Free Energy Score Spaces: Using Generative Information in Discriminative Classifiers

Alessandro Perina, Marco Cristani, *Member*, *IEEE*, Umberto Castellani, *Member*, *IEEE*, Vittorio Murino, *Senior Member*, *IEEE*, and Nebojsa Jojic

Abstract—A score function induced by a generative model of the data can provide a feature vector of a fixed dimension for each data sample. Data samples themselves may be of differing lengths (e.g., speech segments or other sequential data), but as a score function is based on the properties of the data generation process, it produces a fixed-length vector in a highly informative space, typically referred to as "score space." Discriminative classifiers have been shown to achieve higher performances in appropriately chosen score spaces with respect to what is achievable by either the corresponding generative likelihood-based classifiers or the discriminative classifiers using standard feature extractors. In this paper, we present a novel score space that exploits the free energy associated with a generative model. The resulting free energy score space (**FESS**) takes into account the latent structure of the data at various levels and can be shown to lead to classification performance that at least matches the performance of the free energy classifier based on the same generative model and the same factorization of the posterior. We also show that in several typical computer vision and computational biology applications the classifiers optimized in **FESS** outperform the corresponding pure generative approaches, as well as a number of previous approaches combining discriminating and generative models.

Index Terms—Hybrid generative/discriminative paradigm, variational free energy, classification.

1 Introduction

The design of models for classification and recognition purposes is one of the fundamental issues in computer vision. Among the several possible taxonomies, two apparently orthogonal approaches can be found in the literature: the generative and the discriminative paradigms.

Generative models are built to explain how samples could have been generated. Without a separate special notion of discrimination, they simply explain the data in such a way that the model parameters link hidden variables, which often have higher level semantics, to the observations so as to fit the probability density of the observed data. In such a model, classification can be treated as an inference problem: A model per-class is first fit, thus treating the class label during training as an additional higher level, but observed, variable, and then new samples are assigned to the category whose model fits best.

On the other hand, discriminative models target the boundaries among categories rather than the complete density function over the data. The philosophy of this approach is that avoiding modeling complex structure of the density models, the modeling power could potentially be focused only on differentiating the classes, which could thus lead to higher accuracy on the task at hand. Of course, the discriminative models may need to indirectly capture some of the complexities of the latent data structure when this is necessary for accurate classification.

The complementary nature of discriminative and generative approaches to machine learning [1] has motivated lots of research on the ways in which these can be combined [2], [3], [4], [5], [6], [7], [8]. These approaches can loosely be divided into three groups: *blending methods*, *iterative methods*, and *staged methods*.

In a few words, blending methods [2], [4], [5], [9], [10] try to optimize a single objective function that contains different terms coming from the generative and discriminative model. Iterative methods [11], [12], [13] are algorithms involving a generative and a discriminative model that are trained in an iterative process, each influencing the other. Finally, in staged methods [6], [8], [14], [15], [16], the models are trained in separate procedures, but one of the models—usually the discriminative model—is trained on some features provided by the first. This last family is currently the most frequently applied or studied by the community and it contains the family of methods called *generative score spaces* [7], which perform classification after projecting the samples into a fixed-dimensional space induced by a generative model. Empirical results have often indicated that the classification rate in such spaces outperforms both the direct classification based on the inference in the generative model and the discriminative classification in more straightforward feature spaces which are not based on the latent data structure.

In this paper, we present a novel generative score space, called Free Energy Score Space—**FESS**, exploiting variational free energy terms as features. The mapping arises

- *A. Perina is with Microsoft Research, One Microsoft Way, Redmond, WA 98052, and the University of Verona. E-mail: alperina@microsoft.com.*
- *N. Jojic is with Microsoft Research, One Microsoft Way, Redmond, WA 98052. E-mail: jojic@microsoft.com.*
- *M. Cristani and V. Murino are with the Italian Institute of Technology, Via Morego 30, Genova 16163, Italy and the University of Verona, Strada le Grazie 14, Verona 37135, Italy. E-mail: marco.cristani@univr.it.*
- *U. Castellani is with the University of Verona, Strada le Grazie 14, Verona 37135, Italy. E-mail: umberto.castellani@univr.it.*

Manuscript received 7 Sept. 2010; revised 11 June 2011; accepted 7 Oct. 2011; published online 8 Dec. 2011.
Recommended for acceptance by C. Sminchisescu.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2010-09-0688.
Digital Object Identifier no. 10.1109/TPAMI.2011.241.

naturally as a consequence of the factorization of the model. The free energy terms quantify the data fit in different parts of the model according to the posterior distribution and the uncertainty in the posterior distribution. Interestingly, the free energy terms seem to be informative for discrimination even when the model is imperfect. As illustrated in the experimental section, our approach tends to outperform the performances of generative score space-based methods proposed in the literature.

The rest of the paper is organized as follows: The next section reviews the state of the art of hybrid, generative-discriminative, approaches. In Section 3, the free energy score space is described in detail. In Section 4, we show that the proposed generative score space leads to better classification performances than the related generative counterpart. The score space is generalized in Section 5, showing how multiple score functions can be defined starting from the free energy formulation. An exhaustive experimental section is presented in Section 6, and final remarks are finally drawn in Section 7.

2 Generative, Discriminative and Hybrid Classifiers

Although generative models are trained simply to fit the density of the data, they are meant to accomplish this by involving a hierarchy of hidden variables. This latent structure, rather than just the quality of the density fit, is what often makes these models attractive. Human understanding of the world is often easily expressed in terms of a combination of hidden causes, and so specification of a generative model structure is intuitive. Furthermore, in the human understanding of the structure of hidden causes, those at higher levels of the hierarchy are imbued with meaning. This provides a hope that inference and structure learning algorithms for such models may lead to a pseudo-intelligent behavior. It is thus the ability to perform inference of hidden causes, rather than the quality of the density fit, that excites the researchers. The two are, of course, related, as a perfect density estimator would have to capture the true latent structure somehow, but when the models are imperfect, then the imperfection of the density fit could affect inference of some hidden causes more than the others. Actually, when the affected hidden variable is the class label, other approaches, e.g., a different generative model or a simpler feature-based discriminative algorithm, may considerably outperform an otherwise seductively general generative model.

In particular, generative models describe how the input data could be generated through a process involving a hierarchy of hidden variables, connected to each other through a hierarchy of conditional distributions. In this way, a correspondence between parts of the model and features is established. The hierarchical probability density modeling also allows for integration over hidden variables, and generative models handle missing, unlabeled, and varying-length data in an elegant unified way. Classification is also performed in the same way: The likelihood under such parameterized class-specific models can be used for classification using the Bayes rule.

If the sets of all the hidden and observed variables are denoted with H and X, then a generative model specifies the joint probability distribution $P(H, X)$. If $X = \{x^{(t)}\}$ represent a set of i.i.d. samples and $H = \{h^{(t)}\}$ the set of hidden variables associated with each sample, then the joint distribution is

$$\mathcal{L}_G = P(\theta) \cdot P(H, X|\theta) = P(\theta) \cdot \prod_{t=1}^{T} P\big(h^{(t)}, x^{(t)}|\theta\big), \quad (1)$$

where $P(\theta)$ is the parameter prior and the crucial terms $P(h^{(t)}, x^{(t)}|\theta)$ represent the modeling of what the data look like.

In the context of classification, $x^{(t)}$ may be sample descriptors and $c^{(t)}$ their class labels. To use a generative model for classification, for each class j we can learn the class-conditional density $\{P(X|C = j) = P(X|\theta_j\}$ that separately models each class and also estimate the prior $P(C)$ directly from the class labels. The Bayes rule then provides the posterior distribution:

$$P(C|X) = \frac{P(X, C|\theta)}{P(X|\theta)} = \frac{P(X|\theta_C) \cdot P(C)}{\sum_c P(X|\theta_C) \cdot P(C)}. \quad (2)$$

An equivalent view of this procedure is that the class label is simply treated as an additional variable which is observed during training, but not during testing. The class variable is thus not special in the latent structure in any way other than that it is available during training. As implied above, the consequence of this may be that as long as the model is still not reflecting the world perfectly, the modeling power may primarily target parts of the latent structure other than the class label itself.

For this reason, the discriminative methods target the separation boundaries among classes, rather than the distribution over instances of a class. In terms of probabilistic inference, discriminative modeling could be seen as directly targeting the conditional probability distribution $P(C|X)$:

$$\mathcal{L}_D = P(\theta) \cdot P(C|X, \theta) = P(\theta) \cdot \prod_{t=1}^{T} \sum_h P\big(c^{(t)}, h^{(t)}|x^{(t)}, \theta\big). \quad (3)$$

This is sometimes referred to as the conditional likelihood or discriminative likelihood. Integrating over hidden variables that barely affect likelihood near the decision boundaries has little effect on the conditional likelihood, and so optimizing this cost should focus the modeling power to the task at hand—classification—rather than capturing many other causes of variability in the data, which may only be interesting in some other applications. Furthermore, this provides a good reason to expect that the decision boundaries can potentially be modeled in a much simpler fashion than the generative models, and for this reason most discriminative approaches use general-purpose classification framework based on data features which are extracted in a model-free manner (e.g., image features based on local filters, or global measures of distribution of image colors).

Discriminative models perform well in many scientific areas, e.g., object recognition, economics, bioinformatics,

and text recognition. They often outperform generative models in classification tasks, especially when large training sets are available [1]. However, the classification boundaries themselves may be complex enough to require the use of hidden variables for their explanation. In contrast with generative models, typical discriminative approaches in the past suffered from difficulties in encoding a structured prior knowledge that could be relevant for classification. Also, as the decision boundaries are modeled, the approach is not as modular as the generative paradigm: The introduction of new classes into discriminative training typically requires contrasting the samples from the new class with those from all previously studied classes to learn the new boundaries, with little benefit from previous study of the same data with respect to other class boundaries.

In recent years, the complementary properties of the two families have encouraged attempts to combine their strengths. This has led to many different kinds of hybrid frameworks organized in the taxonomy described in the following.

2.1 Blending Methods

Blending methods rely on the optimization of hybrid objective functions that contains at least a discriminative and a generative term. These methods are often referred to as *hybrid learning*.

