

Adjusting the Outputs of a Classifier to New a Priori Probabilities: A Simple Procedure

Marco Saerens*, Patrice Latinne* & Christine Decaestecker[†]

Université Libre de Bruxelles

Brussels, Belgium

Abstract

It sometimes happens, for instance in case-control studies, that a classifier is trained on a data set which does not reflect the true a priori probabilities of the target classes on real-world data. This may

*Marco Saerens and Patrice Latinne are with IRIDIA Laboratory, cp. 194/6, Université Libre de Bruxelles, 50 av. Franklin Roosevelt, B-1050 Brussels, Belgium. Marco Saerens is also with the SmalS-MvM, Research Section, 102 rue du Prince Royal, B-1050 Brussels, Belgium. Email: saerens@ulb.ac.be, platinne@ulb.ac.be.

[†]Christine Decaestecker is a Research Associate with the Belgian Research Funds (F.N.R.S.) at the Laboratory of Histopathology, Université Libre de Bruxelles, 808 route de Lennik, B-1070 Brussels, Belgium. Email: cdecaes@ulb.ac.be.

have a negative effect on the classification accuracy obtained on the real-world data set, especially when the classifier's decisions are based on the a posteriori probabilities of class membership. Indeed, in this case, the trained classifier provides estimates of the a posteriori probabilities that are not valid for this real-world data set (they rely on the a priori probabilities of the training set). Applying the classifier as-is (without correcting its outputs with respect to these new conditions) on this new data set may thus be sub-optimal. In this note, we present a simple iterative procedure allowing to adjust the outputs of the trained classifier with respect to these new a priori probabilities without having to refit the model, even when these probabilities are not known in advance. As a by-product, estimates of the new a priori probabilities are also obtained. This iterative algorithm is a straightforward instance of the EM algorithm and is shown to maximize the likelihood of the new data. Thereafter, we discuss a statistical test that can be applied in order to decide if, indeed, the a priori class probabilities have changed from the training set to the real-word data. The procedure is illustrated on different classification problems involving a multi-layer neural network, and comparisons with a standard procedure for a priori probability estimation are provided. Our original method, based on the EM algorithm, is shown to be superior to the standard one for a priori probability estimation. Experimental results also indicate that the classifier with adjusted outputs always performs better than the original one in terms of classification accuracy, when the a priori probabilities conditions differ from the training set to the

real-world data. The gain in classification accuracy can be significant.

1 Introduction

In supervised classification tasks, it is sometimes the case that the a priori probabilities of the classes from a training set do not reflect the ‘true’ a priori probabilities of real-world data, on which the trained classifier has to be applied. For instance, this happens when the sample used for training is stratified by the value of the discrete response variable (i.e. the class membership). Consider for example an experimental setting – a case-control study – where we select 50% of individuals suffering from a disease (the cases) and 50% of individuals that do not suffer from this disease (the controls), and suppose that we make a set of measurements on these individuals. The resulting observations are used in order to train a model that classifies the data into the two target classes ‘disease’ and ‘no_disease’. In this case, the a priori probabilities of the two classes in the training set are 0.5 each. Once we apply the trained model in a real-world situation (new cases), we have no idea of the true a priori probability of disease (also labeled ‘disease prevalence’ in biostatistics). It has to be estimated from the new data. Moreover, the outputs of the model have to be adjusted accordingly. In other words, the classification model is trained on a data set with a priori probabilities that are different from the real-world conditions.

In this situation, knowledge of the ‘true’ a priori probabilities of the real-world data would be an asset for the following important reasons:

- Optimal Bayesian decision making is based on the *a posteriori probabilities* of the classes conditional on the observation (we have to select the class label that has maximum estimated a posteriori probability). Now, following Bayes' rule, these a posteriori probabilities depend in a non-linear way on the *a priori probabilities*. Therefore, a change of the a priori probabilities (as is the case for the real-world data versus the training set) may have an important impact on the a posteriori probabilities of membership, which themselves affect the classification rate. In other words, even if we use an optimal Bayesian model, if the a priori probabilities of the classes change, the model will not be optimal any more in these new conditions. But knowing the new a priori probabilities of the classes would allow us to correct (by Bayes' rule) the output of the model in order to recover the optimal decision.
- Many classification methods, including neural network classifiers, provide estimates of the a posteriori probabilities of the classes. From the previous point, this means that applying such a classifier as-is on new data having different a priori probabilities from the training set can result in a loss of classification accuracy, in comparison with an equivalent classifier that relies on the "true" a priori probabilities of the new data set.

This is the primary motivation of this paper: to introduce a procedure allowing to correct the estimated a posteriori probabilities, i.e. the classifier's outputs, in accordance with the new a priori probabilities of the real-world

data, in order to make more accurate predictions, even if these a priori probabilities of the new data set are not known in advance. As a by-product, estimates of the new a priori probabilities are also obtained. The experimental section (section 4) will indeed confirm that a significant increase in classification accuracy can be obtained when correcting the outputs of the classifier with respect to new a priori probabilities conditions.

For the sake of completeness, notice also that there exists another approach, the Min-Max criterion, that simply avoids the estimation of the a priori probabilities on the new data. Basically, the Min-Max criterion says that one should use the Bayes decision rule which corresponds to the least favorable a priori probability distribution (see, for instance, Melsa & Cohn, 1978 or Hand, 1981).

