice of $\mathbf{k}$ should minimize the KL divergence between the counts in the bag and the counts $h_{\mathbf{k},z} = \sum \pi_{\mathbf{i},z}$ in the appropriate window $W_{\mathbf{k}}$ in the counting grid. Each q^t, is normalized over all

possible locations.

To optimize the bound B with respect to parameters we note first that only the second term in Eq. 3 involves these parameters, and that it requires another summation before applying the logarithm. The summation is over the grid positions $\mathbf{i}$ within the window $W_{\mathbf{k}}$, which we can again bound using a variational distribution and the Jensen's inequality:

$$\log\sum_{\mathbf{i}\in W_{\mathbf{k}^t}}\pi_{\mathbf{i},z} = \log\sum_{\mathbf{i}\in W_{\mathbf{k}^t}} r^t_{\mathbf{i},\mathbf{k},z}\cdot\frac{\pi_{\mathbf{i},z}}{r^t_{\mathbf{i},\mathbf{k},z}} \geq \sum_{\mathbf{i}\in W_{\mathbf{k}^t}} r^t_{\mathbf{i},\mathbf{k},z}\log\frac{\pi_{\mathbf{i},z}}{r^t_{\mathbf{i},\mathbf{k},z}} \tag{4}$$

where $r^t_{\mathbf{i},\mathbf{k},z}$ is a distribution over locations $\mathbf{i}$. The distribution r is positive and $\sum_{\mathbf{i}\in W_{\mathbf{k}}} r^t_{\mathbf{i},\mathbf{k},z} = 1$: it is indexed by $\mathbf{k}$, as the normalization is done differently in each window, by z, as it can be different for different features, and by t, as the term is inside the summation over t, so a different distribution r could be needed for different bags $\{c^t_z\}$. This distribution could be thought of as information about what proportion of these c_z features of type z was contributed by each of the different sources $\pi_{\mathbf{i},z}$ in the window $W_{\mathbf{k}}$. However, by performing constrained optimization (so that r adds up to one), we find that, assuming a fixed set of parameters π, the distribution $r^t_{\mathbf{i},\mathbf{k},z}$ that maximizes the bound is independent of t, i.e., the same for each bag:

$$r^t_{\mathbf{i},\mathbf{k},z} = \frac{\pi_{\mathbf{i},z}}{\sum_{\mathbf{i}\in W_{\mathbf{k}}}\pi_{\mathbf{i},z}} = \frac{\pi_{\mathbf{i},z}}{\prod_d W_d\cdot h_{\mathbf{k},z}}. \tag{5}$$

If we do consider distributions r as a feature mapping to the counting grid, then this result is again intuitive. If all we know is that a bag containing c_z features of type z is mapped to the grid section $W_{\mathbf{k}}$, and have no additional information about what proportions of these c_z features were contributed from different incremental counts $\pi_{\mathbf{i},z}$, then the best guess is that these proportions follow the proportions among $\pi_{\mathbf{i},z}$ inside the window.

If we assume now that r and q distributions are fixed, then combining Eqs. 3,4 we obtain the following bound

$$B_\pi = \sum_t\sum_{\mathbf{k}|\mathbf{i}\in W_{\mathbf{k}}} q^t_{\mathbf{k}}\sum_z c^t_z\cdot r_{\mathbf{i},\mathbf{k},z}\log\frac{\pi_{\mathbf{i},z}}{r_{\mathbf{i},\mathbf{k},z}} \tag{6}$$

The optimization of B_π wrt parameters $\pi_{\mathbf{i},z}$ under the normalization constraint over features z, results in

$$\frac{\partial B_\pi}{\partial\pi_{\mathbf{i},z}} = \sum_t\sum_{\mathbf{k}|\mathbf{i}\in W_{\mathbf{k}}} q^t_{\mathbf{k}}\cdot c^t_z\cdot r_{\mathbf{i},\mathbf{k},z}\cdot\frac{1}{\pi_{\mathbf{i},z}} - \lambda \tag{7}$$

where λ is the Lagrange multiplier to ensure the normalization.

After setting the derivatives to 0 and solving for $\pi_{\mathbf{i},z}$ we get

$$\begin{aligned}
\hat{\pi}_{\mathbf{i},z} &\propto \sum_t\sum_{\mathbf{k}|\mathbf{i}\in W_{\mathbf{k}}} q^t_{\mathbf{k}}\cdot c^t_z\cdot r^t_{\mathbf{i},\mathbf{k},z} \\
&\propto \sum_t\sum_{\mathbf{k}|\mathbf{i}\in W_{\mathbf{k}}} q^t_{\mathbf{k}}\cdot c^t_z\cdot r^t_{\mathbf{i},\mathbf{k},z} \\
&\propto \sum_t\sum_{\mathbf{k}|\mathbf{i}\in W_{\mathbf{k}}} q^t_{\mathbf{k}}\cdot c^t_z\cdot\frac{\pi_{\mathbf{i},z}}{\sum_{\mathbf{i}}\pi_{\mathbf{i},z}} \\
&\propto \pi_{\mathbf{i},z}\cdot\Big(\sum_t c^t_z\sum_{\mathbf{k}|\mathbf{i}\in W_{\mathbf{k}}}\frac{q^t_{\mathbf{k}}}{h_{\mathbf{k},z}}\Big)
\end{aligned} \tag{8}$$