Discriminative learning optimizes the discriminative likelihood (i.e., (3)), which can also be written as

$$\begin{aligned}\mathcal{L}_C &= P(\theta)\prod_{t=1}^{T} P(c^{(t)}|x^{(t)},\theta)\\ &= P(\theta)\prod_{t=1}^{T}\frac{P(c^{(t)},x^{(t)}|\theta)}{\sum_c P(c^{(t)},x^{(t)}|\theta)}.\end{aligned} \tag{4}$$

Besides, Minka [17] suggests that a more natural view of dealing with it is to consider the class variable as simply a child variable in a generative model. The sample descriptors X are generated from some model (possibly with a rich latent structure) with some parameters $\tilde{\theta}$, and then the class label is generated from the data descriptors themselves according to some parameters θ. Since both sets of variables are observed during training, then the learning of the two sets of parameters decouples and the conditional distribution linking X and C can be trained independently of the model of the data density, i.e., the parameters $\tilde{\theta}$. In this sense, there is really no change in the apparatus for learning and inference when we switch to discriminative training, only a change concerning how the connections among variables are organized. So, Minka refers to these as discriminative models, not discriminative learning. However, this view also raises the question of whether it is possible to set up the models where the two parts (the generative likelihood of the data X and the conditional likelihood for the link from X to C) are not entirely decoupled in training, and if blending can lead to improved classification.

In [9], the authors train a discriminant function based on the log-likelihood ratio (so that it has generative parameters) with the maximum entropy criterion. On the other hand, Bouchard and Triggs [2] and Lasserre et al. [4] suggest merging the two objective functions $\mathcal{L}_D$ (3) and $\mathcal{L}_G$ (1). The idea here is to use a convex combination of the objective functions.

In [2], the authors use the following objective function:

$$\begin{aligned}\log\mathcal{L}(\theta) = P(\theta) &+ \alpha\cdot\log\mathcal{L}_G(\theta)\\ &+ (1-\alpha)\cdot\log\mathcal{L}_D(\theta),\end{aligned} \tag{5}$$

where α is a fixed weight. By varying α, one can work from pure generative training ($\alpha = 1$) to pure discriminative training ($\alpha = 0$). Because the parameters are not decoupled, the training is not decoupled either and some interaction is forced to happen.

In [4], two different sets of parameters θ and $\tilde{\theta}$, for the generative and discriminative parts, respectively, are learned using a prior to keep them near in the parameter space. The marginal likelihood is used in place of the generative likelihood (2):

$$\begin{aligned}\log\mathcal{L}(\theta) = P(\theta,\tilde{\theta}) &+ \alpha\cdot\log\mathcal{L}_D(\tilde{\theta})\\ &+ (1-\alpha)\cdot\log\sum_c P(X,C|\theta).\end{aligned} \tag{6}$$

A different approach, called multiconditional learning, has been studied in [5]. These authors suggest a framework whereby generative and discriminative components have different and unconstrained weights

$$\log\mathcal{L}(\theta) = \alpha\cdot\log\mathcal{L}_G(\theta) + \beta\cdot\log\sum_c P(X,C|\theta). \tag{7}$$

In all of the cases hybrid learning performs better in the middle of the two worlds (i.e., $\alpha \neq 0, 1$).

2.2 Iterative Methods

Iterative methods are examples of generative and discriminative models that help each other and are learned iteratively. The most known example is the wake sleep-like algorithm [12]. The original algorithm is developed in the context of unsupervised training of a neural network. The network is given a set of generative weights and a set of discriminative weights (called recognition in [12]).

In the "wake" phase, neurons are driven by recognition connections, and generative connections are adapted to increase the probability that they would reconstruct the correct activity vector in the layer below. In the "sleep" phase, neurons are driven by generative connections and recognition connections are adapted to increase the probability that they would produce the correct activity vector in the layer above.

An example of a recent application of this idea is [18], where the parameters of the constellation model for object recognition [19] are learned using an EM-like algorithm, where in the M-step, some quantities are learned discriminatively with an SVM. The process is repeated until convergence. They report better results than what is achieved by purely generative models.

Other algorithms that follows the ideas of Hinton et al. [12] can be found in the context of newsgroup categorization [20], biology [21], or motion/pose estimation [13], [22].

2.3 Staged Method and Score Spaces

Staged methods learn interesting features using a generative model, and use the derived feature vectors to train a

TABLE 1
Score Spaces

Score argument f	Score operator $\hat{F}$	Score mapping φ	Ref.
Loglikelihood	Identity	$\varphi^{lik}_{\hat{F}}(x) = [\log P(x\|\theta_c)]_{c=1}^{L}$	[7,26]
Loglikelihood	Gradient wrt θ_i	$\varphi^{fis}_{\hat{F}}(x) = \left[\nabla_{\theta_i} \log P_i(x\|\theta_i)\right]_{i=1}^{P}$	[6]
Loglikelhood of each class c	Gradient wrt $\theta_{i,c}$	$\varphi^{fis2}_{\hat{F}}(x) = \left[\nabla_{\theta_{i,c}} \log P_i(x\|\theta_{i,c})\right]_{c,i=1}^{L,P}$	[27]
Posterior log-odds	Gradient of order $k = 0 \dots K$ wrt θ_i	$\varphi^{TOP}_{\hat{F}}(x) = \left[\nabla^{(k)}_{\theta_i} \log \frac{P_i(x\|\theta_{i,1})}{P_i(x\|\theta_{i,2})}\right]_{k,i=1}^{K,P}$	[8]

discriminative classifier. Note that we have a generative step followed by a discriminative step.

For example, in [14], input samples are described using a conditional distribution coming from a probabilistic latent semantic analysis model (pLSA [23]) previously learned.

Yet another approach requires learning a generative model for each category, and then it performs inference giving different weights to the different components of the model [24]. These weights are learned discriminatively. Note that in this case again we have a generative step followed by a discriminative step. A similar approach was later used in [25] for speaker verification.

We include in this family of hybrid models the methods based on *score spaces*.

Using the notation of Smith and Gales [7], such spaces can be built from data by considering for each observed sequence $x^{(t)} = (x_1^{(t)}, \dots, x_k^{(t)}, \dots, x_K^{(t)})$ of observations $x_k^{(t)} \in \Re^d$, $k = 1, \dots, K$, a family of generative models $\mathcal{P} = \{P(X|\theta_i)\}$ parameterized by θ_i.

The observed sequence $x^{(t)}$ is mapped to the fixed-length score vector $\varphi^f_{\hat{F}}(x^{(t)})$:

$$\varphi^f_{\hat{F}}(x^{(t)}) = \hat{F}(f(\{P_i(x^{(t)}|\theta_i)\})), \tag{8}$$

where f is the function of the set of probability densities under the different models, and $\hat{F}$ is some operator applied to it. For methods that fall in this category, score argument, function, and mapping should be clearly determined.

The most popular example is the Fisher kernel (FK), introduced by Jaakkola and Haussler [6]. The idea is to use a discriminative model using feature vectors coming from a generative model, in this case the derivative of the loglikelihood of the data point with respect to the different parameters θ of the generative model. The features used by the discriminative model for data point x will be $\varphi^f_{\hat{F}}(x) = \nabla_\theta \log p(x|\theta)$. Coming back to (8), the score argument f is the loglikelihood, and the operator $\hat{F}$ produces the first order derivatives with respect to parameters. In [7], higher order derivatives are also included.

Another example of score space is the TOP kernel [8], for which the function f is the posterior log-odds and $\hat{F}$ is again the gradient operator.

The similarity-based approach of Bicego et al. [26] also falls in this category. The idea there is to describe a sample with the vector of its marginal likelihoods under all the classes. Instead of picking the Maximum likelihood (ML) classification (max), a discriminative classifier is used as an additional corrective stage. This approach is presented as the likelihood score space in [7].

In all these cases, the generative score space approaches help to distill the relationship between a model parameter θ_i and the particular data sample. After the mapping, a score space metric must be defined in order to employ discriminative approaches.

A number of useful properties of these mappings, and especially for Fisher score, can be derived. For example, for [6], [8] it was shown that the classification is asymptotically better than the generative classification.

Some popular score spaces are reported in Table 1, where we highlighted their score argument, operator, and mapping. L is the number of the classes, P is the number of the parameters, and k is the order of the gradient. It is also worth noting that [7] is a special case (K = 0) of the TOP kernel.

An experimental comparison on score spaces extracted from topic models in the context of microarray data classification can be found in [28].

One of the major drawbacks of generative score spaces is that they build upon the choice of one (or a few) out of many possible generative models, as well as the parameters fit to a limited amount of data. In practice, these models can therefore suffer from improper parameterization of the probability density function, local minima, overfitting, and undertraining problems. Consider, for instance, the situation where the assumed model over high dimensional data is a mixture of n diagonal Gaussians with a given small and fixed variance, and a uniform prior over the components. The only free parameters are therefore the Gaussian centers, and let us assume that training data are best captured with these centers all lying on (or close to) a hypersphere with a radius sufficiently larger than the Gaussians' deviation. An especially surprising and inconvenient outlier in this case would be a test data point that falls close to the center of the hypersphere, as the derivatives of its loglikelihood with respect to these parameters (Gaussian centers) evaluated at the estimate could be very low when the number of components n in the mixture is large because the derivatives are scaled by the uniform posterior $1/n$. But, this makes such a test point insufficiently distinguishable from the test points that actually satisfy the model perfectly by falling directly into one of the Gaussian centers. If the model parameters are extended to include the prior distribution over mixture components, then derivatives with respect to these parameters would help to disambiguate those points.

In this paper, we propose a novel hybrid method that belongs in this family and which focuses on how well the data point fits different parts of the generative model. The information passed from the generative to the discriminative models is extracted from the variational free energy as a

lower bound on the negative loglikelihood of the data. This affords us several advantages: First of all, the variational free energy can always be computed for an arbitrary structure of the posterior distribution, allowing us to deal with generative models with many latent variables and complex structure without compromising tractability, as was previously done for inference in generative models. The possibility of simplifying the posterior (i.e., using variational approximations of the free energy) allows us to deal with simpler models, which, while less precise, are often not just faster, but are less prone to local minima [29].