In brief, we present a simple iterative procedure that estimates the new a priori probabilities of a new data set and adjusts the outputs of the classifier – supposed to approximate the a posteriori probabilities – accordingly, without having to refit the model (section 2). This algorithm is a simple instance of the EM algorithm (Dempster et al., 1977; McLachlan & Krishnan, 1997) that aims to maximize the likelihood of the new observed data. We also discuss a simple statistical test (a likelihood ratio test) that can be applied in order to decide if the a priori probabilities have changed or not from the training set to the new data set (section 3). We illustrate the procedure on artificial and real classification tasks, and analyze its robustness with respect to imperfect estimation of the a posteriori probabilities provided by the classifier (section 4). Comparisons with a standard procedure used for a priori probabilities

estimation (also in section 4) and a discussion with respect to the related work (section 5) are also provided.

2 Correcting a posteriori probability estimates with respect to new a priori probabilities

2.1 Data classification

One of the most common use of data is classification. Suppose that we want to forecast the unknown discrete value of a dependent (or response) variable ω based on a measurement vector – or observation vector – \mathbf{x} . This discrete dependent variable takes its value in $\Omega = (\omega_1, \dots, \omega_n)$ – the n class labels.

A training example is therefore a realization of a random feature vector, \mathbf{x} , measured on an individual and allocated to one of the n classes $\in \Omega$. A *training set* is a collection of such training examples recorded for the purpose of model building (training), and forecasting based on that model.

The *a priori probability* of belonging to class ω_i in the *training set* will be denoted as $p_t(\omega_i)$ (in the sequel, subscript t will be used for estimates carried out on the basis of the *training set*). In the case-control example mentioned in the introduction, $p_t(\omega_1) = p_t(\text{disease}) = 0.5$, and $p_t(\omega_2) = p_t(\text{no_disease}) = 0.5$.

For the purpose of training, we suppose that, for each class ω_i , observations on N_t^i individuals belonging to the class (with $\sum_{i=1}^n N_t^i = N_t$, the total number of training examples) have been independently recorded according

to the within-class probability density, $p(\mathbf{x}|\omega_i)$. Indeed, case-control studies involve direct sampling from the within-class probability densities, $p(\mathbf{x}|\omega_i)$. In a case-control study with two classes (as reported in the introduction), this means that we made independent measurements on N_t^1 individuals that contracted the disease (the cases), according to $p(\mathbf{x}|\text{disease})$, and on N_t^2 individuals that did not (the controls), according to $p(\mathbf{x}|\text{no_disease})$. The a priori probabilities of the classes in the training set are therefore estimated by their frequencies $\hat{p}_t(\omega_i) = N_t^i/N_t$.

Let us suppose that we trained a classification model (the classifier), and denote by $\hat{p}_t(\omega_i|\mathbf{x})$ the estimated *a posteriori probability* of belonging to class ω_i provided by the classifier, given that the feature vector \mathbf{x} has been observed, in the conditions of the training set. The classification model (whose parameters are estimated on the basis of the training set as indicated by subscript t) could be an artificial neural network, a logistic regression, or any other model that provides, as output, estimates of the a posteriori probabilities of the classes given the observation. This is for instance the case if we use the least-squares error or the Kullback-Leibler divergence as a criterion for training, and if the minimum of the criterion is reached (see for instance Richard & Lippmann, 1991 or Saerens, 2000, for a recent discussion). We therefore suppose that the model has n outputs, $g_i(\mathbf{x})$ ($i = 1, \dots, n$), providing estimated posterior probabilities of membership $\hat{p}_t(\omega_i|\mathbf{x}) = g_i(\mathbf{x})$. In the experimental section (section 4), we will show that even imperfect approximations of these output probabilities allow reasonably good estimates.

Let us now suppose that the trained classification model has to be applied

on another data set (new cases, e.g. real-world data to be scored) for which the class frequencies, estimating the a priori probabilities $p(\omega_i)$ (no subscript t), are suspected to be different from $\hat{p}_t(\omega_i)$. The a posteriori probabilities provided by the model for these new cases will have to be corrected accordingly. As detailed in the two next sections, two cases must be considered according to the fact that estimates of the new a priori probabilities $\hat{p}(\omega_i)$ are, or are not, available for this new data set.

2.2 Adjusting the outputs to new a priori probabilities: New a priori probabilities known

In the sequel, we will assume that the generation of the observations within the classes, and thus the within-class densities, $p(\mathbf{x}|\omega_i)$, do not change from the training set to the new data set (only the relative proportion of measurements observed from each class has changed). This is a natural requirement – it supposes that we choose the training set examples only on the basis of the class labels ω_i , and not on the basis of \mathbf{x} . We also assume in this sub-section that we have an estimate of the new a priori probabilities, $\hat{p}(\omega_i)$.

Suppose now that we are working on a new data set to be scored. Bayes' theorem provides:

$$\hat{p}_t(\mathbf{x}|\omega_i) = \frac{\hat{p}_t(\omega_i|\mathbf{x})\hat{p}_t(\mathbf{x})}{\hat{p}_t(\omega_i)} \quad (1)$$

where the a posteriori probabilities $\hat{p}_t(\omega_i|\mathbf{x})$ are obtained by applying the *trained* model as-is (subscript t) on some observation \mathbf{x} of the new data set (i.e. by scoring the data). These are the estimated a posteriori probabilities

in the conditions of the training set (relying on the a priori probabilities of the training set).

