MULTISCALE MODEL. SIMUL.
Vol. 8, No. 5, pp. 1784–1802

CLUSTERING AND CLASSIFICATION THROUGH NORMALIZING FLOWS IN FEATURE SPACE*

J. P. AGNELLI†, M. CADEIRAS‡, E. G. TABAK§, C. V. TURNER†, AND E. VANDEN-EIJNDEN§

Abstract. A unified variational methodology is developed for classification and clustering problems and is tested in the classification of tumors from gene expression data. It is based on fluid-like flows in feature space that cluster a set of observations by transforming them into likely samples from p isotropic Gaussians, where p is the number of classes sought. The methodology blurs the distinction between training and testing populations through the soft assignment of both to classes. The observations act as Lagrangian markers for the flows, comparatively active or passive depending on the current strength of the assignment to the corresponding class.

Key words. maximum likelihood, expectation maximization, Gaussianization, machine learning, density estimation, inference

AMS subject classifications. 62G07, 62H30

DOI. 10.1137/100783522

1. Introduction.

Some classification problems require few input variables. In clinical diagnosis, for example, patients with active infections may be easily identified by the sole presence of an elevated count of white blood cells in the circulation, while patients with advanced liver failure can be detected by measuring the ammonia levels in the blood. If data from a large enough *training* population are available, these variables can be calibrated so as to produce the desired classification when a new *testing* sample becomes available. In the simplest scenario, this calibration may produce a table of possible ranges of the combined input variables—or *features*—from which to read off the output, or a formula for the output, with parameters fitted to the training data. A more thorough approach may be to estimate a probability density in the space of features associated with each class and use it to infer the likelihood that the new sample belongs to each of the classes [1].

Yet such a combination of few required input variables and large training populations is often unavailable. This is the case when one studies exceptional situations, such as a rare illness, when observations are difficult or costly, and when the fundamental causes underlying the separation into classes—which presumably link the classes to a few well-defined observables—are not thoroughly understood, as is often the case in complex systems: the human body, our planet's climate, the financial world. Modern technology offers us a way to compensate for this lack of understanding: the possibility to monitor not a few, but myriads of variables loosely associated with the classification sought, such as the expression level of genes, the sea-surface

*Received by the editors January 21, 2010; accepted for publication (in revised form) July 6, 2010; published electronically October 14, 2010.
http://www.siam.org/journals/mms/8-5/78352.html
†FaMAF, U. N. Córdoba, Cordoba, Argentina (agnelli@famaf.unc.edu.ar, turner@famaf.unc.edu.ar). The work of these authors was partially supported by grants from CONICET, SECYT-UNC, and PICT-FONCYT.
‡Department of Medicine, Division of Cardiology, University of Alabama at Birmingham, Birmingham, AL (mcadeiras@cardmail.dom.uab.edu).
§Courant Institute, New York University, New York, NY 10012 (tabak@cims.nyu.edu, eve2@cims.nyu.edu). The work of these authors was partially supported by grants from the NSF's Division of Mathematical Sciences.

temperature throughout the globe, and the prices of options. Presumably, we can balance with quantity of data the missing well-defined causal links to a few.

There is a problem though. Consider, for instance, the simplest procedure of compiling a table. Since the number of entries required grows exponentially with the dimension of the feature space, the size of the training population required to fill the table soon becomes unrealistic (billions of patients, of temperature monthly averages, of daily returns). In building a parametric formula, this problem translates into overfitting: one can always find parameters that provide a perfect fit to the training population, but which may yield meaningless results on new samples.

Similarly, for density estimation, one needs a number m of training samples that grows exponentially with the dimension n of the space; this is the "curse of dimensionality." For simple illustration, consider the situation when $m < n$. Since all training points lie on a hyperplane of dimension m, they do not tell us how to estimate the probability density away from this hyperplane. Yet, with probability one, when a testing sample arrives, it will not lie on the hyperplane; how can one then say anything meaningful about its likelihood? The planar geometry of this illustration is not essential: the crucial point is that almost all points in phase space are far from the observations—"not in the table"—so their estimated density is not reliable.

This problem seems insurmountable; yet there is something counterintuitive about it. Since the classification problem appears challenging, we gather more information; can this extra information hurt us? Attempting to diagnose a patient's ailment, we are provided with extra clinical results: should we throw them away unread, lest they confound our judgment? Clearly, if there is a problem, it resides not in the availability of additional data, but in our way of handling it. In this article, we propose to bypass this problem through a general methodology for classification and clustering.

The general principle is quite straightforward. We worry that the testing points may lie far from the training observations, and so their probability density in each class may be poorly estimated. Yet we have observations lying precisely on the testing points: the testing points themselves! Of course, we do not know a priori to which class they belong—that is what we would like to unfold from the data—but we do know that they belong to one of them. This knowledge can be used to provide a robust classification scheme. Such use of unlabeled data for classification lies at the core of semisupervised learning [2] and transductive inference [12]. This methodological direction blurs the distinction between the training and testing populations. One can pursue this idea further to establish a general methodology that does not distinguish between problems in classification and in clustering; the latter do not have a training population at all.

Our proposal, which builds on a density estimation algorithm developed in [11], is based on a set of fluid-like flows that transform the observations into realizations of p isotropic Gaussian processes. The flows use the observations as active Lagrangian markers, which guide the descent of the Kullback–Leibler divergence [10] between the current distributions and the target Gaussians. All observations guide all p flows, but, as some become more firmly assigned to individual classes, they become more active in the flows associated with these, while behaving more and more as passive Lagrangian markers for the others. This procedure allows us to integrate the expectation-maximization methodology into a natural descent framework.

Many of the topics discussed in this paper have points in common with themes in the literature; our contribution provides novel ingredients, but also a unified methodology and viewpoint. A central role is played by the expectation maximization (EM) framework [4], for which we provide an alternative derivation in classification-

clustering settings. The Gaussianization procedure for density estimation was originally developed in [11], but shares some traits with one developed in [3], in the general context of exploratory projection pursuit [6]. Variable selection is a broad field (see [8] for a general review); we use a cluster assessment criterion for variable selection that fits loosely within those based on information theory [5]. Our main innovation is the use of smooth, gradual flows as a clustering technique.

The paper is structured as follows. After this introduction, section 2 presents a general, unified formulation of classification and clustering in the EM framework. Section 3 introduces the flows in feature space, the centerpiece of the methodology proposed. Section 4 extends the methodology to cluster assessment, with focus on its application to the selection of observables for classification. Section 5 illustrates the procedures discussed in the context of a medical application, the classification of tumors, using data from two published sources: one concerning small round blue cell tumors of childhood [9], and the other two classes of acute leukemia [7]. Finally, section 6 closes the paper with some concluding remarks.

2. Clustering and classification: A unified formulation. The clustering problem consists of the following: given a matrix Z of m observations z^j of n variables z_i, one is asked to partition the observations into p clusters C_k with common traits.

By contrast, in a classification problem, one is asked to assign testing observations y^j to the class C_k, $k \in (1, \ldots, p)$, with which each has the most traits in common. To identify these traits, one is told the classes to which a set of training observations x^j belong. A prior belief π_k^j on the attribution of each testing observation may be provided as additional input; π_k^j represents the probability, before observing any feature, that the jth sample belongs to the kth class.