Second, a variational approximation of the posterior typically provides an additive decomposition of the free energy, providing many terms that can be used as features; this procedure identifies the score operator. These terms/features are divided into two categories: the "entropy set" of terms that express uncertainty in the posterior distribution, and the "cross-entropy set" describing the quality of the fit of the data to different parts of the model according to the posterior distribution.

Finally, it is worth noticing now how the posterior entropy is not taken into account in the previous score spaces [6], [8], [27]. In fact, as we will see in Section 3, the entropy of the hidden variables does not depend on the parameters and the differentiation (i.e., the score operator) makes it vanish.

We found the resulting score space to be highly informative for discriminative learning. An earlier version of this paper appeared in [30]. Here we extend [30], using novel score operators (i.e., the gradient), introducing novel ways to decompose the free energy and a table notation to describe such decomposition, and finally extending the experimental section. In particular, we tested our approach on several computational biology problems, as well as computer vision problems (scene/object recognition). The results compare favorably with the state of the art from the recent literature.

3 Free Energy Score Space

A generative model defines the distribution $P(H|\theta) = \prod_{t=1}^{T} P(h^{(t)}, x^{(t)}|\theta)$ over a set of observations $x = \{x^{(t)}\}_{t=1}^{T}$, each with associated hidden variables (hidden states) $h^{(t)}$, for a given set of model parameters θ shared across all observations. In addition, to model the posterior distribution $P(H|X)$, we also define a family of distributions $\mathcal{Q}$ from which we need to select a variational distribution $Q(H)$ that best fits the model and the data. Assuming i.i.d data, the family $\mathcal{Q}$ can be simplified to include only distributions of the form $Q(H) = \prod_{t=1}^{T} Q(h^{(t)})$.

The *free energy* [31], [32] is a function of the data, parameters of the posterior $Q(H)$, and the parameters of the model P, defined as

$$\begin{aligned}\mathcal{F}_{\mathcal{Q}} &= \mathbb{KL}(Q, P(H|X,\theta)) - \log P(X|\theta) \\ &= \sum_{H} Q(H) \log \frac{Q(H)}{P(X,H|\theta)}.\end{aligned} \tag{9}$$

The Kullback-Leibler (KL)-divergence is always positive and zero only if $Q(H)$ equals the true posterior probability; therefore minimizing $\mathcal{F}$ with respect to Q will always provide the negative loglikelihood $-\log P(X)$.

The free energy bounds the loglikelihood, $\mathcal{F}_{\mathcal{Q}} \leq -\log P(X)$ and the equality is attained only if $\mathcal{Q}$ is expressive enough to capture the true posterior distribution, as the free energy is minimized when $Q(H) = P(H|X)$. Constraining Q to belong to a simplified family of distributions $\mathcal{Q}$, however, provides computational advantages for dealing with intractable models P. Examples of distribution families used for approximation are the fully factorized mean field form [33] or the structured variational approximation [34], where some dependencies among the hidden variables are kept.

Minimization of $\mathcal{F}_{\mathcal{Q}}$ as a proxy for negative loglikelihood is usually achieved by an alternating optimization with respect to Q and θ, a special case of which—when Q is fully expressive—is the EM algorithm. Different choices of $\mathcal{Q}$ provide different types of compromise between the accuracy and computational complexity. For some models, accurate inference of some of the latent variables may require excessive computation even though the results of the inference can be correctly reinterpreted by studying the posterior Q from a simpler family and observing the symmetries of the model, or by reparametrizing the model (see, for example, [35]). In what follows, we will develop a technique that uses the parts of the free energy to infer the mapping of the data to a class variable with an increased accuracy despite possible imperfections of the data fit, whether this imperfection is due to the approximations and errors in the model or the posterior.

Having obtained an estimate of parameters $\hat{\theta}$ that fit the given i.i.d. data we can rearrange the free energy (9) as $\mathcal{F}_{\mathcal{Q}} = \sum_t \mathcal{F}^t_{\mathcal{Q}}$, with

$$\begin{aligned}\mathcal{F}^t_{\mathcal{Q}} = &\overbrace{\sum_{h^{(t)}} Q(h^{(t)}|\hat{\theta}) \cdot \log Q(h^{(t)}|\hat{\theta})}^{Entropy} \\ &- \overbrace{\sum_{h^{(t)}} Q(h^{(t)}|\hat{\theta}) \cdot \log P(h^{(t)}, x^{(t)}|\hat{\theta})}^{Cross-entropy}.\end{aligned} \tag{10}$$

The second term in the equation above is the *cross-entropy* term and it quantifies how well the data point fits the model, assuming that hidden variables follow the estimated posterior distribution. This posterior distribution is fitted to minimize the free energy, and the first term in (10) is the *entropy* quantifying the uncertainty of this fit.

If Q and P factorize, then each of these two terms further breaks into a sum of individual terms, each quantifying the aspects of the fit of the data point with respect to different parts of the model. For example, if the generative model is described by a Bayesian network, the joint distribution can be written as $P(z^{(t)} = \prod_n P(z_n^{(t)}|\mathbf{PA}_n)$, where $z^{(t)} = \{x^{(t)}, h^{(t)}\}$ denotes the set of all variables (hidden or visible) and $\mathbf{PA}_n$ are the parents of the $n-th$ of these variables, i.e., of $z_n^{(t)}$.

The cross-entropy term in the equation above further decomposes into

$$\begin{aligned}&\sum_{[z_1^{(t)}]} Q\big(z_1^{(t)} \cup \mathbf{PA}_1|\hat{\theta}\big) \cdot \log P\big(z_1^{(t)}|\mathbf{PA}_1, \hat{\theta}\big) + \cdots + \\ &\quad \sum_{[z_N^{(t)}]} Q\big(z_N^{(t)} \cup \mathbf{PA}_N|\hat{\theta}\big) \cdot \log P\big(z_N^{(t)}|\mathbf{PA}_N, \hat{\theta}\big).\end{aligned} \tag{11}$$

For each discrete hidden variable $z_n^{(t)}$, the appropriate terms above can be further broken down into individual terms in the summation over the D_n possible configurations of the variable, e.g.,

$$\begin{aligned} Q\big(z_n^{(t)} = 1, \cup\, \mathbf{PA}_n|\hat{\theta}\big) \cdot \log P\big(z_n^{(t)} = 1|\mathbf{PA}_n, \hat{\theta}\big) + \cdots + \\ Q\big(z_n^{(t)} = D_n, \cup\, \mathbf{PA}_n|\hat{\theta}\big) \cdot \log P\big(z_n^{(t)} = D_n|\mathbf{PA}_n, \hat{\theta}\big). \end{aligned} \tag{12}$$

In a similar fashion, the entropy term can also be decomposed further into a sum of terms as dictated by the factorization of the family $\mathcal{Q}$.

Therefore, the free energy for a single sample t can be expressed as the sum

$$\mathcal{F}_{\mathcal{Q}}^t = \sum_i f_{i,\hat{\theta}}^t, \tag{13}$$

where all the free energy pieces $f_{i,\hat{\theta}}^t$ derive from the finest decomposition (i.e., (12) or (11)).

The terms $f_{i,\hat{\theta}}^t$ describe how the data point fits possible configurations of the hidden variables in different parts of the model. Such information can be encapsulated in a score space that we call the *free energy score space* or simply **FESS**.

For example, in the case of a binary classification problem, given the generative models for the two classes, we can define as $\mathcal{F}_{(\mathcal{Q},\hat{\theta})}(x^{(t)})$ the mapping of $x^{(t)}$ to a vector of scores f with respect to a particular model with its estimated parameters, and a particular choice of the posterior family $\mathcal{Q}$ for each of the classes, and then concatenate the scores. Therefore, using the notation from [7], the free energy score operator $\varphi_{\hat{F}}^{FESS}(x^{(t)})$ is defined as

$$\varphi_{\hat{F}}^{FESS} : x^{(t)} \rightarrow \big[\mathcal{F}_{(\mathcal{Q}_1,\hat{\theta}_1)}(x^{(t)}); \mathcal{F}_{(\mathcal{Q}_2,\hat{\theta}_2)}(x^{(t)})\big], \tag{14}$$

where

$$\mathcal{F}_{(\mathcal{Q}_c,\hat{\theta}_c)} = \big[\ldots, f_{i,\hat{\theta}_c}^t, \ldots\big]^T, c = 1, 2. \tag{15}$$

If the posterior families are fully expressive, then the MAP estimate based on the generative models for the two classes can be obtained from this mapping by simply summing the appropriate terms to obtain the log-likelihood difference as the free energy equals the negative loglikelihood.

However, the mapping also allows for the parts of the model fit to play uneven roles in classification after an additional step of discriminative training. In this case, the data points do not have to fit either model well in order to be correctly classified. Furthermore, even in the extreme case where one model provides a higher likelihood than the other for the data from *both* classes (e.g., because the models are not nested, and likelihoods cannot be directly compared), the mapping may still provide an abstraction from which another step of discriminative training can benefit. The additional step of training a discriminative model allows for mining the similarities among the data points in terms of the path through different hidden variables that has to be followed in their generation. These similarities may be informative even if the generative process is imperfect.

Obviously, (14) can be generalized to include multiple models (or the use of a single model) and/or multiple posterior approximations, either for two-class or multiclass classification problems.

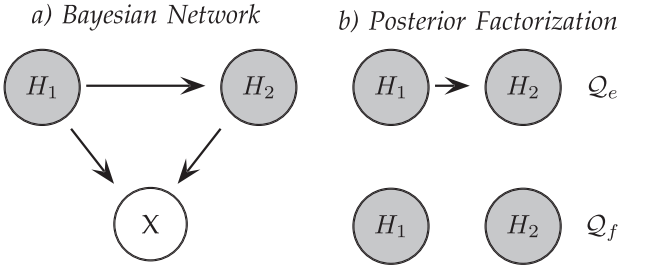

Fig. 1. a) An example of a Bayesian network. b) Two different posterior factorization, each of them identifies a particular family; $\mathcal{Q}_e$ is the exact posterior, and $\mathcal{Q}_f$ is the mean field approximation.

When the exact posterior is used, the minimization of the free energy is equivalent to the standard maximum likelihood approach, and it is anatically ec.