The *corrected* a posteriori probabilities, $\widehat{p}(\omega_i|\mathbf{x})$ (relying this time on the estimated a priori probabilities of the new data set), obey the same equation, but with $\widehat{p}(\omega_i)$ as the new a priori probabilities and $\widehat{p}(\mathbf{x})$ as the new probability density function (no subscript t):

$$\widehat{p}(\mathbf{x}|\omega_i) = \frac{\widehat{p}(\omega_i|\mathbf{x})\widehat{p}(\mathbf{x})}{\widehat{p}(\omega_i)} \quad (2)$$

Since the within-class densities $\widehat{p}(\mathbf{x}|\omega_i)$ do not change from training to real-world data ($\widehat{p}_t(\mathbf{x}|\omega_i) = \widehat{p}(\mathbf{x}|\omega_i)$), by equating equation (1) to (2) and defining $f(\mathbf{x}) = \widehat{p}_t(\mathbf{x})/\widehat{p}(\mathbf{x})$, we find

$$\widehat{p}(\omega_i|\mathbf{x}) = f(\mathbf{x}) \frac{\widehat{p}(\omega_i)}{\widehat{p}_t(\omega_i)} \widehat{p}_t(\omega_i|\mathbf{x}) \quad (3)$$

Since $\sum_{i=1}^n \widehat{p}(\omega_i|\mathbf{x}) = 1$, we easily obtain $f(\mathbf{x}) = \left[\sum_{j=1}^n \frac{\widehat{p}(\omega_j)}{\widehat{p}_t(\omega_j)} \widehat{p}_t(\omega_j|\mathbf{x}) \right]^{-1}$, and consequently

$$\boxed{\widehat{p}(\omega_i|\mathbf{x}) = \frac{\frac{\widehat{p}(\omega_i)}{\widehat{p}_t(\omega_i)} \widehat{p}_t(\omega_i|\mathbf{x})}{\sum_{j=1}^n \frac{\widehat{p}(\omega_j)}{\widehat{p}_t(\omega_j)} \widehat{p}_t(\omega_j|\mathbf{x})}} \quad (4)$$

This well-known formula can be used in order to compute the corrected a posteriori probabilities, $\widehat{p}(\omega_i|\mathbf{x})$, in terms of the outputs provided by the trained model, $g_i(\mathbf{x}) = \widehat{p}_t(\omega_i|\mathbf{x})$, and the new priors $\widehat{p}(\omega_i)$. We observe that the new a posteriori probabilities $\widehat{p}(\omega_i|\mathbf{x})$ are simply the a posteriori probabilities in the conditions of the training set, $\widehat{p}_t(\omega_i|\mathbf{x})$, weighted by the ratio

of the new priors to the old priors, $\widehat{p}(\omega_i)/\widehat{p}_t(\omega_i)$. The denominator of (4) ensures that the corrected a posteriori probabilities sum to one.

However, in many real-world cases, we ignore what the real-world a priori probabilities $p(\omega_i)$ are since we do not know the class labels for these new data. This is the subject of the next section.

2.3 Adjusting the outputs to new a priori probabilities: New a priori probabilities unknown

When the new a priori probabilities are not known in advance, we cannot use the equation (4), and the $p(\omega_i)$ probabilities have to be estimated from the new data set. In this section, we first present an already known standard procedure used for new a priori probability estimation (the only one available in the literature to our knowledge); then we introduce our original method based on the EM algorithm.

2.3.1 Method 1: Confusion matrix

The standard procedure used for a priori probabilities estimation is based on the computation of the confusion matrix, $\widehat{p}(\delta_i|\omega_j)$, an estimation of the probability of taking the decision δ_i to classify an observation in class ω_i , while in fact it belongs to class ω_j (see for instance McLachlan, 1992 or McLachlan & Basford, 1988). In the sequel, this method will be referred to as the *confusion matrix* method. Here is its rationale. First, the confusion matrix $\widehat{p}_t(\delta_i|\omega_j)$ is estimated on the *training set* from cross-tabulated classification frequencies

provided by the classifier. Once this confusion matrix has been computed on the training set, it is used in order to infer the a priori probabilities on a new data set by solving the following system of n linear equations

$$\widehat{p}(\delta_i) = \sum_{j=1}^n \widehat{p}_t(\delta_i|\omega_j) \widehat{p}(\omega_j), \quad i = 1, \dots, n \quad (5)$$

with respect to the $\widehat{p}(\omega_j)$, where the $\widehat{p}(\delta_i)$ is simply the marginal of classifying an observation in class ω_i , estimated by the class label frequency after application of the classifier on the *new data set*. Once the $\widehat{p}(\omega_j)$ are computed from (5), we use equation (4) in order to infer the new a posteriori probabilities.

2.3.2 Method 2: EM algorithm

We now present a new procedure for a priori and a posteriori probabilities adjustment, based on the EM algorithm (Dempster et al., 1977; McLachlan and Krishnan, 1997). This iterative algorithm increases the likelihood of the new data at each iteration until a local maximum is reached.

Once again, let us suppose that we record a set of N new independent realizations of the stochastic variable \mathbf{x} , $\mathbf{X}_1^N = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N)$, sampled from $p(\mathbf{x})$, in a *new data set* to be scored by the model. The likelihood of these new observations is defined as:

$$\begin{aligned} L(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N) &= \prod_{k=1}^N p(\mathbf{x}_k) \\ &= \prod_{k=1}^N \left[\sum_{i=1}^n p(\mathbf{x}_k, \omega_i) \right] \end{aligned}$$

$$= \prod_{k=1}^N \left[\sum_{i=1}^n p(\mathbf{x}_k|\omega_i)p(\omega_i) \right] \quad (6)$$

where the within-class densities, i.e. the probabilities of observing \mathbf{x}_k given class ω_i , remain the same ($p(\mathbf{x}_k|\omega_i) = p_t(\mathbf{x}_k|\omega_i)$) since we assume that only the a priori probabilities change from the training set to the new data set. We have to determine the estimates $\hat{p}(\omega_i)$ that maximize the likelihood (6) with respect to $p(\omega_i)$. While a closed-form solution to this problem cannot be found, we can obtain an iterative procedure for estimating the new $p(\omega_i)$ by applying the EM algorithm.