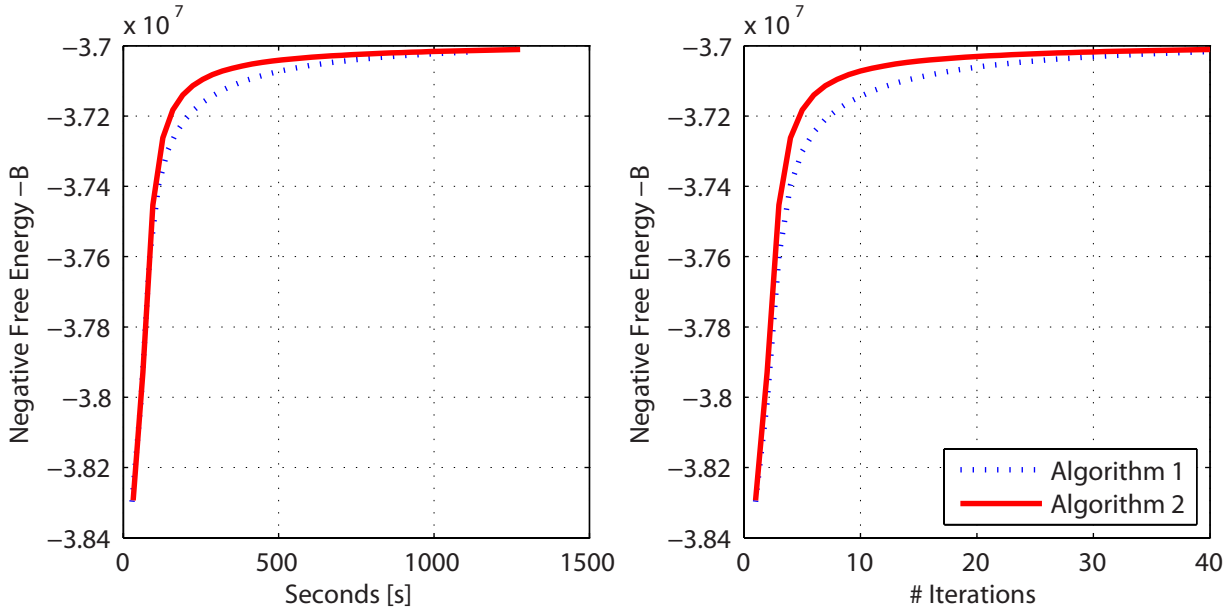

Fig. 3. Time-vs-B (Eq. 3) and Iteration-vs-B. for Algorithm 1 and 2 on a large dataset composed by 45K recipes and crawled from the website www.allrecipes.com. Red an blue curves are significantly different (p-value ≤ 0.05). We also held out 5K documents and compute B; test free energies did not differ.

$$\propto \quad \pi_{\mathbf{i},z} \cdot \underbrace{\left(\sum_t c_z^t \sum_{\mathbf{k}|\mathbf{i}\in W_\mathbf{k}} q_\mathbf{k}^t \right)}_{f_{\mathbf{i},z}} \cdot \sum_{\mathbf{k}|\mathbf{i}\in W_\mathbf{k}} \frac{1}{h_{\mathbf{k},z}} \tag{9}$$

Eq. 8 it's the multiplicative update for π given in [15]. Being multiplicative it is characterized by a slow convergence and in particular the placement of the feature tokens in the window is not optimal.

To fasten the convergence we rewrite Eq. (8) into Eq. (9), to highlight the contribution of the data, computed in the E-Step. This allows us to iterate the updates for π and h, in the M-step, to reach a faster convergence and the "optimal" placement of the features. Please note that despite a loop is introduced in the M-step, the biggest computational burden is the E-step which dominates the complexity of the algorithm. Looping between π and h makes the free energy decrease faster as illustrated in Fig. 3.

At last, the prior over the locations can be updated as follows:

$$p_\mathbf{k} \quad \propto \quad \sum_t q_\mathbf{k}^t \tag{10}$$

Summarizing, in Algorithm 1, we report the free energy minimization procedure presented in [15]. Because of the multiplicative update for π it is more suitable for small-to-medium datasets or for small choice of the window $\mathbf{W}$. On the other hand, Algorithm 2 yields to a faster convergence (without overtraining) and it is more suitable for larger datasets and windows.

As is often the case the two approaches work best if used in conjunction. Both algorithms are iterated till convergence

Algorithm 1: EM-Algorithm to learn a counting grid.

Input: Bag of features, c_z^t for each sample, counting grid size $\mathbf{E}$, window size $\mathbf{W}$

while *Convergence* **do**

% E-Step ;

foreach *Sample* $t = 1 \dots T$ **do**

1. Update $q_{\mathbf{k}}^t \propto \exp\left\{ \sum_z c_z^t \log h_{\mathbf{k},z} \right\}$,;

% M-Step ;

2. Update $\pi_{\mathbf{i},z} \propto \pi_{\mathbf{i},z}^{old} \cdot \sum_t c_z^t \sum_{\mathbf{k}|\mathbf{i}\in W_{\mathbf{k}}} \frac{q_{\mathbf{k}}^t}{h_{\mathbf{k},z}}$;
3. Compute $h_{\mathbf{k},z} = \frac{1}{\prod_d W_d} \sum_{\mathbf{i}\in W_{\mathbf{k}}} \pi_{\mathbf{i},z}$;
4. Update $p_{\mathbf{k}} \propto \sum_t q_{\mathbf{k}}^t$;
5. Compute the Log-Likelihood B with Eq. 3 ;
6. Check for convergence, e.g. $|B - B^{old}| \leq \tau$;

7. Return $\pi_{\mathbf{i},z}$, $p_{\mathbf{k}}$ and $\{q_{\mathbf{k}}^t\}_t$;

Algorithm 2: EM-Algorithm to learn a counting grid on big datasets.

Input: Bag of features, c_z^t for each sample, counting grid size $\mathbf{E}$, window size $\mathbf{W}$

while *Convergence* **do**

% E-Step ;

1. $f_{\mathbf{k},z} = 0$;

foreach *Sample* $t = 1 \dots T$ **do**

$2a$. Update $q_{\mathbf{k}}^t \propto \exp\left\{ \sum_z c_z^t \log h_{\mathbf{k},z} \right\}$, ;

$2b$. $f_{\mathbf{i},z} = f_{\mathbf{i},z} + c_z^t \cdot \sum_{\mathbf{k}|\mathbf{i}\in W_{\mathbf{k}}} q_{\mathbf{k}}^t$;

% M-Step ;

for $i_M = 1 \dots M$ **do**

$3a$. Update $\pi_{\mathbf{i},z} \propto \pi_{\mathbf{i},z}^{old} \cdot f_{\mathbf{i},z} \cdot \sum_{\mathbf{k}|\mathbf{i}\in W_{\mathbf{k}}} \frac{1}{h_{\mathbf{k},z}}$;