3.1 Example: How to Choose the Score Function $\hat{F}$

To better understand how to build and choose a particular score operator, consider the generative model described by the Bayesian network in Fig. 1. It is characterized by a visible variable X and two hidden variables H_1 and H_2. Suppose that both H_i assume values in $\{1, \ldots, D\}$. The joint probability distribution factorizes as follows:

$$P(X, H_1, H_2) = P(X|H_1, H_2) \cdot P(H_2|H_1) \cdot P(H_1). \tag{16}$$

The hidden variables factorize according to the family $\mathcal{Q}$ chosen. In this case, we can choose between an unconstrained, form $\mathcal{Q}_u = Q(H_1, H_2)$, or a still exact form, but parameterized differently, e.g., $\mathcal{Q}_e = Q(H_2|H_1) \cdot Q(H_1)$, or the fully factorized form $\mathcal{Q}_f = Q(H_2) \cdot Q(H_1)$. Please note that in this case the true posterior distribution factorizes as in $\mathcal{Q}_e$.[1] For example, if we take the fully factorized family, the free energy of this model becomes

$$\begin{aligned} \mathcal{F}_{\mathcal{Q}_f} = \sum_t \Bigg(& \sum_{h_1=1}^{D} Q\big(h_1^{(t)}\big) \log Q\big(h_1^{(t)}\big) \\ & + \sum_{h_2=1}^{D} Q\big(h_2^{(t)}\big) \log Q\big(h_2^{(t)}\big) \\ & - \sum_{h_1=1}^{D} Q\big(h_1^{(t)}\big) \log P\big(h_1^{(t)}\big) \\ & - \sum_{h_1,h_2=1}^{D} Q\big(h_1^{(t)}\big) \cdot Q\big(h_2^{(t)}\big) \log P\big(x^{(t)}|h_1^{(t)}, h_2^{(t)}\big) \\ & - \sum_{h_1,h_2=1}^{D} Q\big(h_1^{(t)}\big) \cdot Q\big(h_2^{(t)}\big) \log P\big(h_2^{(t)}|h_1^{(t)}\big) \Bigg), \end{aligned} \tag{17}$$

where the first two terms represent the entropy and the remaining three the cross entropy; each term refers to "local" parts of the model. At this point, we can focus on Table 2 to better understand (11)-(12): Generative classification "maps" a sample in a single value, its free energy $\mathcal{F}^t$ (loglikelihood). At the first level of detail L_1, we can map a sample (i.e., $\varphi_{\hat{F}}^{FESS}(x^{(t)})$) in two values, its entropy and its cross entropy. As a second level of detail L_2, the unique

1. Being unconstrained, $\mathcal{Q}_u$ is always fully expressive and it always captures the true posterior distribution q^*; in this case we have $q^* \subset \mathcal{Q}_e = \mathcal{Q}_u$.

TABLE 2
Definition of the Score Function for Choosing a Particular Score Space

Piece	Level of detail L_1	L_2	L_3
$\mathcal{Q}_f$			
$\sum_{h_1=1}^{D} Q(h_1^{(t)}) \log Q(h_1^{(t)})$	1	1	D
$\sum_{h_2=1}^{D} Q(h_2^{(t)}\|h_1^{(t)}) \cdot Q(h_1^{(t)}) \log Q(h_2^{(t)})$		1	D
$\sum_{h_1=1}^{D} Q(h_1^{(t)}) \log P(h_1^{(t)})$	1	1	D
$\sum_{h_1,h_2=1}^{D} Q(h_1^{(t)}) \cdot Q(h_2^{(t)}) \log P(x^{(t)}\|h_1^{(t)}, h_2^{(t)})$		1	D^2
$\sum_{h_1,h_2=1}^{D} Q(h_1^{(t)}) \cdot Q(h_2^{(t)}) \log P(h_2^{(t)}\|h_1^{(t)})$		1	D^2
$\mathcal{Q}_e$			
$\sum_{h_1=1}^{D} Q(h_1^{(t)}) \log Q(h_1^{(t)})$	1	1	D
$\sum_{h_1,h_2=1}^{D} Q(h_1^{(t)}) \cdot Q(h_2^{(t)}\|h_1^{(t)}) \log Q(h_2^{(t)}\|h_1^{(t)})$		1	D^2
$\sum_{h_1=1}^{D} Q(h_1^{(t)}) \log P(h_1^{(t)})$	1	1	D
$\sum_{h_1,h_2=1}^{D} Q(h_1^{(t)}) \cdot Q(h_2^{(t)}\|h_1^{(t)}) \log P(x^{(t)}\|h_1^{(t)}, h_2^{(t)})$		1	D^2
$\sum_{h_1,h_2=1}^{D} Q(h_1^{(t)}) \cdot Q(h_2^{(t)}\|h_1^{(t)}) \log P(h_2^{(t)}\|h_1^{(t)})$		1	D^2

factorizations induced by the generative model and by the family $\mathcal{Q}$ break the free energy in several contributions, that is, in this case two entropy terms and three cross entropy terms (see Table 2, level of detail or (17)). At the finest level of decomposition L_3, in case of discrete valued variable, each term can be broken down considering the values each hidden variable can assume, so that, for example, the first entropy term $\sum_{h_1=1}^{D} Q(h_1^{(t)}) \log Q(h_1^{(t)})$ is the sum of D contributions, and the last cross-entropy term $\sum_{h_1,h_2=1}^{D} Q(h_1^{(t)}) \cdot Q(h_2^{(t)}) \log P(h_2^{(t)}|h_1^{(t)})$ is the sum of D^2 contributions.

Summarizing, we have three levels of detail which define three different score functions and different score spaces

$$L_1 : \left|\varphi^{FESS}_{\hat{F}_{L_1}}(x^{(t)})\right| = 2, \tag{18}$$

$$L_2 : \left|\varphi^{FESS}_{\hat{F}_{L_2}}(x^{(t)})\right| = 5, \tag{19}$$

$$L_3 : \left|\varphi^{FESS}_{\hat{F}_{L_3}}(x^{(t)})\right| = 2 \cdot D^2 + 3 \cdot D. \tag{20}$$

The same considerations hold for the other factorizations; for example, in Table 2 we reported the three levels of detail if we employ the family $\mathcal{Q}_e$.

4 Free Energy Score Space Classification Dominates the MAP Classification

We use here the terminology introduced in [8], under which **FESS** can be considered a *model-dependent feature extractor*, as different generative models lead to different feature vectors [36]. The family of feature extractors $\varphi_{\hat{F}} : \mathcal{X} \rightarrow \Re^d$ maps the input data $x^{(t)} \in \mathcal{X}$ in a space of fixed dimension derived from a plug-in estimate λ: in our case, the generative model with parameters $\hat{\theta}$ from which the features are extracted.

Given some observations $x^{(t)}$ and the corresponding class labels $c^{(t)} \in \{-1, +1\}$ following the joint probability $P(X, C|\theta^*)$, a generative model can be trained to provide an estimate $\hat{\theta} \neq \theta^*$, where θ^* are the true parameters. As most kernels (e.g., Fisher and TOP) are commonly used in combination with linear classifiers such as linear SVMs, Tsuda et al. [8] propose as a starting point for evaluating the performance of a feature extractor the classification error of a linear classifier $w^T \cdot \varphi_{\hat{F}}(x) + b$ in the feature space $\Re^d$, where $w \in \Re^d$ and $b \in \Re$. Assuming that w and b are chosen by an optimal learning algorithm on a sufficiently large training data set and that the test set follows the same distribution with parameter θ^*, the classification error $R(\varphi_{\hat{F}})$ can be shown to tend to

$$R(\varphi_{\hat{F}}) = \min_{w,b} E_{x,c} \Phi\left[-c \cdot \left(w^T \cdot \varphi_{\hat{F}}\left(x^{(t)}\right) + b\right)\right], \tag{21}$$

where $\Phi[a]$ is an indicator function which is 1 when $a > 0$, and 0 otherwise, and $E_{x,y}$ denotes the expectation with respect to the true distribution $P(X, C|\theta^*)$.

In [6], [8], it has been shown that the Fisher kernel classifier can perform at least as well as its plug-in estimate if the parameters of a linear classifier are properly determined:

$$R\left(\varphi^{FK}_{\hat{F}}\right) \leq E_{x,c} \Phi\left[-c \cdot \left(P\left(c^{(t)} = +1|x^{(t)}, \hat{\theta}\right) - \frac{1}{2}\right)\right] = R(\lambda), \tag{22}$$

where λ represents the generative model used as plug-in estimate.

This property also trivially holds for our method, where $\varphi_{\hat{F}}(x^{(t)}) = \varphi^{FESS}_{\hat{F}}(x^{(t)})$, because the free energy can be expressed as a linear combination of the elements of φ.

In fact, the minimum free energy test (and the maximum likelihood rule when $\mathcal{Q}$ is fully expressive) can be defined on φ derived from the generative models with parameters $\hat{\theta}_{+1}$ for one class and $\hat{\theta}_{-1}$ for the other as

$$\begin{aligned} \hat{y} &= \min_y \left\{\mathcal{F}^t_{(\mathcal{Q},\hat{\theta}_{+1})}, \mathcal{F}^t_{(\mathcal{Q},\hat{\theta}_{-1})}\right\} \\ &= \Phi\left[\mathbf{1}^T \mathcal{F}_{(\mathcal{Q},\hat{\theta}_{+1})}(x^{(t)}) - \mathbf{1}^T \mathcal{F}_{(\mathcal{Q},\hat{\theta}_{-1})}(x^{(t)})\right]. \end{aligned} \tag{23}$$

Given (23), it is straightforward to prove that the error made by the kernel classifier that works in **FESS** (i.e., $R(\varphi^{FESS}_{\hat{F}})$) is as low as the error made by the MAP labeling based on the generative models (i.e., $R_{\mathcal{Q}}(\lambda)$) for the two classes since generative classification is a special case of our framework. In practice, we have to prove that (21) holds for **FESS**, so we have that

$$R\left(\varphi^{FESS}_{\hat{F}}\right) = \min_{w,b} E_{x,c} \Phi\left[-c \cdot \left(w^T \cdot \varphi^{FESS}_{\hat{F}}\left(x^{(t)}\right) + b\right)\right] \tag{24}$$

$$\leq E_{x,c} \Phi\left[-c \cdot \left(w^T \cdot \varphi^{FESS}_{\hat{F}}\left(x^{(t)}\right) + b\right)\right] \;\; \forall \;\; w, b, \tag{25}$$

where (25) holds because we are considering any w and b, and in (24) we were considering them optimally chosen to minimize the error.