As before, let us define $g_i(\mathbf{x}_k)$ as the model's output value corresponding to class ω_i for the observation \mathbf{x}_k of the new data set to be scored. The model outputs provide an approximation of the a posteriori probabilities of the classes given the observation in the conditions of the training set (subscript t), while the a priori probabilities of the training set are estimated by class frequencies:

$$\hat{p}_t(\omega_i|\mathbf{x}_k) = g_i(\mathbf{x}_k) \quad (7)$$

$$\hat{p}_t(\omega_i) = \frac{N_t^i}{N_t} \quad (8)$$

Let us define as $\hat{p}^{(s)}(\omega_i)$ and $\hat{p}^{(s)}(\omega_i|\mathbf{x}_k)$ the estimations of the new a priori and a posteriori probabilities at step s of the iterative procedure. If the $\hat{p}^{(s)}(\omega_i)$ are initialized by the frequencies of the classes in the training set (equation 8), the EM algorithm provides the following iterative steps (see

Appendix), for each new observation \mathbf{x}_k and each class ω_i :

$$\begin{aligned}
\hat{p}^{(0)}(\omega_i) &= \hat{p}_t(\omega_i) \\
\hat{p}^{(s)}(\omega_i|\mathbf{x}_k) &= \frac{\frac{\hat{p}^{(s)}(\omega_i)}{\hat{p}_t(\omega_i)} \hat{p}_t(\omega_i|\mathbf{x}_k)}{\sum_{j=1}^n \frac{\hat{p}^{(s)}(\omega_j)}{\hat{p}_t(\omega_j)} \hat{p}_t(\omega_j|\mathbf{x}_k)} \\
\hat{p}^{(s+1)}(\omega_i) &= \frac{1}{N} \sum_{k=1}^N \hat{p}^{(s)}(\omega_i|\mathbf{x}_k)
\end{aligned} \tag{9}$$

where $\hat{p}_t(\omega_i|\mathbf{x}_k)$ and $\hat{p}_t(\omega_i)$ are given by (7) and (8). The reader will notice the similarity between equations (4) and (9). At each iteration step s , both the a posteriori ($\hat{p}^{(s)}(\omega_i|\mathbf{x}_k)$) and the a priori probabilities ($\hat{p}^{(s)}(\omega_i)$) are re-estimated sequentially for each new observation \mathbf{x}_k and each class ω_i . The iterative procedure proceeds until the convergence of the estimated probabilities $\hat{p}^{(s)}(\omega_i)$.

Of course, if we have some a priori knowledge about the values of the prior probabilities, we can take these starting values for the initialization of the $\hat{p}^{(0)}(\omega_i)$. Notice also that, although we did not encounter this problem in our simulations, we must keep in mind that, potentially, local maxima problems may occur (the EM algorithm finds a *local* maximum of the likelihood function).

In order to obtain good a priori probability estimates, it is necessary that the a posteriori probabilities relative to the training set are reasonably well-approximated, i.e., sufficiently well estimated by the model. The robust-

ness of the EM procedure with respect to imperfect a posteriori probability estimates will be investigated in the experimental section (section 4).

3 Testing for different a priori probabilities

In this section, we show that the likelihood ratio test can be used in order to decide if the a priori probabilities have significantly changed from the training set to the new data set. Before adjusting the a priori probabilities (when the trained classification model is simply applied to the new data), the likelihood of the new observations is:

$$\begin{aligned} L_t(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N) &= \prod_{k=1}^N \hat{p}_t(\mathbf{x}_k) \\ &= \prod_{k=1}^N \left[\frac{\hat{p}(\mathbf{x}_k|\omega_i)\hat{p}_t(\omega_i)}{\hat{p}_t(\omega_i|\mathbf{x}_k)} \right] \end{aligned} \quad (10)$$

whatever the class label ω_i , and where we used the fact that $p_t(\mathbf{x}_k|\omega_i) = p(\mathbf{x}_k|\omega_i)$.

After the adjustment of the a priori and a posteriori probabilities, we compute the likelihood in the same way:

$$\begin{aligned} L(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N) &= \prod_{k=1}^N \hat{p}(\mathbf{x}_k) \\ &= \prod_{k=1}^N \left[\frac{\hat{p}(\mathbf{x}_k|\omega_i)\hat{p}(\omega_i)}{\hat{p}(\omega_i|\mathbf{x}_k)} \right] \end{aligned} \quad (11)$$

So that the likelihood ratio is:

$$\frac{L(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N)}{L_t(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N)} = \frac{\prod_{k=1}^N \left[\frac{\hat{p}(\mathbf{x}_k|\omega_i)\hat{p}(\omega_i)}{\hat{p}(\omega_i|\mathbf{x}_k)} \right]}{\prod_{k=1}^N \left[\frac{\hat{p}(\mathbf{x}_k|\omega_i)\hat{p}_t(\omega_i)}{\hat{p}_t(\omega_i|\mathbf{x}_k)} \right]}$$

$$= \frac{\prod_{k=1}^N \left[\frac{\widehat{p}(\omega_i)}{\widehat{p}(\omega_i|\mathbf{x}_k)} \right]}{\prod_{k=1}^N \left[\frac{\widehat{p}_t(\omega_i)}{\widehat{p}_t(\omega_i|\mathbf{x}_k)} \right]} \quad (12)$$