$3b$. Compute $h_{\mathbf{k},z} = \frac{1}{\prod_d W_d} \sum_{\mathbf{i}\in W_{\mathbf{k}}} \pi_{\mathbf{i},z}$;

4. Update $p_{\mathbf{k}} \propto \sum_t q_{\mathbf{k}}^t$;
5. Compute the Log-Likelihood B with Eq. 3 ;
6. Check for convergence, e.g. $|B - B^{old}| \leq \tau$;

7. Return $\pi_{\mathbf{i},z}$, $p_{\mathbf{k}}$ and $\{q_{\mathbf{k}}^t\}_t$;

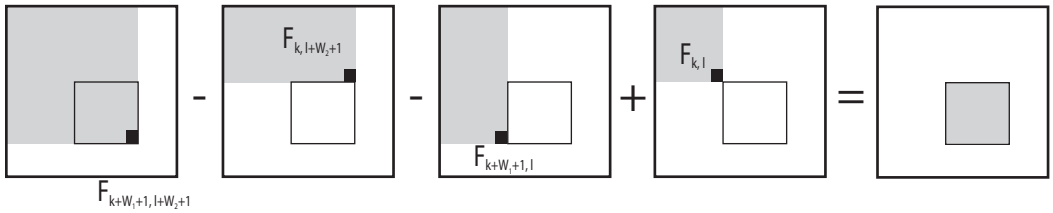

Fig. 4. Cumulative sums allows to efficiently compute the sum of the elements in an arbitrary window inside the grid.

(within a desired precision). In both cases the E-step aligns all bags of features to grid windows that (re)match the bags' histograms, and the second re-estimates the counting grid so that these same histogram matches are even better. Thus, starting with non-informative (but symmetry breaking) initialization, this iterative process will jointly estimate the counting grid and align all bags to it. To avoid severe local minima, it is important, however, to consider the counting grid as a multidimensional torus, and consider all windowing operations accordingly, as was previously proposed for learning epitomes [21], a model that quilts spatially-organized images and videos. This prevents the problems with grid boundaries which otherwise could not be crossed when more space is needed to grow the layout of the features.

B. Computational efficiency

Careful examination of the steps reveals that by the efficient use of cumulative sums, both the E and M steps are linear in the size of the counting grid. Both steps require computing $\sum_{\mathbf{i}\in W_{\mathbf{k}}} f_{\mathbf{i}}$, which can be done by first computing, in linear time the cumulative sums of f and then computing appropriate linear combinations.

For example, in the 2D case we have $\mathbf{i} = (i, j)$, $\mathbf{k} = (k, \ell)$ and one can compute the cumulative sum $F_{m,n} = \sum_{(i,j)\leq(m,n)} f_{i,j}$, and then set $\sum_{(i,j)\in W_{k,\ell}} f_{i,j} = F_{k+W_1+1,\ell+W_2+1} - F_{k,\ell+W_2+1} - F_{k+W_1+1,\ell} + F_{k,\ell}$. This is depicted in Fig. 4. This can be generalized by associating to each vertex v of the hypercube $W_{\mathbf{k}}$ (2^D vertexes in total) a binary vector δ^v. Different vertices's of $W_{\mathbf{k}}$ share various coordinates, as along a dimension, say d, a vertex v can only assume two values (k_d or $k_d + W_d$). We define elements of vector δ_v as follows

$$\delta_d^v = \begin{cases} 1 & \text{if } v_d = k_d + W_d \\ 0 & \text{else} \end{cases}$$

The value of δ for some vertex is shown in Fig.2.

Given this we can write

$$\sum_{\mathbf{i}\in W_{\mathbf{k}}} f_{\mathbf{i}} = \sum_{v=1}^{2^D} (-1)^{|1-\delta^v|} \cdot F_{\mathbf{i}+\delta^v \circ \mathbf{W}} \tag{11}$$

where the $\circ$ is the point-wise multiplication.

III. CLASSIFICATION STRATEGIES

The Counting Grid model can be employed in different ways to perform classification or clustering. Here we investigated three possibilities: *i)* the Nearest Neighbor rule in the Counting Grid space [15], *ii)* an hybrid generative/discriminative classification scheme [23], and *iii)* generative classification via free energy comparisons. In the following the three different procedures are summarized.

A. Nearest Neighbor in the Counting Grid space

Following the original recipe of [15], a single CG is learned using all samples (but ignoring their labels). Data samples are embedded into the CG space and their position in the grid is used to evaluate how well they cluster on the grid. More in detail, we assign to each bag in the test fold the label of the nearest training bag. To compute the distance we used the Euclidean distance on a Torus,

$$D_{1,2} = \sqrt[D]{\sum_d \left(\min \left\{ |i_d^1 - i_d^2|, E_d - |i_d^1 - i_d^2| \right\} \right)^2} \tag{12}$$

where $\mathbf{i}^t = (i_1^t, i_2^t, \dots i_D^t)$ is a generic mapping locations. Learning a model only using the training samples, performing inference on the test fold and then using the Nearest Neighbor rule, did not change the results. The accuracy obtained with NN is very related to the concept of clustering purity and useful to evaluate the counting grid as an embedding method.

Finally, it is worth to note that the strategy of embedding the label as a continue value (see Eq. 4 in [15]) was found to be (empirically) equivalent to the NN.

B. Hybrid generative/discriminative classifier

Following the standard hybrid generative-discriminative recipe [25], [26], we learned a single model using all the samples and then we exploited the labels to train a discriminative classifier on by-products of the generative model.

Here we considered the same approach introduced in [23] where a kernel that exploited the geometric reasoning of the counting grid model was proposed.