If (25) holds for any choice of w and b, it would also hold for the particular choice $w = w_g$ and $b = b_g$.

$$\begin{aligned} R\left(\varphi^{FESS}_{\hat{F}}\right) &\leq E_{x,c} \Phi\left[-c \cdot \left(w_g^T \cdot \varphi^{FESS}_{\hat{F}}\left(x^{(t)}\right) + b_g\right)\right] \\ &for \;\; w_g = \left[\overbrace{+1, \ldots, +1}^{M_1 \;\; times}, \overbrace{-1, \ldots, -1}^{M_2 \;\; times}\right]^T, \\ &b_g = 0. \end{aligned} \tag{26}$$

Above, the first M_1 elements are the components of the free energy for one model and the remaining M_2 for the second model. One can notice that (26) implements the free energy test (23); therefore we have proven that $R(\varphi_{\hat{F}}^{FESS}) \leq R_{\mathcal{Q}}(\lambda)$.

Furthermore, when the family $\mathcal{Q}$ is expressive enough to capture the true posterior distribution, the free energy test is equivalent to maximum likelihood classification, $R_{\mathcal{Q}}(\lambda) = R(\lambda)$. The dominance of the Fisher and TOP kernels [6], [8] over their plug-in holds for **FESS** too, and the same plug-in (the likelihood under a generative model) may be used when this is tractable. However, if the computation of the likelihood (and the kernels derived from it) is intractable, then the free energy test, as well as the kernel methods based on **FESS** that will outperform this test, can still be used with an appropriate family of variational distributions $\mathcal{Q}$.

5 Controlling the Length of the Feature Vector: A Set of Score Spaces Based on Free Energy

In some generative models, especially sequence models, the number of hidden variables may change from one data point to the next. Let us describe this issue with an example. In speech processing, hidden Markov models (HMMs) [37] may be used to model utterances $x_1^{(t)}, \ldots, x_{K(t)}^{(t)}$ of different sequence lengths $K(t)$. As each element in the sequence has an associated hidden variable, the hidden state sequences $s_1^{(t)}, \ldots, s_{K(t)}^{(t)}$ are also of variable lengths. The parameters θ of this model include the prior state distribution π, the state transition probability matrix $\mathbf{A} = a_{\{ij\}} = Q(s_k = i | s_{k-1} = j)$, and the emission probabilities $\mathbf{B} = b_{\{iv\}} = Q(s_k = i | x_k = \{iv\})$. Exact inference is tractable in HMMs and so we can use the exact posterior distribution to formulate the free energy and the free energy minimization is equivalent to the usual Baum-Welch training algorithm [38] and $\mathcal{F}_{\mathcal{Q}_e} = -\log P(X)$. The free energy of each sample $x^{(t)}$ is reported in (27):

$$\begin{aligned}\mathcal{F}_{\mathcal{Q}_e}^t = &\sum_{[s]} Q\big(s_1^{(t)}\big) \log Q\big(s_1^{(t)}\big) \\ &+ \sum_{[s]} \sum_{k=2}^{K(t)} Q\big(s_k^{(t)}, s_{k+1}^{(t)}\big) \log Q\big(s_k^{(t)} | s_{k-1}^{(t)}\big) \\ &- \sum_{[s]} Q\big(s_1^{(t)}\big) \log \pi_{s_1^{(t)}} \\ &- \sum_{[s]} \sum_{k=2}^{K(t)} Q\big(s_k^{(t)}, s_{k-1}^{(t)}\big) \log a_{\{s_k^{(t)}, s_{k-1}^{(t)}\}} \\ &- \sum_{[s]} \sum_{k=1}^{K(t)} Q\big(s_k^{(t)}\big) \log b_{\{s_k^{(t)}, x_k^{(t)}\}}.\end{aligned} \tag{27}$$

Depending on how this is broken into terms f_i, we could get feature vectors whose dimension depends on the length of the sample $K(t)$. To solve this problem, we first note that a standard approach to dealing with utterances of different lengths is to normalize the likelihood by the sequence length, and this approach is also used for defining other score spaces. If, before the application of the score operator, we simply evaluate the sums over k in the free energy and divide each by $K(t)$, we obtain a fixed number of terms independent of the sequence length. This results in a length-normalized score space **nFESS**, where the granularity of the decomposition of the free energy is dramatically reduced.

In general, even for fixed-length data points and arbitrary generative models, we do not need to create large feature vectors corresponding to the finest level of granularity described in (12), or for that matter the slightly coarser level of granularity in (11). Some of the terms in these equations can be grouped and summed up to ensure for shorter feature vectors, if this is warranted by the application. The longer the feature vector, the finer the level of detail with which the generative process for that data sample is represented, but more data are also needed for the training of the discriminative classifier. Domain knowledge can often be used to reduce the complexity of the representation by summing appropriate terms without sacrificing the amount of useful information packed in the feature vectors. Moreover, as happens for Jaakkola and Haussler [6], Li et al. [16], standard dimensionality reduction techniques, e.g., PCA, can be employed.

In Table 3, we reported two examples of normalized FESS (**nFESS**), showing how a different level of detail for different pieces can be chosen.

In the first case, $\hat{F}_a$, we are not interested in entropy and we only kept one term.[2] Since we are summing $D + D$ contributes, we normalize multiplying for $\frac{1}{2 \cdot D}$ (see Table 3, column "Norm.Const"). Then, we kept the maximum granularity for the local contribution:

$$\sum_{h_1=1}^{D} Q\big(h_1^{(t)}\big) \log P\big(h_1^{(t)}\big)$$

because we suppose it is very important for the problem at hand; finally, we only keep a term for each of the two remaining cross-entropy components. In this way, we have defined the score function $\hat{F}_a$ and the resulting score space has dimension equal to $D + 3$. Analogously, we can choose to group different terms and define the score space $\hat{F}_b$ whose final dimension is $D^2 + 2 \cdot D + 2$ (see Table 3).

When a term at the second level (L_2) is the result of more than a summation, a spurious level of detail can be used. For example, consider the term $\sum_{h_1,h_2=1}^{D} Q(h_1^{(t)}) \cdot Q(h_1^{(t)}) \log P(h_2^{(t)} | h_1^{(t)})$. At level 2, one has to perform the summations over H_1 and H_2 yielding to a single value, and at level 3 each addendum of the summation is taken as feature, yielding to D^2 values. In this case, intermediate levels can be obtained performing only the summation of H_1 (or H_2), yielding to only D values: We call this intermediate level $L_3^{(H_1)}$ (or $L_3^{(H_2)}$), where the apex identifies the summation performed. As we will see in the experiments' section, this level of detail is very important for variable length descriptions like the possible inputs to hidden Markov models or latent Dirichlet allocation.

Such control of the feature vector length does not negate the previously discussed advantages of the classification in

2. We circled the values in Table 3.

TABLE 3
Definition of the Score Function for the Normalized FESS

Pieces	Score Function $\hat{F}_a$					Norm.	Score Function $\hat{F}_b$					Norm.
	L_1	L_2	$L_3^{(h_1)}$	$L_3^{(h_2)}$	L_3	Const.	L_1	L_2	$L_3^{(h_1)}$	$L_3^{(h_2)}$	L_3	Const.
$\sum_{h_1}^{D} Q(h_1^{(t)}) \log Q(h_1^{(t)})$	(1)	1			D	$\frac{1}{2 \cdot D}$	1	1			(D)	1
$\sum_{h_2}^{D} Q(h_2^{(t)}) \log Q(h_2^{(t)})$		1			D			1			(D)	1
$\sum_{h_1}^{D} Q(h_1^{(t)}) \log P(h_1^{(t)})$	1	1			(D)	1	1	(1)			D	$\frac{1}{D}$
$\sum_{h_1,h_2}^{D} Q(h_1^{(t)}) \cdot Q(h_2^{(t)}) \log P(x^{(t)} \mid h_1^{(t)}, h_2^{(t)})$		(1)	D	D	D^2	$\frac{1}{D}$		1	D	D	(D^2)	1
$\sum_{h_1,h_2}^{D} Q(h_1^{(t)}) \cdot Q(h_2^{(t)}) \log P(h_2^{(t)} \mid h_1^{(t)})$		(1)	D	D	D^2	$\frac{1}{D^2}$		(1)	D	D	D^2	$\frac{1}{D^2}$

The decomposition refers to the Bayesian network depicted in Fig. 1a, the family chosen is the fully factorized family $\mathcal{Q}_f$ (see also Fig. 1b). In the column "Norm. Const." we explicitly reported the normalization constants for each piece f_i.

TABLE 4
Free Energy Score Space for Hidden Markov Models

Pieces	Score Function $\hat{F}_{HMM}$				
	L_1	L_2	$L_3^{(S_k,S_{k+1})}$	$L_3^{(k)}$	L_3
$\sum_s Q(s_1^{(t)}) \log Q(s_1^{(t)})$	1	1			(Q)
$\sum_{[s]} \sum_{k=1}^{K(t)-1} Q(s_k^{(t)}, s_{k+1}^{(t)}) \log Q(s_{k+1}^{(t)} \mid s_k^{(t)})$		1	$K(t)$	(Q^2)	$(K(t)-1) \cdot Q^2$
$\sum_s Q(s_1^{(t)}) \log \pi_{s_1^{(t)}}$	1	1			(Q)
$\sum_{[s]} \sum_{k=1}^{K(t)-1} Q(s_k^{(t)} \mid s_{k+1}^{(t)}) \log a_{\{s_k^{(t)}, s_{k+1}^{(t)}\}}$		1	$K(t)$	(Q^2)	$(K(t)-1) \cdot Q^2$
$\sum_s Q(s_1^{(t)}) \log b_{\{s_k^{(t)}, x_k^{(t)}\}}$		1			(Q)

the free energy score space compared with the straightforward application of free energy, likelihood, or in the case of sequence models, length-normalized likelihood tests.

Since the free energy score space defined in Section 3, Table 2, is generalized by **nFESS**, in the following we will refer to both families of spaces as **FESS**. What differentiates the various score spaces is the particular choice of the score operator $\hat{F}$.