And the log-likelihood ratio is:

$$\log \left[\frac{L(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N)}{L_t(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N)} \right] = \sum_{k=1}^N \log [\widehat{p}_t(\omega_i|\mathbf{x}_k)] - \sum_{k=1}^N \log [\widehat{p}(\omega_i|\mathbf{x}_k)] \\ + N \log [\widehat{p}(\omega_i)] - N \log [\widehat{p}_t(\omega_i)] \quad (13)$$

From standard statistical inference (see for instance Mood et al., 1974; Papoulis, 1991), $2 \times \log [L(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N)/L_t(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N)]$ is asymptotically distributed as a chi-square with $(n-1)$ degrees of freedom ($\chi_{(n-1)}^2$, where n is the number of classes). Indeed, since $\sum_{i=1}^n \widehat{p}(\omega_i) = 1$, there are only $(n-1)$ degrees of freedom. This allows us to test if the new a priori probabilities differ significantly from the original ones and thus to decide if the a posteriori probabilities (i.e. the model outputs) need to be corrected.

4 Experimental evaluation

4.1 Simulations on artificial data

In this first subsection, we present a simple experiment that illustrates the iterative adjustment of the a priori and a posteriori probabilities. We chose a conventional multilayer perceptron (with one hidden layer, softmax output functions, trained with the Levenberg-Marquardt algorithm) as classification model, as well as a database labeled ‘Ringnorm’, introduced by Breiman

True priors	Estimated prior by using		Log-likelihood ratio test:
	EM	Confusion matrix	Number of times the test is significant
10%	14.7%	18.1%	10
20%	21.4%	24.2%	10
30%	33.0%	34.4%	10
40%	42.5%	42.7%	10
50%	49.2%	49.0%	0
60%	59.0%	57.1%	10
70%	66.8%	64.8%	10
80%	77.3%	73.9%	10
90%	85.6%	80.9%	10

Table 1: Results of the estimation of priors on the test sets, averaged on ten runs, for the Ringnorm artificial data set. The neural network has been trained on a learning set with a priori probabilities of (50%, 50%).

(1998)¹. This database is constituted of 7400 cases described by 20 numerical features, and divided into two equidistributed classes (each drawn from a multivariate normal distribution with a different variance-covariance matrix).

Ten replications of the following experimental design were applied. Firstly, a training set constituted of 500 cases of each class was extracted from the data ($p_t(\omega_1) = p_t(\omega_2) = 0.50$) and was used for training a neural network

¹Available at <http://www.cs.utoronto.ca/~delve/data/datasets.html>.

with 10 hidden units. For each training set, nine independent *test sets* of 1000 cases were selected according to the following a priori probability sequence: $p(\omega_1) = 0.10, 0.20, 0.30, 0.40, 0.50, 0.60, 0.70, 0.80, 0.90$ (with $p(\omega_2) = 1 - p(\omega_1)$). Then, for each test set, the EM procedure (9) as well as the confusion matrix procedure (5) were applied in order to estimate the new a priori probabilities and to adjust the a posteriori probabilities provided by the model ($\hat{p}_t(\omega_1|\mathbf{x}) = g(\mathbf{x})$). In each experiment, a maximum of 5 iteration steps of the EM algorithm was sufficient to ensure the convergence of the estimated probabilities.

Table 1 shows the estimated a priori probabilities for ω_1 . With respect to the EM algorithm, it also shows the number of times the likelihood ratio test was significant at $p < 0.01$ on these 10 replications. Table 2 presents the classification rates (computed on the test set) before and after the probability adjustments, as well as when the true priors of the test set ($p(\omega_i)$, which are unknown in a real-world situation) were used to adjust the classifier's outputs (using equation 4). This latter result can be consider as an optimal reference in the present experimental context.

The results reported in Table 1 show that the EM algorithm was clearly superior to the confusion matrix method for a priori probability estimation, and that the a priori probabilities are reasonably well estimated. Except in the cases where $p(\omega_i) = p_t(\omega_i) = 0.50$, the likelihood ratio test revealed in each replication a significant difference (at $p < 0.01$) between the training and the test set a priori probabilities ($\hat{p}_t(\omega_i) \neq \hat{p}(\omega_i)$). The a priori estimates appeared as slightly biased towards 50%; this appears as a bias affecting the

True priors	Percentage of correct classification			
	No adjustment	After adjustment by using		
		EM	Confusion matrix	True priors
10%	90.1%	93.6%	93.1%	94.0%
20%	90.3%	91.9%	91.7%	92.2%
30%	88.6%	89.9%	89.8%	90.0%
40%	90.4%	90.4%	90.4%	90.6%
50%	87.0%	86.9%	86.8%	87.0%
60%	90.0%	90.0%	90.0%	90.0%
70%	89.2%	89.8%	89.7%	90.2%
80%	89.5%	90.7%	90.7%	91.0%
90%	88.5%	91.6%	91.3%	92.0%

Table 2: Classification rates on the test sets, averaged on ten runs, for the Ringnorm artificial data set.

neural network classifier trained on an equidistributed training set.