In fact, by construction each point in the grid depends by its neighborhood, defined by $\mathbf{W}$ and the idea of [23] was to spread around a neighbourhood region defined by $W_{\mathbf{k}}$ by convolving $q_{\mathbf{k}}^t$ with a binary mask $M_{\mathbf{k}}$ of size $\mathbf{W}$. Actually, by construction, a sample is mapped in $\mathbf{k}$ if agrees with the feature distribution in the subwindow $\mathbf{W}$. More formally, given two sample t and u, the so called Spreading Similarity Measure is defined by [23]:

$$g^t = q_{\mathbf{k}}^t * M_{\mathbf{W}} \qquad SSM_f(t,u) = f(g^t, g^u), \tag{13}$$

where $f(\cdot,\cdot)$ is any similarity measure. In our experiments we considered histogram intersection [27] which performed well in [23].

C. Generative classifier

We evaluated for the first time the counting grid as generative classifier. We learned a generative model per class using the samples in training folds. Then, for each test sample t and for each class model c a three steps procedure is employed: i) we inferred the position in the grid $\mathbf{k}_c^t$ by performing the E-step of the learning algorithm (Line 2a in 2), ii) we computed its free energy B_c using Eq. 3, and iii) we assigned it the label of the class which yielded to the lower value (maximum likelihood classification). We used a uniform prior on the classes.

IV. Experiments

An exhaustive experimental section is proposed in order to evaluate the Counting Grid models in various application scenarios. In particular, we investigated the classification performances of the three classification schemes in 4 domains by varying i) dimensionality, ii) window shape, and iii) window size of the Counting Grid. Finally we also compared Counting Grids with the state-of-the-art.

A. Details

In all the tests we employed 10-folds crossevaluation, repeating each test over 3 random fold partitions and averaging the results. As learning algorithm we used 2, we set $M = 3$ and the convergence threshold $\tau = 1e - 5$. In all the tests, convergence is achieved after $\sim$70 iterations of the EM-algorithm.

B. Datasets considered

We considered 4 datasets from different domains: Natural Language processing, Computer Vision, Bioinformatics and Medicine.

Text: In the CMU newsgroup dataset [28] each news post is treated as a document (a bag of words) labeled by one of 20 labels representing the news group of its origin. Following previous work [15], [24], [29], [30] we reduced the dataset into subsets with varying similarities among the news groups. We consider and report here the results only for the more challenging, **news-20-same**, with 1700 posts from the highly related groups `comp.os.ms-windows`, `comp.windows.x` and `comp.graphics`.

Images: The visual scene dataset, introduced in [31], is composed by two datasets composed by four natural and four artificial (man-made) categories and it is widely used by the vision community. Each class contains roughly 250 images.

Following the standard bag-of-*visual*-words [12] approach from each image, we extracted SIFT features from 16x16 pixel windows computed over a grid spaced of 8 pixels and clustered the descriptors in $Z = 200$ visual words. We describe an image as a bag of their features.

TABLE I

COMPLEXITIES USED IN THE EXPERIMENT 1. WE TRIED TO KEEP THE WINDOWS' VOLUMES OF THE VARIOUS DIMENSIONS AS SIMILAR AS POSSIBLE.

Dimension	W	Counting Grid sizes considered
5	[2 2 2 2 2]	[3 3 3 3 3], [4 4 4 4 4], [5 5 5 5 5] [6 6 6 6 6], [7 7 7 7 7], [8 8 8 8 8]
4	[2 2 2 2]	[3 3 3 3], [4 4 4 4], [5 5 5 5], [6 6 6 6] [7 7 7 7], [8 8 8 8], [9 9 9 9] [10 10 10 10], [12 12 12 12]
3	[3 3 3]	[4 4 4], [5 5 5], [6 6 6], [7 7 7], [8 8 8] [10 10 10], [12 12 12], [15 15 15] [20 20 20], [25 25 25], [30 30 30]
2	[5 5]	[7 7], [9 9], [10 10], [12 12], [15 15] [20 20], [25 25], [30 30], [40 40], [50 50] [60 60], [70 70], [80 80], [90 90] [100 100], [120 120]
1	[5]	[7], [10], [12], [15], [20], [25], [35] [45], [60], [80], [100], [140], [180], [220] [260], [300], [340]

Microarray Expressions: Previous work [1], [2], [32] has interpreted microarray expression values as counts in "bags-of-genes", and good classification rates have been reached. The starting point is a microarray gene expression matrix, where the element at position (i, j) represents the expression level of the $i-th$ gene in the $j-th$ subject/sample. In our experiments we considered the Colon dataset by Alon [33] consisting of 62 samples with 7984 features. As preprocessing step, we filtered the genes by variance and keep only the most variable two thousand.

Brain MRI: The study population used in this work consists of 42 patients who were being treated for schizophrenia and 40 controls. The original MRI image size is 384x512x144. A Regions of Interest (ROIs) approach was adopted in order to focus the analysis on well defined brain sub-parts [34]. In this work, we used left Thalamus which is found to be impaired in schizophrenic patients. DARTEL [35] tools within SPM software [36] was used to pre-process the data in order to align properly the subjects onto the canonical space and normalize the MRI intensity according to a well defined medical protocols [36]. Finally, we computed the histogram of normalized intensities of Thalamus for every subject. Number of bins in each histogram is chosen to be 40 which showed the best performance in our experiments.

C. Experiment 1: Evaluation of the effect of the dimensionality

In this first test we evaluated the effect of the different dimensionality of the Counting Grids models. We considered Counting Grids of dimensions 1 to 5 with complexities reported in Tab.I. Results are shown in Fig. 5 where we show the classification accuracy on the y-axis and the model capacity κ on the x-axis. As described in