6 Experiments

We evaluated our approach on several standard data sets and compared its performance with the classification results provided by the data sets' creators, those estimated using the plug-in estimate λ, and those obtained using the Fisher (**FK**) and TOP (**TK**) kernels [6], [8] derived from the plug-ins.[3] Support vector machines (SVMs) with linear and RBF kernels were used as discriminative classifiers. As plug-ins, or generative models/likelihoods λ, for the three score spaces compared across experiments, we used hidden Markov models [37] in Experiments 1-2, and latent Dirichlet allocation (LDA) [39] in Experiments 3-5.

Comparisons are based on the same validation procedure used in the papers that introduced the data sets. To ensure the repeatability of results, we detailed their procedure in every experiment. The code to extract FESS for pLSA and HMM is available on our webpages.

6.1 Hidden Markov Models

Using the HMM as plug-in estimate we first focused on computational biology examples.

We considered three different families for the posterior distribution: exact ($\mathcal{Q}_e$), mean field ($\mathcal{Q}_f$), and a structured approximation ($\mathcal{Q}_c$). In formulas:

$$\mathcal{Q}_e = Q(s_1) \cdot \prod_{k=2}^{K} Q(s_k \mid s_{k-1}),$$

$$\mathcal{Q}_c = \prod_{k=1}^{K} Q(s_k, s_{k+1}), \quad \mathcal{Q}_f = \prod_{k=1}^{K} Q(s_k).$$

For what concerns the exact posterior, the free energy of an HMM is reported in (27), and we report the score function that defines the score argument in Table 4. The free energy, score functions, and score argument for the other two families are straightforward to extract. The dimensionality of the score vectors and other details on the experiment are reported separately for each experiment. We have always chosen the maximum level of detail. For what concerns the HMM parameters, we used a random initialization and we estimated the number of the states Q using hold-out likelihood, with a 10-folds cross evaluation. In all the tests, the loglikelihood peaked around $Q \approx 10$.

6.1.1 Experiment 1—E. Coli Promoter Gene Sequences

The first analyzed data set consists of the E. coli promoter gene sequences (DNA) with associated imperfect domain theory [40].[4] The standard task on this data set is to recognize promoters in strings of nucleotides (A, G, T, or C). A promoter is a genetic region which facilitates the transcription of gene located nearby. The input features are 57 sequential DNA nucleotides. The results are obtained using leave-one-out (LOO). We trained a generative model λ_{HMM} once for each left-out sample. For

3. When computable.

4. This data set is available at [41].

TABLE 5
Promoter Classification Results

E.Coli Algorithm	Exact [37]	Posterior $\mathcal{Q}_f$	$\mathcal{Q}_c$
λ_{HMM}	74.53%	71.23%	72.09%
$\hat{F}_{HMM}$ (lin)	75.47%	75.47%	74.53%
$\hat{F}_{HMM}$ (rbf)	81.13%	83.96%	81.13%
$\rightarrow$ *len.*	$2Q^2+3Q$	$2Q^2+3Q$	Q^2+3Q
$\hat{F}_{L_3}$ (lin)	91.51%	93.40%	91.51%
$\hat{F}_{L_3}$ (rbf)	94.33%	90.57%	91.51%
$\rightarrow$ *len.*	$2KQ^2+2KQ$	$2KQ^2+2KQ$	$2KQ^2+2KQ$
FK	79.20%	-	-
$\rightarrow$ *len.*	Q^2+2Q	-	-
TK	85.30%	-	-
$\rightarrow$ *len.*	Q^2+2Q+1	-	-

TABLE 6
Introns/Exons Classification Results

E.Coli Algorithm	Exact [37]	Posterior $\mathcal{Q}_f$	$\mathcal{Q}_c$
λ_{HMM}	76.35%	68.70%	70.54%
$\hat{F}_{HMM}$ (lin)	95.27%	95.27%	95.27%
$\hat{F}_{HMM}$ (rbf)	96.48%	97.06%	96.87%
$\rightarrow$ *len.*	$2Q^2+3Q$	$2Q^2+3Q$	Q^2+3Q
FK	89.94%	-	-
$\rightarrow$ *len.*	Q^2+2Q	-	-
TK	87.18%	-	-
$\rightarrow$ *len.*	Q^2+2Q+1	-	-
[15]	92.50%	-	-
$\rightarrow$ *len.*	Q^2+2Q	-	-

each such test point, the model is learned only on the training set consisting of all data but that point. The training data points are mapped in **FESS** via $\hat{F}_{HMM}$ based on the model, and the discriminative training is performed only on these same training points. This procedure yields two rules: 1) the way to assign features to any data point, and 2) the rule for assigning the class to the data point based on these features. These two rules, solely based on the training data, are then used to assign the class to the test point. This procedure has been repeated for the three posterior families we considered.

Results are reported in Table 5 and illustrate that **FESS** represents well the fixed size genetic sequences, leading to a superior performance over other score spaces as well as over the plug-in λ_{HMM}. This test also gives the opportunity to compare two different score functions: In this particular experiment, the sequences all have the same (reasonable) length $K = 57$, so the maximum level of detail can be employed L_3 (i.e., (20)); the length of the score vectors is reported in Table 5 in the rows labeled with *"len."*

As results show, when **FESS** is employed using its maximum resolution L_3, the improvement with respect to **FK** and **TK** is impressive. The underlying motivation is that dimensions of feature vectors **FK**, **TK**, and $\hat{F}_{HMM}$ are calculated via "temporal means"; therefore they do not keep the information for each temporal instant k (position, in our case) of sequences separate, whereas L_3 has several dimensions that refer explicitly at each position in the sequence; this information is very useful when dealing with biological sequences like promoters or genotypes [42]. Moreover, $\hat{F}_{L_3}$ outperforms $\hat{F}_{HMM}$ since fewer optimization factors w_i are involved.

As expected, the posterior family has influence in the generative classification: the coarser the approximation, the worse the performances. This does not hold once discriminative classifiers are used indeed all the three families seem perform equally well.

6.1.2 Experiment 2—Introns/Exons Classification in HS³D Data Set

The HS^3D data set[5] [15] contains labeled intron and exon sequences of nucleotides. The task here is to distinguish between the two types of gene sequences that can both vary in length (from dozens of nucleotides to tens of thousands of nucleotides). This setting gives us the opportunity of asserting the validity of the score normalizations (see Section 5). For the sake of comparison, we adopted the same experimental setting of Jebara et al. [15].

We learn a single generative model using all the training samples; subsequently, we extracted the scores for all the data, using once again the scores of the training samples to learn the SVMs. The length of the score vectors are reported in Table 6, in the rows labeled with *"len."* Table 6 also summarizes the results, showing that, **FESS** once again outperforms all the comparisons with statistical significance, beating the state of the art on this data set.

6.2 Latent Dirichlet Allocation

Using latent Dirichlet allocation as plug-in estimate, we focused on computer vision examples.

Topics models such as pLSA [23] and LDA [39] have been successfully employed in computer vision tasks such as scene classification [43], [44]. In this formulation, each image $I^{(t)}$ is represented as a collection of $N(t)$ detected patches or visual words $\{x_n^{(t)}\}_{n=1}^{N(t)}$, taking word labels from a previously trained codebook of W words. LDA uses a finite number of hidden topics Z to model the co-occurrence of visual words inside and across images. Each visual word $x_n^{(t)}$ is assigned a hidden topic $z_n^{(t)}$, where $P(x_n|z_n) = \beta$, and each image is explained as a mixture of hidden topics $\theta_z^{(t)}$. For convenience, the mixture of topics is sampled from a Dirichlet distribution of hyperparameter α.

For more details and for LDA free energy, see [39].

In Table 7, we report the score function that defines the score argument; the final length of the score vectors is $4Z$. Unlike probabilistic latent semantic analysis, LDA adds the Dirichlet prior α on the per-document topic distribution [45]; therefore **FESS** can be easily extracted simply ignoring

TABLE 7
Free Energy Score Space for Latent Dirichlet Allocation

Pieces	Score Function $\hat{F}_{LDA}$			
	L_2	$L_3^{(Z_k)}$	$L_3^{(k)}$	L_3
$\sum_z Q(z_k^{(t)}) \log Q(z_k^{(t)})$	1	$K(t)$	Z	$K(t) \cdot Z$
$\sum_z Q(z_k^{(t)}) \log P(z_k^{(t)}\|\theta)$	1	$K(t)$	Z	$K(t) \cdot Z$
$\sum_z Q(z_k^{(t)}) \log P(x_k^{(t)}\|z_k^{(t)}, \beta)$	1	$K(t)$	Z	$K(t) \cdot Z$
$\sum_z \theta_z \log \alpha_z$	1			Z

We omitted L_1.

5. http://www.sci.unisannio.it/docenti/rampone.

TABLE 8
Scene Classification: Comparison with the State of the Art

Dataset	*Best*	*Auth.*	$\hat{F}_{LDA}$
VS (6)	85.7% [44]	75.1% [47]	**90.30%**
OT^N (4)	90.2% [44]	89.0% [46]	**95.21%**
OT^A (4)	92.5% [44]	89.0% [46]	**94.38%**
FP (13)	73.4% [44]	65.2% [43]	**84.31%**
LZP (15)	81.1% [48]	81.1% [48]	**82.32%**

the last free energy piece in Table 7. This yield to a score vector of length $3Z$.

As input for LDA, we extracted SIFT features from 16×16 pixel patches computed over a grid with spacing of 8 pixels; we used 40 topics ($Z = 40$) and 175 codewords ($W = 175$). We use the wide literature on these models [43], [44] to choose a good estimate of the model parameters.

For each test we trained C generative models,[6] one for each class, with a random initialization of the model parameters. We used half of the training set designated by the database authors to do this, keeping the second half to learn the discriminative model. Afterward, we mapped the samples using the appropriate score argument from the rest of the data set.

6.2.1 Experiment 3—Scene Classification on Several Data Sets

We used these models as the generative starting point and evaluated our classification algorithm on three different popular data sets: 1) Oliva and Torralba [46], 2) Vogel and Schiele [47], and 3) Fei Fei and Perona [43]. We will refer to these data sets as **OT**, **VS**, and **FP**, respectively. For each test, we calculated the classification accuracy over the test set, repeating the process 10 times and averaging the results.