The classification problem can be generalized and softened, regarding both the goal sought and the input required. One may seek, instead of a rigid assignment, a probability p_k^j that the testing observation y^j belong to the class C_k. Also, one may be provided with just a soft assignment of the training population: the probability p_k^j that x^j is in C_k. The clustering problem can be generalized in a similar way: one may seek a soft partition, in which each observation z_j has probability p_k^j of belonging to the cluster C_k.

It should be clear at this point that the two problems, clustering and classification, in their generalized formulation, can be posed in a unified way: given a matrix X of m observations x^j of n variables x_i, and, for a subset J_{train} of the observations, the probability p_k^j that the observation x^j is in class C_k, one seeks the corresponding *posterior* probabilities p_k^j for the remaining observations, $j \in J_{\text{test}}$, for which we only have a prior, π_k^j. The only difference between the two problems in this formulation is that, for pure clustering, J_{train} is empty.

Notice too that one recovers the "hard" version of the two problems if the training observations have probabilities p_k^j that are either zero or one; a rule is established to assign a testing observation x^j to a class, such as choosing the class C_k with maximal p_k^j.

2.1. Clustering and classification through density estimation. How can one characterize the "common traits" that define each class C_k? The most natural and general way is through a probability density $\rho_k(x)$, which specifies how likely it is to find a sample with observables x in the class C_k. Given one such probability density for each class, the posterior probability p_k^j that the observation x^j belongs to

the class C_k follows from Bayes' formula,

$$p_k^j = \frac{\pi_k^j \, \rho_k(x^j)}{\sum_q \pi_q^j \, \rho_q(x^j)} \, . \tag{1}$$

Assume that we are in possession of a density-estimation algorithm that, given a set of m observations y^j of n variables, produces an estimate for the underlying probability density $\rho(y)$. The way density estimation is usually applied to classification problems involves estimating the distributions $\rho_k(x)$ from the training data, and then applying (1) to each member x^j of the testing population, to infer the probability that it belong to each class k.

Yet this procedure does not use all the information at our disposal. This leads to problems when, as is often the case, one has observations of many variables and a relatively small training population. (For instance, in microarray-based diagnosis, one may have records of the expression level of tens of thousands of genes, from a training population of a few hundred patients.) In classical procedures, such as linear regression, this yields the problem of overfitting: with so many variables at one's disposal, one can produce a perfect fit of the training data, yet obtain poor results on the tests.

In procedures based on density estimation, this problems manifests itself as undersampling: one needs to estimate a probability density in a high-dimensional space from only a handful of observations. Clearly, any such density estimation is necessarily poor. When the testing observations become available, they are likely to be far from all training observations, and hence assigned an inaccurate probability density in each class.

Rephrasing this, we say that a problem arises in density-based classification, because the probability densities of some or all classes may be underresolved at the testing points due to the lack of training samples nearby. Yet we do have information located precisely at the testing points: the testing samples themselves! We do not know to which class they belong—that's precisely the point of the classification exercise—but we do know that they belong to one class. Hence at least one of the p distributions ρ_k should be nonzero at each testing point.

Consider density-estimation algorithms based on the maximization of the likelihood of the data,

$$\rho = \arg \left(\max_{\rho} \left(L[\rho] \right) \right) \, ,$$

where L is the logarithm of the likelihood function,

$$L[\rho] = \sum_{j=1}^{m} \log \left(\rho \left(y^j \right) \right) \, , \tag{2}$$

and the maximization is carried over a proposed set of permissible distributions $\rho(y)$. The standard procedure would perform this maximization over each class in the training population, and then infer the probabilities for the testing population from Bayes' formula (1). For each class, we would maximize

$$L_k = \sum_{\text{training}}^{j \in C_k} \log(\rho_k(x^j)). \tag{3}$$

Yet this likelihood function does not take into the account the testing observations, in particular, the fact that each must belong to one of the classes. Since the probability density for a testing observation x^j is

$$\rho(x^j) = \sum_k \pi_k^j \, \rho_k(x^j) \,, \tag{4}$$

where π_k^j is the prior probability that the jth sample belongs to the kth class, the complete log-likelihood for all observations involves now a sum over all classes,

$$L = \sum_{\text{training}} \sum_k p_k^j \log(\rho_k(x^j)) + \sum_{\text{testing}} \log\left(\sum_k \pi_k^j \, \rho_k(x^j) \right) , \tag{5}$$

where p_k^j is one if the jth training observation belongs to the class k, and zero otherwise.

Consider the derivative of the testing component of the likelihood function with respect to $\rho_k^j = \rho_k(x^j)$:

$$\frac{\partial}{\partial \rho_k^j} \log\left(\sum_k \pi_k^j \, \rho_k^j \right) = \frac{\pi_k^j}{\sum_k \pi_k^j \, \rho_k^j} = \frac{p_k^j}{\rho_k^j} \,, \tag{6}$$

where p_k^j is the posterior probability from (1). Notice that this is the same as the partial derivative of the weighted log-likelihood

$$\sum_k p_k^j \, \log(\rho_k^j) \tag{7}$$

if the posteriors p_k^j are kept fixed. Then the complete log-likelihood (5) has the same partial derivatives with respect to the densities as the sum

$$L = \sum_k L_k \,, \quad \text{where} \quad L_k = \sum_j p_k^j \, \log(\rho_k(x^j)) \,, \tag{8}$$

where the only difference between the training and testing populations is that the priors π_k^j of the former are typically far more biased, possibly all the way to Kronecker deltas, as in the derivation above.

Initially, the probabilities p_k^j can be taken equal to the priors π_k^j. Maximizing the L_k's with fixed p_k^j gives rise to a set of estimated densities $\rho_k^0(x)$. Then, in an expectation maximization (EM) approach, we can iterate the procedure with the probabilities p_k^j now given by the posterior from (1), and hence update the $\rho_k^t(x)$'s into $\rho_k^{t+1}(x)$'s, until convergence. Notice that this procedure remains unchanged when the training population is only softly assigned to their classes; then the provided p_k^j are not Kronecker deltas, but more general discrete probabilities. Finally, the procedure applies also when the training population is empty, yielding a methodology for clustering.

To fix ideas, we re-enunciate the procedure more formally below. If a parametric density estimation procedure is provided, then, given

- a matrix X of m observations x^j of n variables x_i,
- a number p of classes C_k,
- a prior probability π_k^j that each observation j belongs to class C_k,

- a family of probability distributions $\rho(x;\alpha)$,

we perform an iterative procedure that computes, at each step $t \geq 0$, an estimated probability $P_k^j[t]$ that each observation j is in class C_k, and, for $t > 0$, a set of estimated probability densities $\rho_k^t(x)$ describing each class C_k, as follows:

- Set the initial probabilities at their prior values, $P_k^j[0] = \pi_k^j$.
- For all steps $t > 0$,
  - compute $\rho_k^t(x)$ by maximizing over the parameters α the expected value of the log-likelihood,

$$L_k = \sum_j P_k^j[t-1] \, \log(\rho_k^t(x^j)) \,, \quad \rho_k^t(x) \in \rho(x;\alpha) \,, \tag{9}$$

    for each class C_k;
  - update the probabilities P_k^j through Bayes' formula,

$$P_k^j[t] = \frac{\pi_k^j \, \rho_k^t(x^j)}{\sum_q \pi_q^j \, \rho_q^t(x^j)} \,, \tag{10}$$

    until a convergence criterion is satisfied.