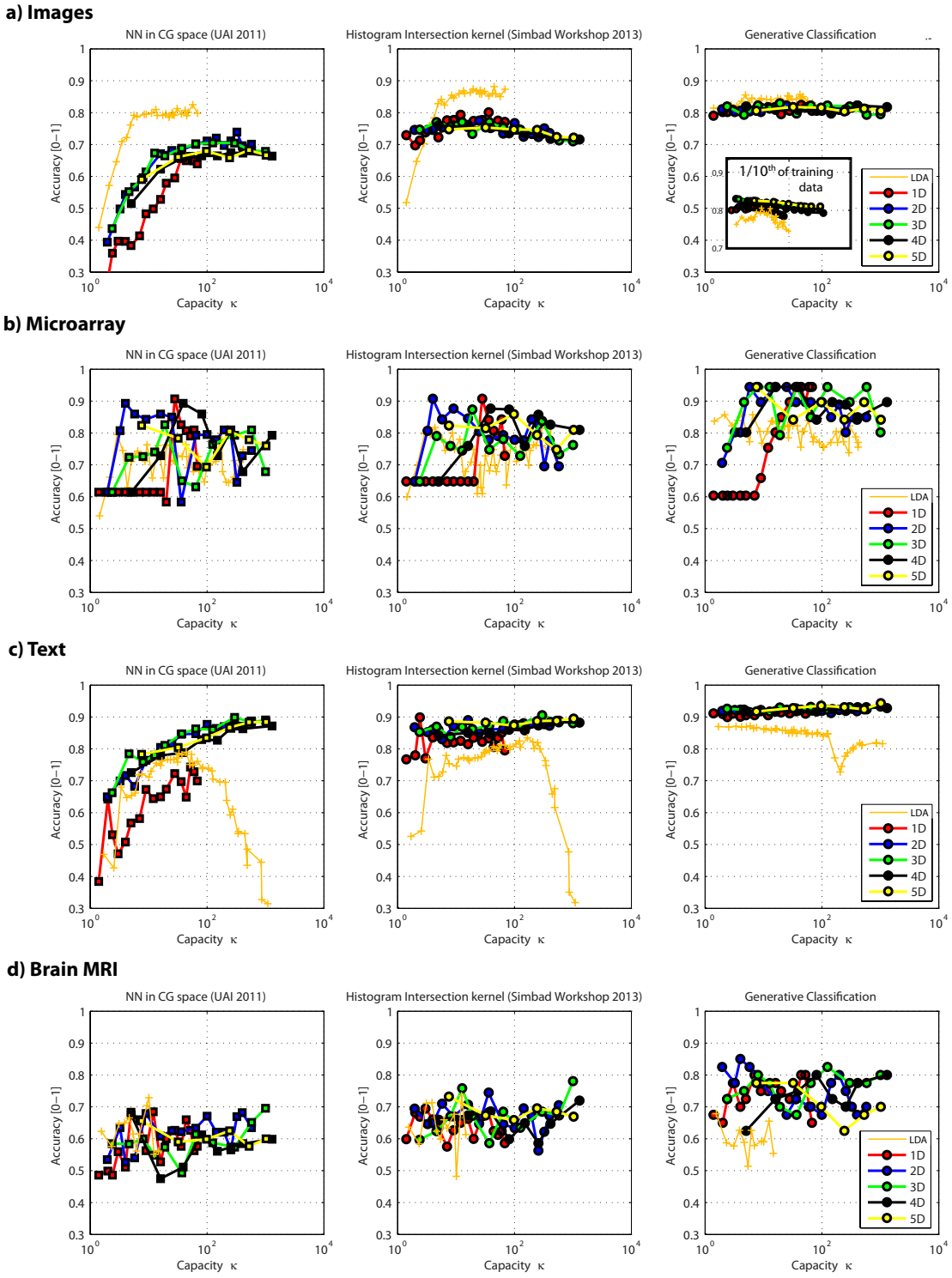

Fig. 5. a) Image classification results. b) Microarray classification results. c) Text classification results. d) MRI Classification results.

the previous sections, the capacity is defined as the ratio between counting grid size and window size; different complexities can lead to same capacity (e.g., [20 20] / [5 5] and [40 40] / [10 10]).

The generative classifier reaches the best results in all the cases. Text and Images datasets, characterized by a large number of sample (no overtraining) behave similarly: accuracies does not seem depend on the dimensionality. For what concern the unsupervised scenario (NN is used to evaluate counting grids unsupervised embedding), the accuracy (purity) grows with κ as some space is needed to cluster the samples.

Finally in all the cases, results are quite regular across complexities.

On the other hand, the life science datasets, exhibits irregular behavior even if, especially for the maximum likelihood classifier, some areas where performances are higher (low capacities κ) can be noticed. This is clearly due to overtraining, in fact when $\kappa > T$, the grid can fit an independent window for each sample. Indeed, samples overlap disappear and all classification strategies have little sense.

Fig. 5 also reports the comparison with Latent Dirichlet Allocation [11].

We evaluated LDA in a very similar way as Counting Grids; in fact, i) Akin to Sec. III-A, LDA embedding (topic simplex) is evaluated using the KL-divergence on the documents' topic proportions, ii) Akin to Sec. III-B we built an hybrid classifier by learning a linear SVM on the topic proportions and finally iii) as in Sec. III-C, we compared loglikelihoods under different models to evaluate the maximum likelihood approach.

As visible Counting Grids outperform LDA in nearly all the tests. Surprisingly LDA performs better in the vision dataset where Counting Grids would be expected to excel. This is mostly due to the amount of training data used: If we put in more standard Computer vision conditions by reducing the training images and using, as training set, only the images of the current fold (e.g., we use $1/10^{th}$ of the data), the gap between the two methods vanishes (See plate in Fig. 5a - Generative classification). This is also consistent with the findings of [15].

D. Experiment 2: Evaluation of the effect of unbalanced windows

For this second test we only considered 3-D Counting Grids, and microarray and computer vision datasets. We tested if rectangular windows, when the size of the window dimensions W_d differs, influence the accuracy. We considered the complexities reported in Tab. II.