By looking at Table 2 (classification results), we observe that the impact of the adjustment of the outputs on classification accuracy can be significant. The effect was most beneficial when the new a priori probabilities, $p(\omega_i)$, are far from the training set ones ($p_t(\omega_i) = 0.50$). Notice that, in each case, the classification rates obtained after adjustment were close to those obtained by using the true a priori probabilities of the test sets. While the EM algorithm provides better estimates of the a priori probabilities, we found no difference between the EM algorithm and the confusion matrix method in terms of clas-

sification accuracy. This could be due to the high recognition rates observed for this problem. Notice also that we observe a small degradation in classification accuracy if we adjust the a priori probabilities when not necessary ($p_t(\omega_i) = p(\omega_i) = 0.5$), as indicated by the likelihood ratio test.

4.2 Robustness evaluation on artificial data

This section aims to investigate the robustness of the EM-based procedure with respect to imperfect estimates of the a posteriori probability values provided by the classifier, as well as to the size of the training and the test set (the test set alone is used to estimate the new a priori probabilities). In order to degrade the classifier outputs, we gradually decreased the size of the training set in steps. Symmetrically, in order to reduce the amount of data available to the EM and the confusion matrix algorithms, we also gradually decreased the size of the test set. For each condition, we compared the classifier outputs with those obtained with a Bayesian classifier based on the true data distribution (which is known for an artificial data set such as ‘Ringnorm’). We were thus able to quantify the error level of the classifier with respect to the true a posteriori probabilities (how far is our neural network from the Bayesian classifier) and to evaluate the effects of a decrease in the training and/or test sizes on the a priori estimates provided by EM and the classification performances.

As for the experiment reported above, a multilayer perceptron was trained on the basis of an equidistributed training set ($p_t(\omega_1) = 0.5 = p_t(\omega_2)$). An

independent and unbalanced test set (with $p(\omega_1) = 0.20$ and $p(\omega_2) = 0.80$) was selected and scored by the neural network. The experiments (10 replications in each condition) were carried out on the basis of training and test sets with decreasing sizes (1000, 500, 200 and 100 cases) as detailed in Table 3.

We first compared the artificial neural network's output values ($g(\mathbf{x}) = \hat{p}_t(\omega_1|\mathbf{x})$, obtained by scoring the test sets with the trained neural network) with those provided by the Bayesian classifier ($b(\mathbf{x}) = p_t(\omega_1|\mathbf{x})$, obtained by scoring the test sets with the Bayesian classifier) on the test sets before outputs readjustment, i.e. we measured the discrepancy between the outputs of the neural and the Bayesian classifiers before outputs adjustment. For this purpose, we computed the averaged absolute deviation between the output value of the neural and the Bayesian classifiers (the average of $|b(\mathbf{x}) - g(\mathbf{x})|$) before outputs adjustment.

Then, for each test set, the EM and the confusion matrix procedures were applied to the outputs of the neural classifier in order to estimate the new a priori probabilities and the new a posteriori probabilities. The results for a priori probability estimation are detailed in Table 3.

By looking at the mean absolute deviation in Table 3, it can be seen that, as expected, decreasing the training set size results in a degradation in the estimation of the a posteriori probabilities (an increase of absolute deviation of about 0.10 between large, i.e. $N_t = 1000$, and small, i.e. $N_t = 100$, training set sizes). Of course, the prior estimates degraded accordingly, but only slightly. The EM algorithm appeared to be more robust than the confusion

Training set size ($\#\omega_1, \#\omega_2$)	Test set size ($\#\omega_1, \#\omega_2$)	Mean absolute deviation $\frac{1}{N} \sum_{k=1}^N b(\mathbf{x}_k) - g(\mathbf{x}_k) $	Estimated prior for ω_1 ($p(\omega_1) = 0.20$) by using	
			EM	Confusion matrix
(500, 500)	(200, 800)	0.107	22.0%	24.7%
	(100, 400)	0.110	21.6%	24.5%
	(40, 160)	0.104	20.4%	23.5%
	(20, 80)	0.122	22.7%	26.7%
(250, 250)	(200, 800)	0.139	22.1%	25.3%
	(100, 400)	0.140	22.6%	25.6%
	(40, 160)	0.134	23.1%	25.8%
	(20, 80)	0.167	22.7%	26.0%
(100, 100)	(200, 800)	0.183	24.1%	27.5%
	(100, 400)	0.185	24.4%	28.2%
	(40, 60)	0.181	23.5%	27.3%
	(20, 80)	0.180	26.6%	29.2%
(50, 50)	(200, 800)	0.202	24.9%	28.5%
	(100, 400)	0.199	25.3%	29.0%
	(40, 160)	0.203	24.3%	27.6%
	(20, 80)	0.189	22.3%	26.0%

Table 3: Averaged results for the estimation of the priors, in function of training and test sets of different sizes, for the Ringnorm data set. The results are averaged on 10 runs. Notice that the true priors of the test sets are (20%,80%).

matrix method. Indeed, in average (on all the experiments), the EM method overestimated the prior $p(\omega_1)$ by 3.3%, while the confusion matrix method overestimated by 6.6%. In contrast, decreasing the size of the test set seems to have very few effect on the results.