(When performing pure clustering—i.e., when the priors π_k^j do not depend on j—we need to break the symmetry between classes in order to start the algorithm. This can be achieved by making some of the initial assignments $P_k^j[0]$ slightly different for the various samples j.)

3. A normalizing flow. The procedure above seeks, for each class k, a fit to a parametric family of distributions, $\rho_k(x;\alpha)$. Here we propose an alternative, in which each of these distributions is characterized by a map $y_k(x)$ and a common target distribution $\mu(y)$, so that

$$\rho_k(x) = J_k(x) \, \mu(y_k(x)) \,, \tag{11}$$

where $J_k(x)$ is the Jacobian of the map $x \to y_k$. If the maps $y_k(x)$ are described in terms of a set of parameters α, this appears to be just a convoluted rewriting of the parametric proposal $\rho_k(x;\alpha)$. Yet we shall see that not only is such a rewriting natural, giving rise to a geometric, "dual" view of the clustering-classification problem, but also that it provides a full class of novel, effective algorithms for implementing its solution.

The duality comes about from looking at the proposal (11) from two alternative perspectives: given a sample of x, we seek either the density $\rho(x)$ that best represents it or the map $y(x)$ that best transforms it into a sample from the known density $\mu(y)$. We have developed such an approach to density estimation in [11]; in the classification-clustering context of this article, each map $y_k(x)$ acquires an extra degree of signification, as it either "absorbs" or "rejects" each observation into the geometric cluster of its corresponding class.

We may, as in [11], introduce an "algorithmic time" t and think of the maps $y_k(x)$ as terminal points of flows $z_k(x;t)$, with

$$z_k(x;0) = x, \quad z_k(x;\infty) = y_k(x).$$

At each time t, we have a current estimate for the probability density in each class,

$$\rho_k^t(x) = J_k(x;t) \, \mu(z_k(x;t)) \,. \tag{12}$$

These densities, in turn, determine the soft assignments $P_k^j[t]$ from (10), and hence the compounded log-likelihood

$$L = \sum_{k,j} P_k^j[t] \, \log(\rho_k^t(x^j;t)) \, . \tag{13}$$

Following the EM-like algorithm of the prior section, one could, in each time step, maximize L over the parameters in the densities, with the assignments $P_k^j[t]$ fixed, and then update these using Bayes' formula. Yet, because the densities are defined by flows $z_k(x;t)$, we can switch from discrete to continuous times t and evolve the flows through their corresponding velocity fields $u_k = \frac{\partial}{\partial t} z_k(x;t)$, computed by ascent of the log-likelihood L:

$$u_k \propto \frac{\delta L}{\delta u_k},$$

where the variations are taken with P_k^j fixed, from the argument in the previous section extended to the continuous scenario.

In the presence of infinitely many observations, the log-likelihood (13) adopts the form

$$L = \sum_k \int P_k(x;t) \, \log(\rho_k^t(x;t)) \, \rho_k(x) \, dx, \tag{14}$$

where $\rho_k(x)$ is the actual probability density for the class k evaluated at the point x, $\rho_k^t(x;t)$ is given by (12), $P_k(x;t)$ by

$$P_k(x;t) = \frac{\pi_k(x) \, \rho_k^t(x)}{\sum_q \pi_q(x) \, \rho_q^t(x)} \, , \tag{15}$$

and the flow $z_k(x;t)$ satisfies the system of integrodifferential equations

$$\frac{\partial}{\partial t} z_k(x;t) = \frac{\delta L}{\delta z_k} \, , \tag{16}$$

where again the variations are taken with $P_k(x;t)$ fixed:

$$\frac{\delta L}{\delta z_k} = P_k(z_k) \, J_k(x) \left(\frac{\nabla_{z_k} \mu(z_k)}{\mu(z_k)} \rho_k(z_k) - \nabla_{z_k} \rho_k(z_k) \right) . \tag{17}$$

Here

$$\rho_k(z_k(x)) = \frac{\rho_k(x)}{J_k(x)}$$

is the current probability density of the z_k's and $P_k(z_k)$ is shorthand for $P_k(x)$, with $z_k = z_x(x)$.

In this expression, it is only the Jacobian $J_k(x)$ that keeps track of the original observation x where the flow started. Here is where the dual view of the flow becomes useful: if, instead of performing density estimation, i.e., finding $\rho_k(x)$, we were normalizing x through a map $z_k(x;t)$ converging to a $y_k(x)$ with μ-statistics, there would be no need, at each time, to remember with which x we began: the present values of z are all we need to continue deforming them into a sample of μ. Then, adopting

this dual view, we can remove the Jacobian $J_k(x)$ from the dynamics, turning the equations memoryless, with all times formally identical to $t = 0$.

The resulting algorithm is a blend of the normalizing ideas developed in [11] and the clustering and classification through EM and soft assignments of section 2. As before, we start with a matrix X of m observations x^j of n variables x_i, and a prior probability π_k^j that observation j belongs to class C_k, $k \in 1, \ldots, p$. Then we follow p flows $z_k(x;t)$ through the following procedure:

- **Preconditioning:** In a first, preconditioning step, one considers each class k, with all observations x^j softly assigned to it according to their prior π_k^j. Computing the weighted mean

$$\bar{x}_k = \frac{\sum \pi_k^j x^j}{\sum \pi_k^j}$$

and average standard deviation

$$\sigma_k = \sqrt{\frac{\sum \pi_k^j \|x^j - \bar{x}_k\|^2}{n \sum \pi_k^j}},$$

one obtains, for each flow, the particles' centered and normalized initial positions:

$$z_k^j = z_k(x^j; 0^+) = \frac{x^j - \bar{x}_k}{\sigma_k}\,. \tag{18}$$

The corresponding initial Jacobians of the flow are given by $J_k^j = \sigma_k^{-n}$.

- **Flow:** For all (discretized) steps $t > 0$, do the following:
  1. Compute the soft assignments,

$$P_k^j = \frac{\pi_k^j \rho_k^j}{\sum_q \pi_q^j \rho_q^j}\,, \tag{19}$$

where

$$\rho_k^j = J_k^j \mu(z_k^j)\,. \tag{20}$$

  2. Perform a normalizing step in each class. Following the procedure developed in [11], we propose for target distribution the isotropic Gaussian

$$\mu(y) = \frac{1}{(2\pi)^{\frac{n}{2}}} e^{-\frac{1}{2}|y|^2}.$$

Then each step starts with a random unitary transformation,

$$z_k^j \to U_k z_k^j,$$

followed by n one-dimensional, near-identity transformations (one per dimension i),

$$\begin{aligned} z_k^j(i) &\to F_k^i(z_k^j(i)) \\ J_k^j &\to \frac{d}{dz} F_k^i(z_k^j(i))\, J_k^j\,, \end{aligned}$$

that moves its marginal distribution toward Gaussianity. The transformation F_k^i is selected from a parametric family $F(z;\alpha)$, with $F(z;0)=z$, by ascent of the log-likelihood:

$$\alpha = \gamma \nabla_\alpha L_i \,|_{\alpha=0}\,, \tag{21}$$

where

$$\gamma = \frac{\epsilon}{\sqrt{\epsilon^2 + \|\nabla_\alpha L_i\|^2}}, \tag{22}$$

$$L_i(\alpha) = \sum_{j=1}^{m} P_k^j \left[\log\left|F_z\left(z_k^j(i);\alpha\right)\right| + \log\mu\left(F\left(z_k^j(i);\alpha\right)\right)\right]\,,$$

and $\mu(z)$ is the one-dimensional normal distribution. The choice of the learning rate ϵ is a matter of considerable interest permeating all descent procedures; in the examples below, we have made the simplest choice of adopting a constant $\epsilon = 0.02$.