TABLE II
COMPLEXITIES USED IN THE EXPERIMENT 2

Dimension	W	Counting Grid sizes considered
3	[4 4 4]	[18 18 18], [20 20 20], [25 25 25], [30 30 30]
3	[4 2 8]	[18 18 18], [20 20 20], [25 25 25], [30 30 30]
3	[4 8 2]	[18 18 18], [20 20 20], [25 25 25], [30 30 30]
3	[2 2 16]	[18 18 18], [20 20 20], [25 25 25], [30 30 30]
3	[16 2 2]	[18 18 18], [20 20 20], [25 25 25], [30 30 30]

Results are shown in Fig. 6. It seems evident, from this figure, that maximum likelihood classifier is not affected by

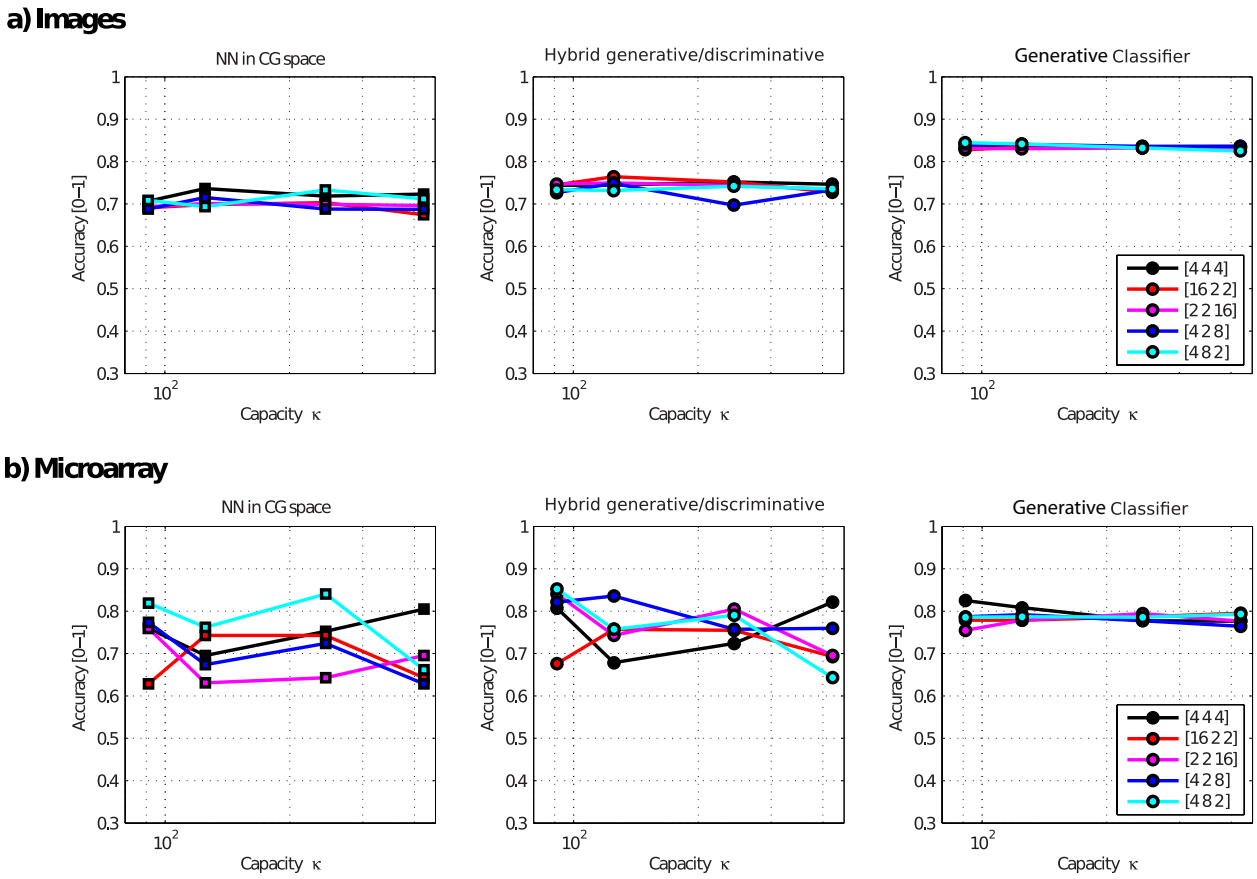

Fig. 6. a) Image classification results. b) Microarray classification results

rectangular windows; the other two methods, being based on geometrical reasoning, present some small fluctuations in the results.

E. Experiment 3: Evaluation of the effect of windows of different sizes

For this last test we only considered 2-D Counting Grids, and microarray and computer vision datasets and we tested how the window size affects the classification accuracy. We considered the complexities in Tab. III.

Results are shown in Fig. 7; as evident the window size $\mathbf{W}$ is not a critic parameter. Model of same capacity κ tend to behave similarly, once the window size is "sufficiently big" [2] and the model has sufficient "overlapping" power. For larger windows, local minima or slow convergence may occur as the algorithm must figure out how to arrange the features in a larger space. In this case one may want to use a larger (e.g., 5) number for M-step iterations. This is also confirmed by other counting grids literature [30], [37].