Results for each data set are summarized in Table 8, where we compare the accuracy of our approach with the accuracy achieved by the data sets' authors and the current state of the art. The methods presented in [46], [47], [48] are purely discriminative: The features (SIFT or image patches) are directly used for SVM classification with well-suited kernels. In particular, for [47], the training requires manual annotation of nine semantic concepts for 60,000 patches making the preprocessing step rather expensive. The unsupervised approach of Bosch et al. [44] trains a single pLSA model for all the classes, and then uses the marginal distribution $P(topic|document)$ as the input for a discriminative classifier, thus employing a hybrid (**staged**) technique. Finally, we also considered the semi-supervised generative approach of FeiFei and Perona [43], which makes use of LDA likelihood for classification.

6.2.2 Experiment 4—Scene Recognition Using Various Discriminative Methods in FESS

Obviously, a number of discriminative methods can be utilized to design a classifier based on the features extracted from the free energies under a set of previously learned generative models. As discussed above, if linear discriminant functions are adopted, the sum of the pieces of free energy f_i^t will be reweighted by a set of weights $\{w_i\}$. For example, if we employ a logistic regression, we can estimate a set of weights w_i, and classify using the sigmoid function:

$$P(x) = \frac{1}{1 + e^{(-\beta_1 + \sum_i \beta_i \cdot f_i^t)}}.$$

It is especially interesting to impose sparsity so that only few $\beta_i \neq 0$, i.e., only some free energy pieces will be taken into account for classification. This can be done efficiently by adding L1 regularization term for the weights of the logistic regressor to the optimization criterion.

As experiments we consider the **VS** and **FP** data sets and we apply several discriminative methods. Results are reported in Table 9, where LDF stands for linear discriminant functions, LR for logistic regression, S-LR stands for sparse logistic regression, and L-SVM stands for linear support vector machine. For how the score space is built, each of them outperforms generative classification (λ_{LDA}).

TABLE 9
Scene Classification: Generative Classification and Various Discriminative Methods in FESS

Dataset	λ_{LDA}	*LDF*	*LR*	*S-LR*	*L-SVM*
VS (6)	52.50% [49]	67.33%	76.25%	76.25%	83.71%
FP (13)	57.11%	64.1%	74.04%	80.84%	85.62%

6.2.3 Experiment 5—Using the Gradient as Score Operator, *gFESS*

In this final test, we focus only on the OT data set. Although we find that **FESS** outperforms the previously studied score spaces that depend on the derivatives, its use as score operator for **FESS** is, of course, possible. This allows for the construction of kernels similar to **FK** and **TK**, but derived from intractable generative models like latent LDA. In Table 10, we report the score spaces based on free energy; we refer to the score space defined by the free energy as score argument, and to the gradient as score operator with $\hat{F}_\nabla$ as gradient-FESS (**gFESS**).

In Table 10, **FESS** can be defined by the decomposition in entropy and cross entropy (first level of detail, $\hat{F}_{L_1}$), by the unique factorization properties of the network and $\mathcal{Q}$ (second level of detail, $\hat{F}_{L_2}$), or by considering the values each hidden variable can assume (third level of detail, $\hat{F}_{L_3}$), as shown in Table 2.

nFESS is defined by the decomposition performed by the score operator $\hat{F}_*$, which can be illustrated using the "tabular" notation previously described (see Tables 3 and 2).

gFESS uses the gradient as score operator and it corresponds to the Fisher score [6] only if the likelihood of the generative model upon which it is built is tractable; this does not hold for LDA, where approximate learning algorithms have to be used and $\mathcal{L} = \log P(X|\theta) < \mathcal{F}$, and the Fisher score is not computable.

The derivatives of the free energy of LDA with respect to the parameters β_{ij} (word-topic distribution) and α (Dirichlet parameters on topic) are

6. LDA performances were found to be slightly inferior to pLSA.

TABLE 10
Score Spaces Based on Free Energy

Score argument f	Score operator $\hat{F}$	Score mapping φ	Name
Free Energy $\mathcal{F}$	$\hat{F}_{L_i}$	$\varphi^{FESS}_{\hat{F}}(x^{(t)}) = \left[\mathcal{F}_{(\mathcal{Q},\hat{\theta})}(x^{(t)})\right]_{L_i}$	**FESS**
Free Energy $\mathcal{F}$	Table $\hat{F}_*$	$\varphi^{nFESS}_{\hat{F}}(x^{(t)}) = \left[\mathcal{F}_{(\mathcal{Q},\hat{\theta})}(x^{(t)})\right]_{\hat{F}_*}$	**nFESS**
Free Energy $\mathcal{F}$	∇_{θ_i}	$\varphi^{FESS}_{\hat{F}}(x^{(t)}) = \left[\nabla_{\theta_i}\mathcal{F}_{(\mathcal{Q},\hat{\theta})}(x^{(t)})\right]_{i=1}^{p}$	**gFESS**

*Since **nFESS** generalizes the family **FESS**, one can refer to both simply as **FESS**.*

$$\frac{\partial \mathcal{F}_{LDA}}{\partial \alpha_i} = \Psi\left(\sum_j \alpha_j\right) - \Psi(\alpha_i) + \Psi\left(\theta_i^{(t)}\right) - \Psi\left(\sum_j \theta_j^{(t)}\right),$$
$$\frac{\partial \mathcal{F}_{LDA}}{\partial \beta_{ij}} = \sum_{n=1}^{K(t)} \frac{Q\left(z_n^{(t)}\right) \cdot w_n^{(t)}}{\beta_{ij}}, \tag{28}$$

where $\theta^{(t)}$s define the mixture of topics that characterize the documents.

Classification results on the OT data set are reported in Table 11. We found that $\hat{F}_{LDA}$ outperforms $\hat{F}_{\nabla}$, and this is due to the fact that the entropy terms do not depend on θ while the gradient sets them to zero, making the resulting score space less expressive.

7 Discussion and Conclusions

In this paper, we present a novel generative score space, **FESS**, exploiting variational free energy terms as features. The additive free energy terms arise naturally as a consequence of the factorization of the model P and the posterior Q. We show that the use of these terms as features in discriminative classification leads to more robust results than the use of the Fisher scores, which are based on the derivatives of the loglikelihood of the data with respect to the model parameters. As has been previously observed, we find that the Fisher score space suffers from the so-called "wrap-around" problem, where very different data points may map to the same derivative (an example was discussed in the introduction). On the other hand, free energy terms quantify the data fit in different parts of the model, and are informative even when the model is imperfect. This indicates that the rescaling of these terms, carried out by the subsequent discriminative training, in some way leads to improved modeling of the data. Scaling a term in the free energy composition, e.g., the term $\sum_h Q(h) \log P(x|h)$, by a constant w is equivalent to raising the appropriate conditional distribution to the power w. This is indeed reminiscent of some previous approaches to correcting generative modeling problems. In speech applications, for example, it is a standard practice to raise the observation likelihood in HMMs to a power less than 1, before inference is performed on the test sample, as the acoustic signal would otherwise overwhelm the hidden process modeling the language constraints [50]. This problem arises from the approximations in the acoustic model. For instance, a high-dimensional acoustic observation is often modeled as following a diagonal Gaussian distribution, thus assuming independent noise in the elements of the signal, even though the true acoustics of speech is far more constrained. This results in overaccounting for the variations in the observed acoustic signal, and to correct for this in practice, the log probability of the observation given the hidden variable is scaled down.

The technique described here proposes a way to automatically infer the best scaling, but it also goes a step further in allowing for such corrections at all levels of the model hierarchy, and even for specific configurations of hidden variables. Furthermore, the use of kernel methods provides for nonlinear corrections too. This extremely simple technique is shown here to work remarkably well, outperforming previous score space approaches as well as the state of the art in several diverse applications.

TABLE 11
Scene Classification Results
Using Gradient as Score Operator

Dataset	λ_{LDA}	$\hat{F}_{LDA}$	$\hat{F}_{\nabla}$ (**gFESS**)	[48]
$\mathbf{OT}^N$ (4)	63.93%	**95.21**%	90.10%	84.51%
$\mathbf{OT}^A$ (4)	67.21%	**94.38**%	90.32%	89.43%

Acknowledgments

The authors acknowledge financial support from the FET programme within the EU FP7, under the SIMBAD project (contract 213250).

References

[1] A.Y. Ng and M.I. Jordan, "On Discriminative vs. Generative Classifiers: A Comparison of Logistic Regression and Naive Bayes," *Proc. Advances in Neural Information Processing Systems 14,* pp. 841-848, 2001.

[2] G. Bouchard and B. Triggs, "The Trade-Off between Generative and Discriminative Classifiers," *Proc. 16th IASC Symp. Computational Statistics,* pp. 721-728, 2004.

[3] S. Kapadia, "Discriminative Training of Hidden Markov Models," PhD dissertation, Univ. of Cambrdige, 1998.

[4] J.A. Lasserre, C.M. Bishop, and T.P. Minka, "Principled Hybrids of Generative and Discriminative Models," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition,* pp. 87-94, 2006.

[5] A. Mccallum, C. Pal, G. Druck, and X. Wang, "Multi-Conditional Learning: Generative/Discriminative Training for Clustering and Classification," *Proc. 21st Nat'l Conf. Artificial Intelligence,* pp. 433-439, 2006.

[6] T. Jaakkola and D. Haussler, "Exploiting Generative Models in Discriminative Classifiers," *Proc. Advances in Neural Information Processing Systems 11,* pp. 487-493, 1998.

[7] N. Smith and M. Gales, "Speech Recognition Using SVMs," *Proc. Advances in Neural Information Processing Systems 15,* pp. 1197-1204, 2002.

[8] K. Tsuda, M. Kawanabe, G. Rätsch, S. Sonnenburg, and K.-R. Müller, "A New Discriminative Kernel from Probabilistic Models," *Neural Computation,* vol. 14, no. 10, pp. 2397-2414, 2002.

[9] T. Jaakkola, M. Meila, and T. Jebara, "Maximum Entropy Discrimination," *Proc. Advances in Neural Information Processing Systems 12,* pp. 470-476, 1999.