The flows $z_k(x;t)$ and their Jacobian need only be computed on the observations x^j. If there are other points $\tilde{x}$ where the density ρ is sought, these can be carried passively by the algorithm, without affecting the log-likelihood. In regular classification, the x^j's are the observations in each class, and the $\tilde{x}$'s those in the testing population. In the unified framework for classification and clustering presented here, all observations constitute "active" markers for all classes, with their contributions to the log-likelihood corresponding to class k weighted by the probabilities P_k^j.

This is an effective algorithm that converges typically even faster than just one full density estimation with fixed attributions P_k^j. The reason for this bonus is also the algorithm's possible pitfall: making the attributions follow the evolution of the density estimates through (19) reinforces any bias that the estimation may have at its initial stages. Thus, the algorithm converges rapidly to a local maximum of the log-likelihood, even though this maximum may not be global. In particular, because of the Gaussian preconditioning step, the algorithm would be biased toward the attributions resulting from this very simple, linear preprocessor, thus potentially defeating the purpose of the more sophisticated, nonlinear procedure.

A way to reduce the effect of this original bias is to implement a more gradual version of the attribution to go alongside the density estimation by ascent, replacing (19) in step 1 above by

$$P_k^j \to P_k^j + \epsilon\left(\frac{\pi_k^j\,\rho_k^t(x^j;t)}{\sum_q \pi_q^j\,\rho_q^t(x^j;t)} - P_k^j\right)\,, \tag{23}$$

where ϵ is the same learning rate used for the ascent of the log-likelihood. This way, the soft attributions P_k^j have a relaxation time $\frac{1}{\epsilon}$ of the same order as the density estimation, so they will not instantly reinforce any bias associated with the initial guess for the density. Notice that with this modification the initial attributions $P_k^j[0]$ are no longer the result of the preprocessing step but equal the externally provided priors π_k^j. This is the procedure followed in the examples below.

4. Cluster assessment and variable selection. The methodology developed above, which follows flows $z_k(x;t)$ in phase space in order to softly assign a set of observations to classes, can be used to address the reciprocal problem: to assess how

well a set of variables x supports a given clustering, i.e., a distribution of a population into classes with probability q_k^j. This includes as a particular case the hard attribution $q_k^j = \delta_{k_j}^k$, where k_j is the class assigned to the jth observation.

There are a number of situations where a clustering assessment criterion is useful. The one that motivates us here is the problem of variable selection. When the number of observed variables greatly exceeds the number of independent observations, one may want to use only a subset of these variables. It is natural then to pick those that "best" cluster the data. In classification problems, for instance, one may choose the variables that optimize the clustering of the training data given by its actual class attribution.

A natural measure of how well the variables x_i support a clustering q_k^j is provided by the negative of the cross-entropy:

$$M = \sum_{k,j} q_k^j \log\left(p_k^j\right) \tag{24}$$

(the minus arising because we seek to discriminate among classes, i.e., order, not disorder). Here the probabilities p_k^j are the ones implied by the observed variables x_i under the given clustering; they follow from Bayes' formula (1), where the densities $\rho_k(x)$ maximize the log-likelihood functions

$$L_k = \sum_j q_k^j \log(\rho_k(x^j))\,. \tag{25}$$

The densities ρ_k can be computed by the same flow methodology described above, with the q_k^j's either provided—as when hard assignments into classes are known for the training population—or equated to the posterior assignments p_k^j. In the latter case, the procedure is identical to the one for clustering, except for a matter of emphasis: the quantity sought is not the set of assignments p_k^j or densities $\rho_k(x)$, though these are computed in the process, but the negative cross-entropy M.

Since the measure in (24) agrees, up to a sign and an additive constant, with the Kullback–Leibler (KL) divergence [10] of q and p,

$$D_{KL}(q,p) = \sum_{k,j} q_k^j \log\left(\frac{q_k^j}{p_k^j}\right), \tag{26}$$

the maximum possible value of M is achieved when $p_k^j = q_k^j$, i.e., when the variables x_i yield precisely the attribution provided. Then $D_{KL}(p,p) = 0$, and M becomes the negative entropy derived from the observations,

$$M = \sum_{k,j} p_k^j \log\left(p_k^j\right), \tag{27}$$

a meaningful measure of the effectiveness of the underlying variables for clustering.

To better understand the meaning of the attributions q_k^j, let us consider some typical applications to the selection of a subset $i \in I$ of variables from a larger set I_T.

- *Classification problem (first approach)*: In the classification problem, we typically have a training population for which the classes are known. We can then select those variables x_i that maximize M on the training population, where $q_k^j = \delta_{k_j}^k$. In this case, the measure M represents the log-likelihood of the posterior probabilities p_k^j under the observed attributions.

- *Clustering*: For clustering, one does not know the attributions q_k^j beforehand. Then we must adopt $q_k^j = p_k^j$, i.e., perform a clustering procedure from the variables x_i and assess its performance by its own implied negative entropy. Since $M \leq 0$, its maximum possible value is zero. This is achieved when the p_k^j are sharp ones and zeros, corresponding to clustering with complete certainty.
- *Classification problem (refined approach)*: When the training population is small and the testing one large, one might be tempted to use the clustering approach above rather than the one suggested for classification. In fact, one should always do this: using the training population alone for the assessment misses the information available in the observed values of x in the testing population. The best approach, then, uses the full combined population, with the known values of q_k^j for the training observations, and the posteriors $q_k^j = p_k^j$ for the testing ones.

Selecting a subset $I \subset I_T$ of the variables involves a combinatorial search, which can be prohibitively expensive when the cardinalities of I and I_T are large. Moreover, each density estimation can involve a significant amount of work. Simple practical strategies to reduce this work come in two flavors:

- Not to test all of I at once, but instead smaller subsets. In the simplest case, one computes the performance of each individual variable acting alone and selects those that rank at the top.
- Not to perform a full-blown density estimation for the ρ_k's, but a straightforward one, such as simple parametric estimation to an isotropic Gaussian.

Clearly, more sophisticated searches can be devised. The right balance depends on the significance of the variable reduction and the resources available. If the number of variables is being reduced just to make the problem more tractable, then the simplest strategy may be used. If the goal is to identify key variables, such as sets of genes related to a particular disease, a more thorough search is indicated. In the clinical examples below for tumor classification from microarray data, we have adopted the simplest approach of assessing each variable individually through a density estimation based on isotropic Gaussians, with very good results.