[2]for example bigger than $[2, 2]$ in the 2-D case

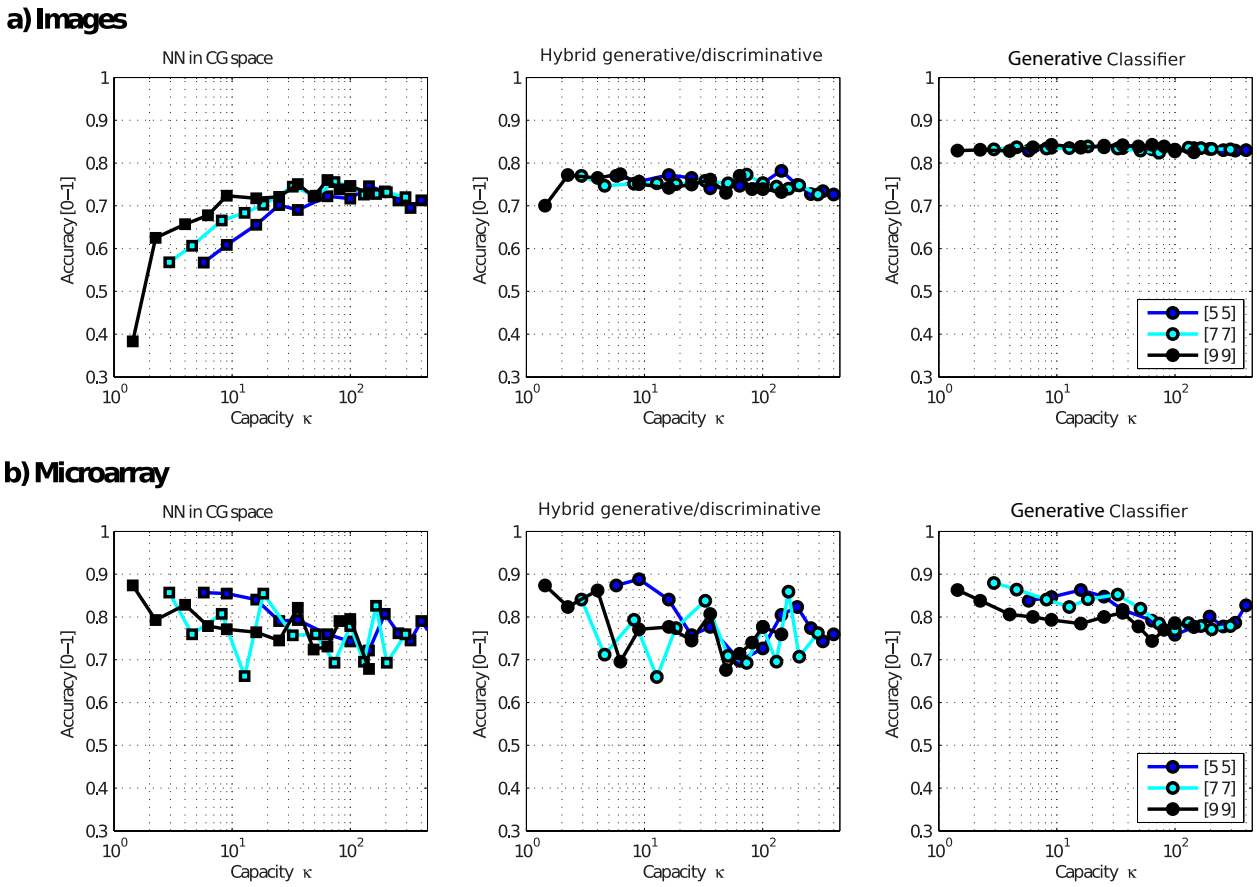

Fig. 7. a) Image classification results. b) Microarray classification results

TABLE III
COMPLEXITIES USED IN THE EXPERIMENT 3

Dimension	W	Counting Grid sizes considered
2	[5 5]	[12 12], [15 15], [20 20], [25 25], [30 30] [40 40], [50 50], [60 60], [70 70], [80 80] [90 90], [100 100], [120 120]
2	[7 7]	[12 12], [15 15], [20 20], [25 25], [30 30] [40 40], [50 50], [60 60], [70 70], [80 80] [90 90], [100 100], [120 120]
2	[10 10]	[12 12], [15 15], [20 20], [25 25], [30 30] [40 40], [50 50], [60 60], [70 70], [80 80] [90 90], [100 100], [120 120]

F. Comparison with the state of the art

The previous sections were meant to analyze the performance of counting grids in different settings and with different classification strategies. Despite counting grids were already proven to be superior to topic model in the

TABLE IV
COMPARISON WITH THE STATE OF THE ART. TO COMPARE ON MICROARRAYS WE USED LEAVE-ONE-OUT, AS IN [38].

Dataset	CG	LDA	Other - SoA
Text	**92,5**%	82,6 %	78.1% [24]
Images	81,3%	81,9%	**94,0**% [25]
Microarray	**93,9**%	91,8%	93,5% [38]
Brain MRI	**87,6**%	73,1%	-

conference versions of this paper [15], [23], for the sake of completeness we report in Tab. IV some comparisons with the state of the art on the dataset considered[3] We only considered squared 2-D counting grids (Sec.IV-C), the generative classifier (proposed here) and we chose the complexity via cross-validation on a subset of the complexities of Tab. I.

To the best of our knowledge the results obtained on microarrays and text set a new state of the art. Results on the image dataset are far from the state of the art but, they can be improved keeping some spatial information [16], [37] and sophisticated generative kernels [25] can be built upon them.

Finally as visible by Fig. 5-7, the generative classification strategy outperformed [15] and [23].

V. CONCLUSIONS

This paper revised the Counting Grid models by proposing a new learning algorithm, by deeply exploiting the effect of CG-dimensional variations, and by comparing several classification strategies. Counting grids outperformed subspace models in evaluating a variety of datasets where we found that thematic shifts seem to be a better fit to capturing correlations in word occurrence. In particular, we shown that classification by CG reached the state of the art on Microarrays and Text set. Moreover, a clear improvement was observed when the CG is trained in a supervised fashion by simply adopting a generative classification approach.

Quite surprisingly the model has found to be sensitive only to the so called capacity which can be easily estimated either manually or via crossevaluation. Indeed, CG performed quite stable in varying such parameters in all the experiments by showing the robustness of CG in avoiding the risk of overtraining when complexity of the model increased or the number of training sample was reduced like the cases of Microarray and Brain MRI datasets.

REFERENCES

[1] S. Rogers, M. Girolami, C. Campbell, and R. Breitling, "The latent process decomposition of cDNA microarray datasets," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 2, no. 2, pp. 143–156, 2005.

[2] M. Bicego, P. Lovato, A. Perina, M. Fasoli, M. Delledonne, M. Pezzotti, A. Polverari, and V. Murino, "Investigating topic models' capabilities in expression microarray data classification," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 9, no. 6, pp. 1831–1836, 2012.

[3]MRI is a private dataset.

[3] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *In Workshop on Statistical Learning in Computer Vision, ECCV*, 2004, pp. 1–22.

[4] R. Toldo and A. F. U. Castellani, "The bag of words approach for retrieval and categorization of 3d objects," *The Visual Computer*, vol. 26, no. 10, pp. 1257–1268, 2010.

[5] G. Brelstaff, M. Bicego, N. Culeddu, and M. Chessa, "Bag of peaks: interpretation of NMR spectrometry," *Bioinformatics*, vol. 25, pp. 258–264, 2009.