Figure 1 shows the classification rates (averaged on the 10 replications) of the neural network before and after the output adjustments made by the EM and the confusion matrix methods. It also illustrates the degradation in classifier performances due to the decrease in the size of the training sets (a loss of about 8% between large, i.e. $N_t = 1000$, and small, i.e. $N_t = 100$, training set sizes). The classification rates obtained after the adjustments made by the confusion matrix method are very close to those obtained with the EM method. In fact, the EM method almost always (15 times on the 16 conditions) provided better results, but the differences in accuracy between the two methods are very small (0.3% in average). As already observed in the first experiment reported above (see Table 2), the classification rates obtained after adjustment by the EM or the confusion matrix method are very close to those obtained by using the true a priori probabilities (a difference of 0.2% in average). Finally, we clearly observe (figure 1) that by adjusting the outputs of the classifier, we always increase significantly classification accuracy.

4.3 Tests on real data

We also tested the a priori estimation and outputs readjustment method on three real medical data sets from the UCI repository (Blake et al., 1998) in

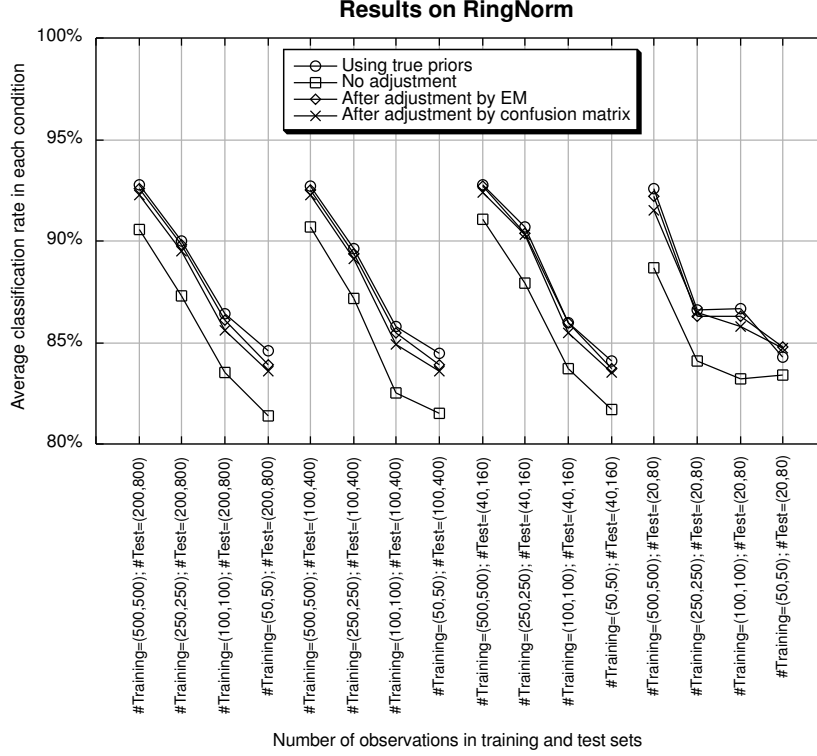


Figure 1: Classification rates obtained on the RingNorm data set. Results are reported for four different conditions: (1) Without adjusting the classifier output (i.e. No adjustment); (2) Adjusting the classifier output by using the confusion matrix method (i.e. After adjustment by confusion matrix); (3) Adjusting the classifier output by using the EM algorithm (i.e. After adjustment by EM); (4) Adjusting the classifier output by using the true a priori probability of the new data (i.e. Using true priors). The results are plotted in function of different sizes of both the training and the test sets.

order to confirm our results on more realistic data. These data are ‘Pima Indian Diabetes’ (2 classes of 268 and 500 cases, 8 features), ‘Breast Cancer Wisconsin’ (2 classes of 239 and 444 cases after omission of the 16 cases with missing values, 9 features) and ‘Bupa Liver Disorders’ (2 classes of 145 and 200 cases, 6 features). A training set constituted of 50 cases of each class was selected in each data set and used for training a multilayer neural network; the remaining cases were used for selecting an independent test set. In order to increase the difference between the class distributions in the training (0.50, 0.50) and the test sets, we omitted a number of cases from the smallest class in order to obtain a class distribution of $(p(\omega_1) = 0.20, p(\omega_2) = 0.80)$ for the test set. Ten different selections of training and test set were carried out, and for each of them the training phase was replicated 10 times, giving a total of 100 trained neural networks for each data set.

In average over the 100 experiments, Table 4 details the a priori probabilities estimated by means of the EM and the confusion matrix methods as well as the classification rates before and after the probability adjustments. These results show that the EM prior estimates were generally better than the confusion matrix ones (except for ‘Liver’). Moreover, adjusting the classifier outputs on the basis of the new a priori probabilities always increased classification rates, and provided accuracy levels not too far from those obtained by using the true priors for adjusting the outputs (given in the last column of Table 4). However, except for ‘Diabetes’ for which EM gave better results, the adjustments made on the basis of the EM and the confusion matrix methods seemed to have the same effect on the accuracy improvement.

Data set	True priors	Priors estimated by		Percentage of correct classification			
		EM	Confusion matrix	No adjustment	After adjustment by using		
					EM	Confusion matrix	True priors
Diabetes	20%	24.8%	31.3%	67.4%	76.3%	74.4%	78.3%
Breast	20%	18.0%	26.2%	91.3%	92.0%	92.1%	92.6%
Liver	20%	24.6%	21.5%	68.0%	75.7%	75.5%	79.1%

Table 4: Classification results on three real data sets. The neural network has been trained on a learning set with a priori probabilities of (50%, 50%).