5. Clinical examples: Classification of tumors. In order to illustrate the use of the methodology proposed here, we applied it to two well-characterized data sets available in the literature, both concerned with the diagnosis of tumors from gene expression. The first data set [9] has the expression level of 2308 genes from tumor biopsies and cell lines of 83 patients with one of four types of the small, round blue cell tumors of childhood: neuroblastoma (NB), rhabdomyosarcoma (RMS), non-Hodgkin's lymphoma (NHL), and the Ewing family (EWS). The second set [7] has the expression level of 6817 genes from bone marrow and peripheral blood samples, of 72 patients with one of two classes of acute leukemia: acute lymphoblastic leukemia (ALL) and acute myeloid leukemia (AML). These two data sets are qualitatively different, as will become apparent in the plots below. We attribute this difference mainly to the fact that the 2308 genes in [9] are a subset from a total of 6567, filtered so that they all have at least a minimal level of expression. By contrast, the 6817 genes in [7] are unfiltered, which results in many genes having a uniform expression level in many of the samples. We shall see that the methodology proposed works well with both filtered and unfiltered data.

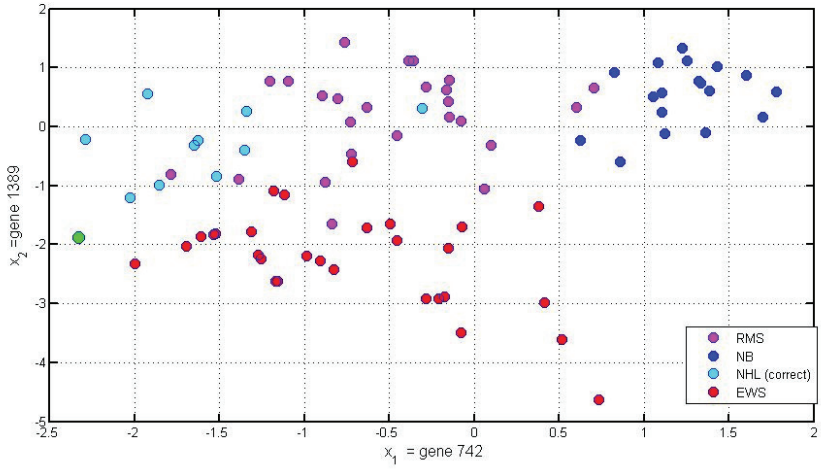

FIG. 1. *Raw data in the plane of two genes, selected because they ranked first in their ability to cluster the population. To make the selection procedure expedient, each gene was only considered individually, and the associated density estimation was reduced to its preconditioning step, which consists only of a rigid displacement and an isotropic linear rescaling. In the plot, the training population is colored according to class. The testing sample in green was chosen for this illustration because, lying on the outskirts of the clusters, it is among the most challenging to classify.*

5.1. Diagnosing one sample at a time. We first concerned ourselves with classification. In a first set of experiments, we picked each sample in turn as a testing case and used the remaining ones for training, with the goal of diagnosing the type of cancer of the test. This involves the following steps:

- Assigning a prior probability π_k^j that sample j belongs to the kth class. We used a uniform $\pi_k^j = \frac{1}{4}$ for the four childhood tumors and $\pi_k^j = \frac{1}{2}$ for the two lymphomas. Notice that these priors are assigned to the training population too, as they are needed for gene selection.
- Selecting a subset of genes. We rank the genes by the measure (27), where the P_k^j are the posteriors computed by the clustering algorithm using one gene at a time, the Q_k^j are ones or zeros for the training population, and $Q_k^j = P_k^j$ for the tests. Then we pick the n top-ranking genes.
- Classifying the testing samples (one in this case). Using the selected genes, we compute the posterior P_k^j for each test and assign it to the class with largest posterior.

When the number n of selected genes is large enough, just the preconditioning step of the algorithm scores a nearly perfect performance. Thus, with $n \geq 20$, all 83 cases with the round blue cell tumors are classified correctly, as are all but two of the 72 cases with acute leukemia with $n \geq 10$. The nonlinear steps allow us to further reduce the number of required genes for a correct diagnosis.

For illustration, we show the results of using just two genes to diagnose one particular patient with non-Hodgkin's lymphoma. Figure 1 shows the raw data in the "gene space" of the two genes selected by the algorithm based on the training population. The various types of tumors are shown in different colors, and the patient to be diagnosed is shown in green, since the code is not informed of the actual diagnosis. This patient tumor's type (NHL) is colored in cyan." Notice in this figure how well the two

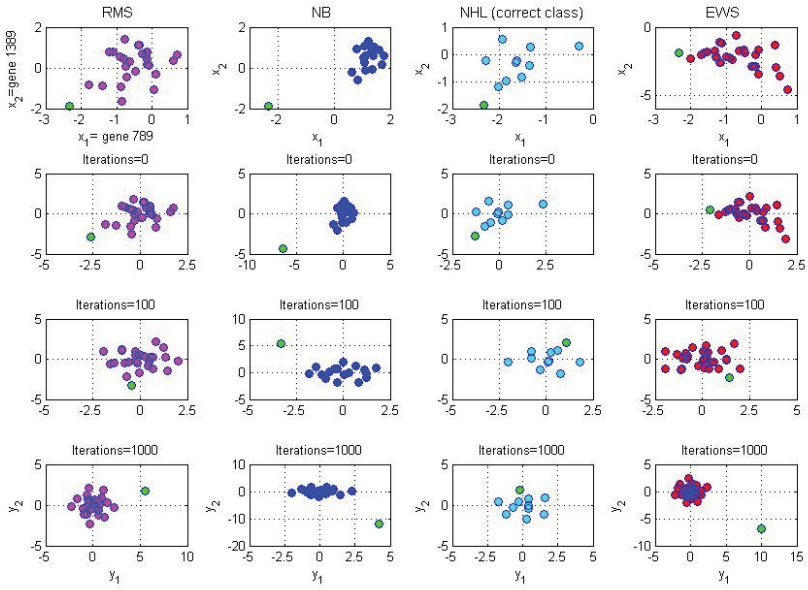

FIG. 2. *Datapoints transformed according to class, with the testing point in green softly assigned to the classes according to its posterior probability of belonging to each. The top panels display the raw data in the plane of the two selected genes, as in Figure* 1*, sorted by class. The second row is a rescaled and recentered version of the one above, corresponding to the preconditioning step. The third and fourth rows are snapshots of the nonlinear normalizing map that the algorithm performs. As each class approaches a Gaussian distribution, the testing point is either absorbed or rejected.*