[6] U. Castellani, E. Rossato, V. Murino, M. Bellani, G. Rambaldelli, C. Perlini, L. Tomelleri, M. Tansella, and P. Brambilla, "Classification of schizophrenia using feature-based morphometry," *Journal of Neural Transmission*, vol. 119, pp. 395–404, 2012.

[7] A. Perina, P. Lovato, and N. Jojic, "Bags of words models of epitope sets: Hiv viral load regression with counting grids," in *Pacific Symposium on Biocomputing (PSB)*, 2014.

[8] M. Yamamoto and K. Sadamitsu, "Dirichlet mixtures in text modeling," University of Tsukuba, Tech. Rep. CS-TR-05-1, 2005.

[9] S. T. Dumais, "Latent semantic analysis." *ARIST*, vol. 38, 2004.

[10] T. Hofmann, "Probabilistic latent semantic indexing," in *ACM conference on Research and development in information retrieval (SIGIR)*, 1999, pp. 50–57.

[11] D. Blei, A. Ng, and M. Jordan, "Latent dirichlet allocation," *Journal of machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[12] L. Fei-Fei and P. Perona, "A bayesian hierarchical model for learning natural scene categories." in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005, pp. 524–531.

[13] B. T. Inbal, N. Yuval, and K. Rachel, "FragBag, an accurate representation of protein structure, retrieves structural neighbors from the entire PDB quickly and accurately." *Proceedings of the National Academy of Sciences of the United States of America*, vol. 107, no. 8, pp. 3481–6, Feb. 2010.

[14] U. Castellani, A. Perina, V. Murino, M. Bellani, G. Rambaldelli, M. Tansella, and P. Brambilla, "Brain morphometry by probabilistic latent semantic analysis," in *MICCAI (2)*, 2010, pp. 177–184.

[15] N. Jojic and A. Perina, "Multidimensional counting grids: Inferring word order from disordered bags of words," in *Proceedings of conference on Uncertainty in artificial intelligence (UAI)*, 2011, pp. 547–556.

[16] A. Perina and N. Jojic, "Image analysis by counting on a grid," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011, pp. 1985–1992.

[17] P. Lovato, M. Bicego, M. Cristani, N. Jojic, and A. Perina, "Feature selection using counting grids: Application to microarray data," in *SSPR/SPR*, 2012, pp. 629–637.

[18] A. Globerson, G. Chechik, F. Pereira, and N. Tishby, "Euclidean embedding of co-occurrence data," *Journal of Machine Learning Research*, vol. 8, pp. 2265–2295, 2007.

[19] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *SCIENCE*, vol. 290, pp. 2323–2326, 2000.

[20] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A Global Geometric Framework for Nonlinear Dimensionality Reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, Dec. 2000.

[21] N. Jojic, B. J. Frey, and A. Kannan, "Epitomic analysis of appearance and shape." in *International Conference on Computer Vision (ICCV)*, 2003, pp. 34–43.

[22] N. Jojic, V. Jojic, B. Frey, C. Meek, and D. Heckerman, "Using "epitomes" to model genetic diversity: Rational design of hiv vaccine cocktails," in *Neural Information Processing Systems (NIPS)*, 2006, pp. 587–594.

[23] A. Perina, M. Bicego, U. Castellani, and V. Murino, "Exploiting geometry in counting grids," in *SIMBAD*, 2013, pp. 250–264.

[24] J. Reisinger, A. Waters, B. Silverthorn, and R. Mooney, "Spherical Topic Models," in *International Conference on Machine Learning (ICML)*, 2010.

[25] A. Perina, M. Cristani, U. Castellani, V. Murino, and N. Jojic, "Free energy score spaces: Using generative information in discriminative classifiers," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 7, pp. 1249–1262, 2012.

[26] T. Jaakkola and D. Haussler, "Exploiting generative models in discriminative classifiers," *NIPS*, 1998.

[27] M. J. Swain and D. H. Ballard, "Color indexing," *International Journal of Computer Vision*, vol. 7, no. 1, pp. 11–32, 1991.

[28] T. Joachims, "A probabilistic analysis of the rocchio algorithm with TFIDF for text categorization," in *International conference on machine learning (ICML)*, 1997, pp. 143–151.

[29] A. Banerjee and S. Basu, "Topic models over text streams: a study of batch and online unsupervised learning," in *In Proceedings 7th SIAM International Conference on Data Mining*, 2007.

[30] A. Perina, N. Jojic, M. Bicego, and A. Turski, "Documents as multiple overlapping windows into grids of counts," in *Neural Information Processing Systems (NIPS)*, 2013.

[31] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *International Journal on Computer Vision*, vol. 42, no. 3, pp. 145–175, 2001.

[32] A. Perina, P. Lovato, V. Murino, and M. Bicego, "Biologically-aware latent dirichlet allocation (BaLDA) for the classification of expression microarray," in *Pattern recognition in bioinformatics (PRIB)*, 2010, pp. 230–241.

[33] U. Alon, "Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays," *PNAS*, vol. 96, pp. 745–750, 1999.

[34] M. Baiano, C. Perlini, G. Rambaldelli, R. Cerini, N. Dusi, M. Bellani, G. Spezzapria, A. Versace, M. Balestrieri, R. P. Mucelli, M. Tansella, and P. Brambilla, "Decreased entorhinal cortex volumes in schizophrenia," *Schizophrenia Research*, vol. 102, no. 1–3, pp. 171–180, 2008.

[35] J. Ashburner, "A fast diffeomorphic image registration algorithm," *Neuroimage*, vol. 38, no. 1, pp. 95–113, 2007.

[36] K. Friston, J. Ashburner, S. Kiebel, T. Nichols, and W. Penny, *Statistical Parametric Mapping: The Analysis of Functional Brain Images*. Academic Press, 2007.

[37] A. Perina and N. Jojic, "Spring lattice counting grids: Scene recognition using deformable positional constraints." in *ECCV (6)*, vol. 7577, 2012, pp. 837–851.

[38] H. Liu, L. Liu, and H. Zhang, "Ensemble gene selection by grouping for microarray data classification." *Journal of Biomedical Informatics*, vol. 43, no. 1, 2010.