[10] O. Yakhenko, L.V. Lita, R. Rosales, and S. Niculescu, "Principled Generative-Discriminative Hybrid Hidden Markov Model," *Proc. NIPS Workshop Representations and Inference on Probability Distributions,* 2007.

[11] A. Fujino, N. Ueda, and K. Saito, "Semisupervised Learning for a Hybrid Generative/discriminative Classifier Based on the Maximum Entropy Principle," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 30, no. 3, pp. 424-437, Mar. 2008.

[12] G. Hinton, P. Dayan, B. Frey, and R. Neal, "The Wake-Sleep Algorithm for Unsupervised Neural Networks," *Science,* vol. 268, pp. 1158-1161, 1995.

[13] C. Sminchisescu, A. Kanaujia, and D. Metaxas, "Learning Joint Top-Down and Bottom-Up Processes for 3D Visual Inference," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition,* pp. 1743-1752, 2006.

[14] A. Bosch, A. Zisserman, and M. Xavier, "Scene Classification Using a Hybrid Generative/Discriminative Approach," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 30, no. 4, pp. 712-727, Apr. 2008.

[15] T. Jebara, R. Kondor, A. Howard, K. Bennett, and N. Cesa-bianchi, "Probability Product Kernels," *J. Machine Learning Research,* vol. 5, pp. 819-844, 2004.

[16] X. Li, T.S. Lee, and Y. Liu, "Hybrid Generative-Discriminative Classification Using Posterior Divergence," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition,* pp. 2713-2720, 2011.

[17] T. Minka, "Discriminative Models, Not Discriminative Training," Technical Report TR-2005-144, Microsoft Research Cambridge, 2005.

[18] D.-Q. Zhang and S.-F. Chang, "A Generative-Discriminative Hybrid Method for Multi-View Object Detection," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition,* pp. 2017-2024, 2006.

[19] R. Fergus, P. Perona, and A. Zisserman, "Object Class Recognition by Unsupervised Scale-Invariant Learning," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition,* pp. 264-271, 2003.

[20] A. Epshteyn and G. DeJong, "Generative Prior Knowledge for Discriminative Classification," *J. Artificial Intelligence Research,* vol. 27, no. 1, pp. 25-53, 2006.

[21] C. Weber, S. Wermter, and M. Elshaw, "A Hybrid Generative and Predictive Model of the Motor Cortex," *Neural Networks,* vol. 19, no. 4, pp. 339-353, 2006.

[22] R. Rosales and S. Sclaroff, "Combining Generative and Discriminative Models in a Framework for Articulated Pose Estimation," *Int'l J. Computer Vision,* vol. 67, pp. 251-276, May 2006.

[23] T. Hofmann, "Probabilistic Latent Semantic Indexing," *Proc. 22nd Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval,* pp. 50-57, 1999.

[24] R. Raina, Y. Shen, A.Y. Ng, and A. McCallum, "Classification with Hybrid Generative/Discriminative Models," *Proc. Advances in Neural Information Processing Systems 16,* pp. 12-19, 2004.

[25] A. Subramanya, Z. Zhang, A. Surendran, P. Nguyen, M. Narasimhan, and A. Acero, "A Generative-Discriminative Framework Using Ensemble Methods for Text-Dependent Speaker Verification," *Proc. IEEE Int'l Conf. Acoustics, Speech, and Signal Processing,* pp. 225-228, 2007.

[26] M. Bicego, V. Murino, and M. Figueiredo, "Similarity-Based Clustering of Sequences Using Hidden Markov Models," *Proc. Third Int'l Conf. Machine Learning and Data Mining in Pattern Recognition,* P. Perner and A. Rosenfeld, eds., pp. 86-95, 2003.

[27] A.D. Holub, M. Welling, and P. Perona, "Combining Generative Models and Fisher Kernels for Object Class Recognition," *Proc. IEEE Int'l Conf. Computer Vision,* pp. 136-143, 2005.

[28] A. Perina, P. Lovato, M. Cristani, and M. Bicego, "A Comparison on Score Spaces for Expression Microarray Data Classification," *Proc. Sixth IAPR Int'l Conf. Pattern Recognition in Bioinformatics,* pp. 12-28, 2011.

[29] N. Jojic, J. Winn, and L. Zitnick, "Escaping Local Minima through Hierarchical Model Selection: Automatic Object Discovery, Segmentation, and Tracking in Video," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition,* pp. 117-124, 2006.

[30] A. Perina, M. Cristani, U. Castellani, V. Murino, and N. Jojic, "Free Energy Score Space," *Proc. Advances in Neural Information Processing Systems 22,* pp. 1428-1436, 2009.

[31] R.M. Neal and G.E. Hinton, "A View of the EM Algorithm that Justifies Incremental, Sparse, and Other Variants," *Learning in Graphical Models,* M.I. Jordan, ed., pp. 355-368, MIT Press, 1999.

[32] M.I. Jordan, Z. Ghahramani, T. Jaakkola, and L.K. Saul, "An Introduction to Variational Methods for Graphical Models," *Machine Learning,* vol. 37, no. 2, pp. 183-233, 1999.

[33] H.J. Kappen and W.J. Wiegerinck, "Mean Field Theory for Graphical Models," *Adavanced Mean Field Theory: Theory and Practice,* M. Opper and D. Saad, eds., pp. 37-49, MIT Press, 2001.

[34] Z. Ghahramani, "On Structured Variational Approximations," Technical Report CRG-TR-97-1, Univ. of Cambridge, 1997.

[35] B. Frey and N. Jojic, "A Comparison of Algorithms for Inference and Learning in Probabilistic Graphical Models," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 27, no. 9, pp. 1392-1413, Sept. 2005.

[36] N. Smith and M. Gales, "Using SVMs to Classify Variable Length Speech Patterns," Technical Report CUED/F-INGENF/TR.412, Univ. of Cambridge, 2002.

[37] L.R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications In Speech Recognition," *Proc. IEEE,* vol. 77, no. 2, pp. 257-286, Feb. 1989.

[38] D. MacKay, "Ensemble Learning for Hidden Markov Models," technical report, Univ. of Cambridge, 1997.

[39] D. Blei, A. Ng, and M.I. Jordan, "Latent Dirichlet Allocation," *J. Machine Learning Research,* vol. 3, pp. 993-1022, 2003.

[40] G.G. Towell, J.W. Shavlik, and M.O. Noordewier, "Refinement of Approximate Domain Theories by Knowledge-Based Neural Networks," *Proc. Eighth Nat'l Conf. Artificial Intelligence,* pp. 861-866, 1990.

[41] A. Frank and A. Asuncion, "UCI Machine Learning Repository," http://archive.ics.uci.edu/ml, 2010.

[42] J.C. Huang, A. Kannan, and J. Winn, "Bayesian Association of Haplotypes and Non-Genetic Factors to Regulatory and Phenotypic Variation in Human Populations," *Bioinformatics,* vol. 23, no. 13, pp. 212-221, 2007.

[43] L. FeiFei and P. Perona, "A Bayesian Hierarchical Model for Learning Natural Scene Categories," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition,* pp. 524-531, 2005.

[44] A. Bosch, A. Zisserman, and X. Munoz, "Scene Classification via Plsa," *Proc. European Conf. Computer Vision,* pp. 517-530, 2006.

[45] M. Girolami and A. Kabán, "On an Equivalence between PLSI and LDA," *Proc. 26th Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval,* pp. 433-434, 2003.

[46] A. Oliva and A. Torralba, "Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope," *Int'l J. Computer Vision,* vol. 42, no. 3, pp. 145-175, 2001.

[47] J. Vogel and B. Schiele, "Semantic Modeling of Natural Scenes for Content-Based Image Retrieval," *Int'l J. Computer Vision,* vol. 72, no. 2, pp. 133-157, 2007.

[48] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition,* pp. 2169-2178, 2006.

[49] A. Perina, M. Cristani, and V. Murino, "Learning Natural Scene Categories by Selective Multi-Scale Feature Extraction," *Image and Vision Computing,* vol. 28, no. 6, pp. 927-939, 2010.

[50] L. Deng and D. O'Shaughnessy, *Speech Processing: A Dynamic and Optimization-Oriented Approach.* Marcel Dekker, Inc., 2003.

Alessandro Perina received the PhD degree in computer science from the University of Verona with a thesis on classification with generative models. From 2006 to 2010, he was a member of the Vision, Image Processing, and Sound group (VIPS) at the University of Verona. He is now a postdoctoral researcher at Microsoft Research, Redmond, Washington, working with the eScience group. His research interests are in computer vision and machine learning.

Marco Cristani has been an assistant professor since 2007 at the University of Verona, and since 2009 he has also been a team leader at the Istituto Italiano di Tecnologia, Genova, Italy. He is currently a scientific collaborator in national and European projects. His research interests regard statistical pattern recognition, applied to video surveillance and social signaling. He is a member of the IEEE, ACM, and IAPR.

Umberto Castellani received the PhD degree in computer science from the University of Verona in 2003 working on 3D data modeling and reconstruction. He is an assistant professor at the University of Verona. His research is focused on 3D data processing, statistical learning, and medical image analysis. He has coauthored more than 50 papers published in leading conference proceedings and journals. He is a member of Eurographics, IAPR, and the IEEE.

Vittorio Murino received the PhD degree in electronic engineering and computer science in 1993 from the University of Genova, Italy. Then, he was first at the University of Udine and, since 1998, at the University of Verona, where he served as chairman of the Department of Computer Science from 2001 to 2007. He is a full professor and head of the Computer Imaging facility (PLUS laboratory) at the Istituto Italiano di Tecnologia, Genova, Italy. His research interests are in computer vision and machine learning, in particular, probabilistic techniques for image and video processing, with applications on video surveillance, biomedical image analysis, and bioinformatics. He is a senior member of the IEEE.

Nebojsa Jojic received the PhD degree from the University of Illinois at Urbana-Champaign in 2001, where he received a Microsoft Fellowship in 1999 and a Robert T. Chien Excellence in Research award in 2000. He has been a researcher at Microsoft Research in Redmond, Washington, since 2000. He has published more than 100 papers in the areas of computer vision, machine learning, signal processing, computer graphics, and computational biology.