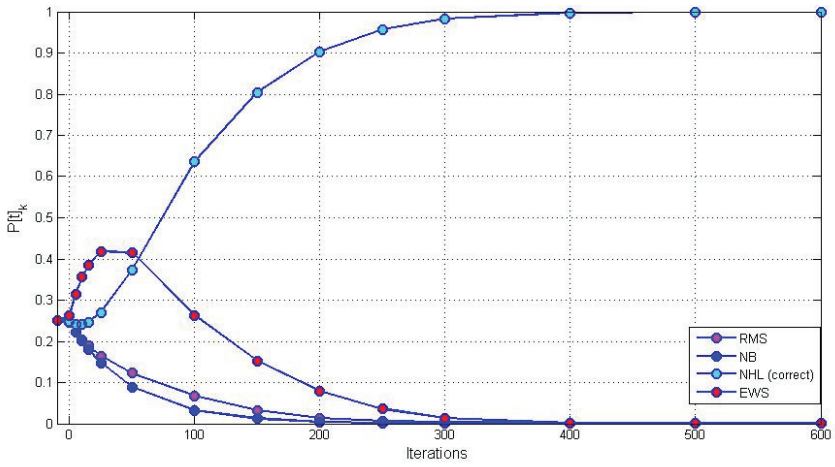

FIG. 3. *Evolving assignment* $P[t]_k$ *that the testing sample belongs in each class. Even though the initial trend assigns the sample incorrectly, as, after the linear rescaling, it is closest to the center of class D, this is corrected as the normalizing procedure unveils more detailed structure of the probability density of each class.*

genes chosen by the algorithm cluster the four types of tumors. Figures 2 and 3 tell graphically the story of how the diagnosis of this particular "green" patient evolves as the iterations progress. In Figure 2, we see the transformed variables $z = \phi_t(x)$

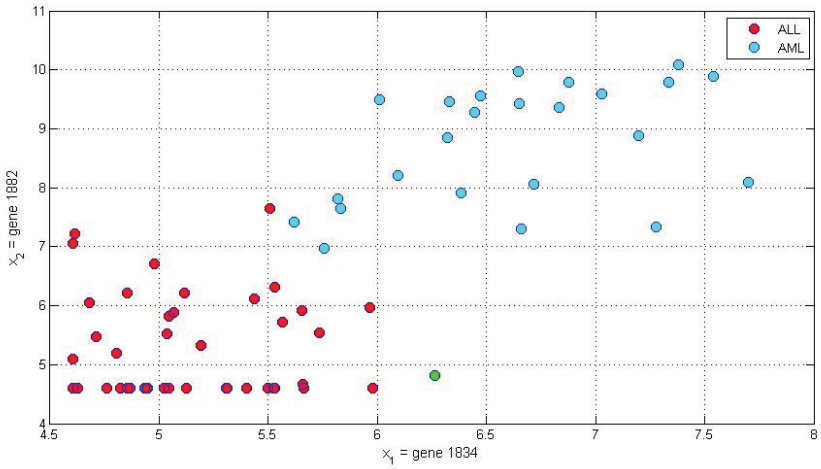

FIG. 4. *Same as Figure* 1*, but for the classification of lymphomas. Even though the samples are shown in the plane of the first two-genes selected by their clustering capability, the actual number of genes used is* 13*. Notice the presence, in this unfiltered dataset, of samples where the genes sit at their nominal minimal level of expression, requiring more than two genes for sensible clustering and classification. In particular, it would be impossible to classify correctly the test sample, plotted in green, from these two genes alone.*

for the four classes of tumors, including in each, through EM, the still undiagnosed patient. The top row of panels represents the same data as in Figure 1, separated by class. The second row shows the results of the preconditioning, which corresponds simply to a displaced and rescaled version of the top panel. At this level, with zero iterations, the code misassigns the patient to class EWS, since it is closer to its center than to that of NHL. However, neither of the corresponding clusters corresponds yet to an isotropic Gaussian. As the iterations progress in the next two rows of panels, we see the clusters evolving toward Gaussianity, with the green point clearly included in the cloud of NHL, and not in that of EWS. This evolution, at the level of the actual diagnosis, can be seen in Figure 3, which displays the evolution of the assignment $P[t]_k$, which relaxes to the posterior probability p_k^j that the tumor belongs to each of the four classes. Initially, all $P[t]_k$'s are equal to 0.25—the prior—and, though initially the probability of the wrong class (EWS) grows, it is eventually overcome overwhelmingly by the probability of the correct diagnosis, NHL. An example from the leukemia populations is depicted in Figures 4, 5, and 6. This particular diagnosis was produced with 13 genes, though the plots show only the plane of the first two genes at time $t = 0$, and of the first two components of $z = \phi_t(x)$ for later times. As Figure 6 shows, the patient is diagnosed correctly with AML from time zero, but with a probability only barely above 50%. As the iterations evolve, this probability reaches one. Figure 5 shows this evolution in the space of the first two variables where, as the clusters become more clearly Gaussian, the green point gets ejected from the wrong cloud, ALL, and absorbed into the one of its true class, AML.

5.2. Multisample diagnosis. Given the success of the classification procedure above, it is natural to ask whether we could have done as well with less information at our disposal. In particular, can we achieve similar results inverting the ratio of testing to training samples, i.e., using only a handful of training cases to diagnose most of the population?

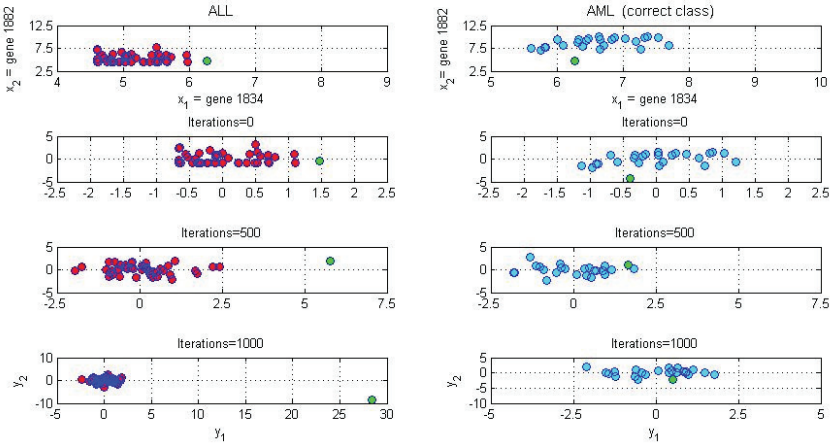

FIG. 5. *Same as Figure* 2*, for the two cases of acute leukemia. The evolution is displayed in the plane of the first two variables, which correspond to two genes at the initial time. Yet the algorithm works in a* 13*-dimensional space. It is out of the information in these* 13 *genes that the procedure manages to identify the correct class and, in the plots displayed, "eject" the sample from the Gaussian cluster of the incorrect class.*

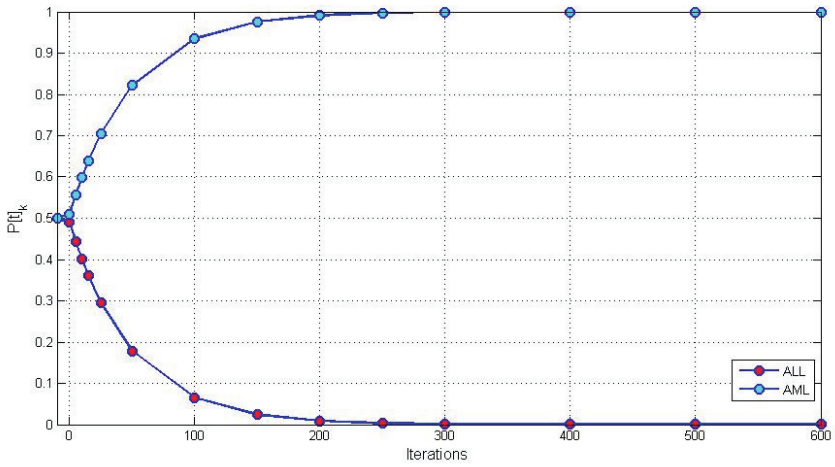

FIG. 6. *Same as Figure* 3*, but for the evolving assignment of the testing sample to one of the two classes of leukemia.*

To address this question, we reduced the training set for the classification of the four childhood tumors to only five samples per cancer type, and we used it to classify all the remaining samples. The results were invariably very good: using 60 genes, for instance, we classified correctly 95% of the samples. Similar results were obtained from the leukemia samples: from a training set of just two patients per class, 110 genes yielded 90% correct diagnoses. In all cases, the nonlinear component of the algorithm was fundamental to its success: the linear preconditioning step alone yielded a poorer outcome.

The weakest step of the procedure, when the training population is so small, is not the classification itself, but the selection of the genes to use: it is difficult to

assess which variables best cluster the various classes, when each class is severely unrepresented. This is compounded by the fact that the gene selection process itself is reduced to its bare essence: assessing one variable at a time, and just by linear means. When we select the best set of genes using a larger training population, we obtain a classification that is close to 100% correct, even when the training population for the classification itself is reduced to just two samples per class. This will be more thoroughly discussed in the subsection below in the context of clustering, where the situation is even more extreme, with an empty training population.

5.3. Clustering (class discovery). We can carry the multisample idea to the limit and get rid of the training population altogether. Here we cannot classify, since there is no longer a name attached to the various classes. Yet we can see if the clustering that the algorithm proposes agrees with the known classification by tumor type. In the language of [7], this is the process of "class discovery": patterns in the gene expression suggest the existence of various underlying tumor types.

The steps are the same as in the procedures above, except that we need to break the initial symmetry among classes. We do this by picking as initial soft assignments Q_k^j's not the priors π_k^j, but rather a small random perturbation,

$$Q_k^j = \pi_k^j + r_k^j\,, \tag{28}$$

where the r_k^j's are small random numbers adding to zero over k.

However, the problem of gene selection, already present when the training population was small, becomes acute when this vanishes completely. This is to be expected even from a purely philosophical perspective: we are asked to figure out, from a set of thousands of genes, which are the ones that best cluster a population of around a hundred patients. Yet without a training population that weights the cancer type in, the procedure may blindly cluster the datapoints from a different angle, be it by the patient's age, gender, ethnicity, blood type, or heart condition. Even random variations of the gene expression, not attributable to any specific biological cause, may give rise to robust clusters when the ratio of candidate variables to patients is so big.

Therefore, in our experiments, we selected which genes to utilize by making use of the diagnosis of all tumors, and only forgot these diagnoses when performing the clustering itself. The success rate is huge, reaching 100% once enough genes are used: for the childhood tumors, 70 genes are enough; for the two kinds of acute leukemia, just 10 do the job. We illustrate this in experiments with a smaller number of genes. Figure 7 shows the result of using 40 genes for clustering the childhood tumors: only one sample is placed in the wrong class. Figure 8 uses 6 genes for clustering the leukemia samples, discovering classes that agree with the type of leukemia in 93% of the samples.

By contrast, Figures 9 and 10 show the results of the same procedure when the gene-selection component is not informed of the actual class of any observation. Again 40 genes are used for the clustering of childhood tumors, and 6 genes for the clustering the leukemia samples. However, these genes are the ones that rank highest not in discriminating the tumor types, but in clustering the samples into four and two classes, respectively. The figures contrast the results with those obtained before, displaying them in the plane of the two top-ranking genes. The top panels, displaying the actual classes, show how unnatural the real classification looks in the space of the blindly selected genes, and, on the left, how natural it looks for the genes chosen in a more informed manner. The bottom panels show how, accordingly, the two different sets of

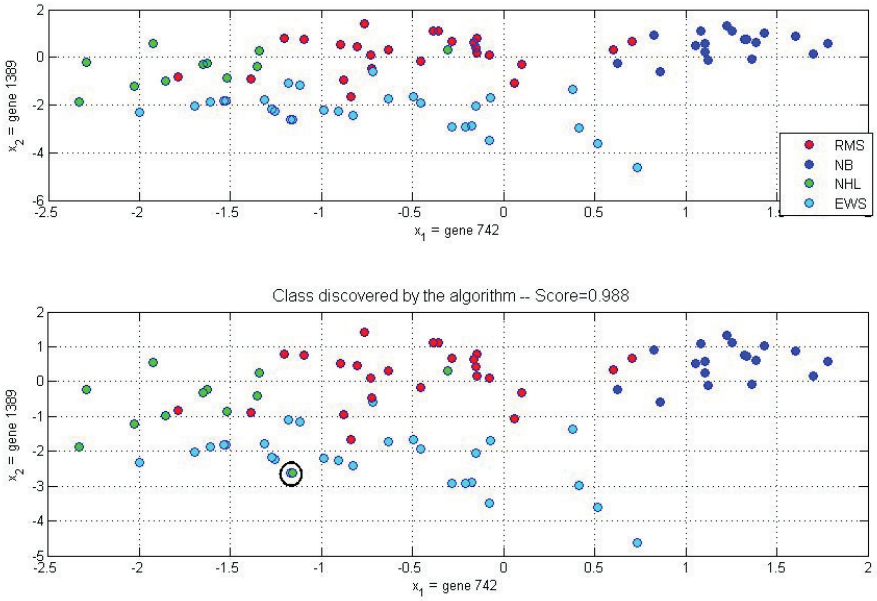

FIG. 7. *Clustering of the samples of childhood tumors, performed with* 40 *genes, but displayed in the plane of the first two. The top panel shows the actual tumor type, the bottom one the classes discovered by the algorithm. Only one observation, shown circled, was incorrectly assigned to a class.*

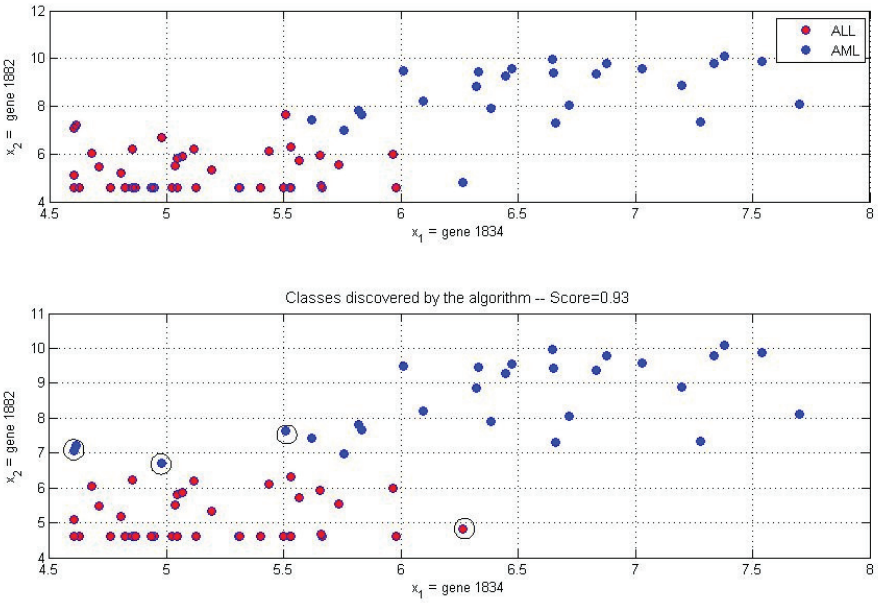

FIG. 8. *Same as Figure* 7*, but for the two types of acute leukemia, clustered using* 6 *genes. Shown circled are the observations assigned to classes that do not agree with the type of leukemia.*

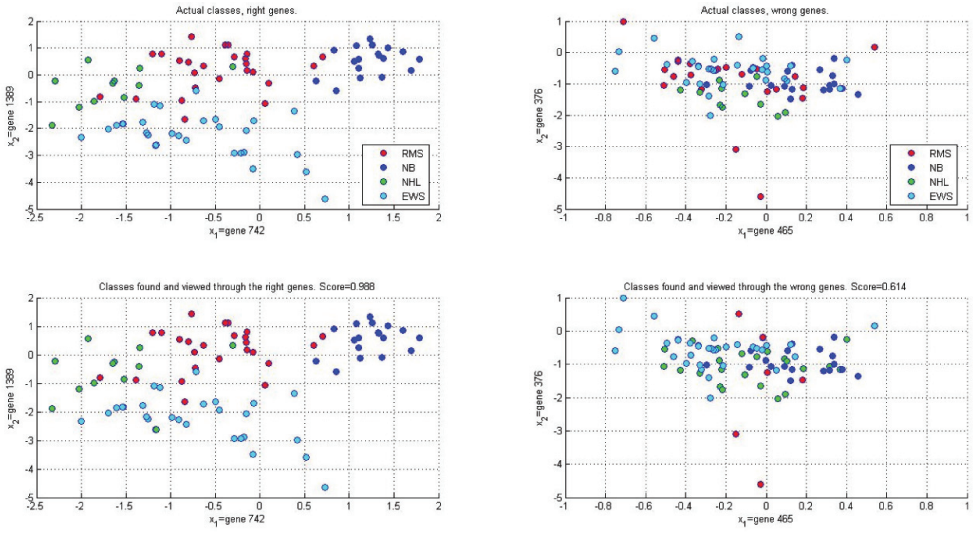

FIG. 9. *Clustering of the samples of the childhood tumors, performed with* 40 *genes, but displayed in the plane of the first two. The top panel shows the actual tumor type, the bottom one the classes discovered by the algorithm. The left panels display the results when the* 40 *genes were selected for their potential to distinguish the tumor classes; the results on the right panels, on the other hand, show genes selected only for their clustering potential using unlabeled observations.*

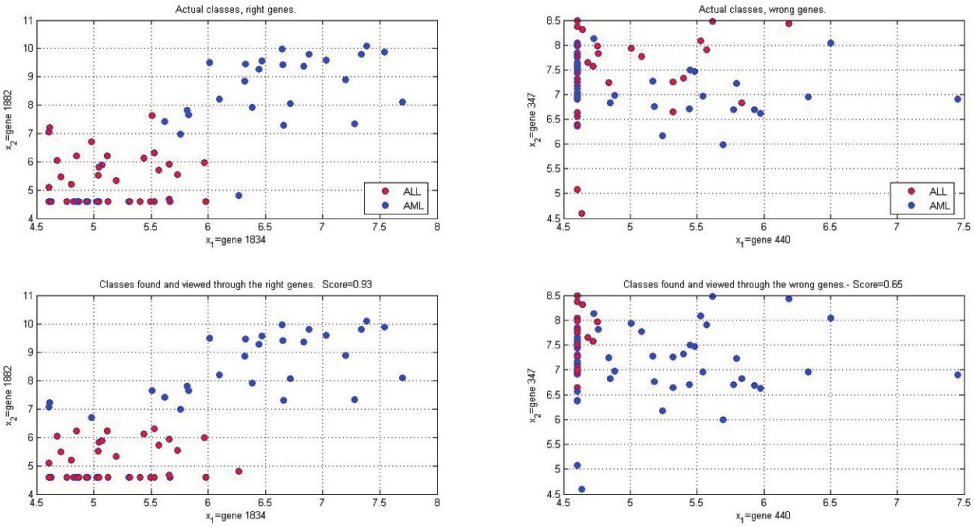

FIG. 10. *Same as Figure* 9*, but for the two types of acute leukemia, clustered using* 6 *genes.*

genes lead to different classifications, only one agreeing robustly with tumor type. It is not clear a priori that we should call the second classification "wrong" though: since the types were not provided, the scheme might be classifying the tumors not by type but, for instance, by state of development or any other unobserved characterization.

6. Conclusions. A general methodology was developed for classification and clustering and demonstrated on two well-characterized clinical examples involving the classification of tumors. The building block is a density-estimation procedure based on joint, multi-Gaussianization of the variables through fluid-like flows, where the observations play the role of active Lagrangian markers. As an observation becomes more clearly assigned to one class, it plays a more active role in the corresponding flow, while acting as a nearly passive marker for the others. The methodological framework involves the blurring of distinctions between training and testing populations. This serves the purpose not just of unifying the procedures for classification and clustering, but also of palliating the curse of dimensionality in classification problems with high-dimensional data through the use of unlabeled data.

Acknowledgment. The original motivation for this study arose from a problem posed by the cardiac transplant research group at Columbia University, directed by Dr. Mario Deng.

REFERENCES

[1] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, New York, 2006.
[2] O. Chapelle, B. Schölkopf, and A. Zien, *Semi-supervised Learning*, MIT Press, Cambridge, MA, 2006.
[3] S. S. Chen and R. A. Gopinath, *Gaussianization*, in Advances in Neural Information Processing Systems 13, T. K. Leen, T. G. Dietterich, and V. Tresp, eds., MIT Press, Cambridge, MA, 2001, pp. 423–429.
[4] A. Dempster, N. Laird, and D. Rubin, *Likelihood from incomplete data via the EM algorithm*, J. Roy. Statist. Soc. Ser. B, 39 (1977), pp. 1–38.
[5] I. S. Dhillon, S. Mallela, and R. Kumar, *A divisive information-theoretic feature clustering algorithm for text classification*, J. Mach. Learn. Res., 3 (2003), pp. 1265–1287.
[6] J. H. Friedman, W. Stuetzle, and A. Schroeder, *Projection pursuit density estimation*, J. Amer. Statist. Assoc., 79 (1984), pp. 599–608.
[7] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield, and E. S. Lander, *Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring*, Science, 286 (1999), pp. 531–537.
[8] I. Guyon and A. Elisseeff, *An introduction to variable and feature selection*, J. Mach. Learn. Res., 3 (2003), pp. 1157–1182.
[9] J. Khan, J. S. Wei, M. Ringnér, L. H. Saal, M. Ladanyi, F. Westermann, F. Berthold, M. Schwab, C. R. Antonescu, C. Peterson, and P. S. Meltzer, *Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks*, Nature Medicine, 7 (2001), pp. 673–679.
[10] S. Kullback and R. A. Leibler, *On information and sufficiency*, Ann. Math. Statistics, 22 (1951), pp. 79–86.
[11] E. Tabak and E. Vanden-Eijnden, *Density estimation by dual ascent of the log-likelihood*, Commun. Math. Sci., 8 (2010), pp. 217–233.
[12] V. N. Vapnik, *Statistical Learning Theory*, Wiley, New York, 1998.