5 Related work

The problem of estimating parameters of a model by including unlabeled data in addition to the labeled samples has been studied in both the machine learning and the artificial neural network communities. In this case, we speak about learning from partially labeled data (see for instance Shahshahani & Landgrebe, 1994; Ghahramani & Jordan, 1994; Castelli & Cover, 1995; Towell, 1996; Miller & Uyar, 1997; Nigam et al., 2000). The purpose is to use both labeled and unlabeled data for learning since unlabeled data are usually easy to collect, while labeled data are much more difficult to obtain. In this framework, the labeled part (the training set in our case) and the unlabeled part (the new data set in our case) are combined in one data set, and a partly-supervised EM algorithm is used in order to fit the model (a classifier) by maximizing the full likelihood of the complete set of data

(training set + new data set). For instance, Nigam et al. (2000) use the EM algorithm to learn classifiers that take advantage of both labeled and unlabeled data.

This procedure could easily be applied to our problem at hand, i.e. adjusting the a posteriori probabilities provided by a classifier to new a priori conditions. Moreover, it makes fully efficient use of the available data. However, at the down side, the model has to be completely refitted each time it is applied to a new data set. This is to be opposed to the approach discussed in this paper, where the model is fitted only on the training set. When applied to a new data set, the model is not modified: only its outputs are recomputed based on the new observations.

Related problems involving missing data have also been studied in applied statistics. Some good recent reference pointers are Scott & Wild (1997) and Lawless, Kalbfleisch & Wild (1999).

6 Conclusion

We presented a simple procedure allowing to adjust the outputs of a classifier to new a priori class probabilities. This procedure is a simple instance of the EM algorithm. When deriving this procedure, we relied on three fundamental assumptions:

1. The a posteriori probabilities provided by the model (our readjustment procedure can only be applied if the classifier provides as output an estimate of the a posteriori probabilities) are reasonably well-

approximated, which means that it provides predicted probabilities of belonging to the classes that are sufficiently close to the observed probabilities.

2. The training set selection (the sampling) has been performed on the basis of the discrete dependent variable (the classes), and not of the observed input variable \mathbf{x} (the explanatory variable), so that the within-class probability densities do not change.
3. The new data set to be scored is large enough in order to be able to estimate accurately the new a priori class probabilities.

If sampling also occurs on the basis of \mathbf{x} , the usual sample survey solution to this problem is to use weighted maximum likelihood estimators with weights inversely proportional to the selection probabilities, which are supposed to be known (see for instance Kish and Frankel, 1974).

Experimental results show that our new procedure based on EM performs better than the standard method (based on the confusion matrix) for new a priori probabilities estimation. The results also show that, even if the classifier's output provides imperfect a posteriori probability estimates

- The EM procedure is able to provide reasonably good estimates of the new a priori probabilities;
- The classifier with adjusted outputs always performs better than the original one if the a priori conditions differ from the training set to the real-world data. The gain of classification accuracy can be significant;

- The classification performances after adjustment by EM are relatively close to the results obtained by using the true priors (which are unknown in a real-world situation), even when the a posteriori probabilities are imperfectly estimated.

Additionally, the quality of the estimates does not appear to depend strongly on the size of the new data set. All these results enable us to relax to a certain extent the first and third assumptions here above.

We also observed that adjusting the outputs of the classifier when not needed (i.e. when the a priori probabilities of the training set and the real-world data do not differ) can result in a decrease in classification accuracy. We therefore showed that a likelihood ratio test can be used in order to decide if the a priori probabilities have significantly changed from the training set to the new data set. The readjustment procedure should be applied only when we find a significant change of a priori probabilities.

Notice that the EM-based adjustment procedure could be useful in the context of disease prevalence estimation. In this application, the primary objective is the estimation of the class proportions in an unlabeled data set (i.e. class a priori probabilities); classification accuracy is not important per se.

Another important problem, also encountered in medicine, concerns the automatic estimation of the proportions of different cell populations constituting, for example, a smear or a lesion (such as a tumor). Mixed tumors are composed of two or more cell populations with different lineages, as for

example in brain glial tumors (Decaestecker et al., 1997). In this case, a classifier is trained on a sample of images of ‘reference cells’ providing from tumors with a pure lineage (which did not present diagnostic difficulties) and labeled by experts. When a tumor is suspected to be mixed, the classifier is applied to a sample of cells from this tumor (a few hundred) in order to estimate the proportion of the different cell populations. The main motivation for the determination of the proportion of the different cell populations in these mixed tumors is that the different lineage components may significantly differ with respect to their susceptibility for aggressive progression and may thus influence patients’ prognoses. In this case, the primary goal is the determination of the proportion of cell populations, corresponding to the new a priori probabilities.

Another practical use of our readjustment procedure is the automatic labeling of geographical maps based on remote sensing information. Each portion of the map has to be labeled according to its nature (forest, agricultural zone, urban zone, etc). In this case, the a priori probabilities are unknown in advance and vary considerably from one image to another, since they directly depend on the geographical area that has been observed (urban area, country area, etc).

The authors are now actively working on these biomedical and geographical problems.

Acknowledgements

Part of this work was supported by the project RBC-BR 216/4041 from the ‘Région de Bruxelles-Capitale’, and funding from the SmalS-MvM. Patrice Latinne is supported by a grant under an ARC (Action de Recherche Concertée) program of the Communauté Française de Belgique. We also thank the two anonymous reviewers for their pertinent and constructive remarks.

References

- Blake, C., Keogh, E., & Merz, C. (1998). Uci repository of machine learning databases.
- Breiman, L. (1998). Arcing classifiers. *Annals of Statistics*, 26, 801–849.
- Castelli, V., & Cover, T. (1995). On the exponential value of labelled samples. *Pattern Recognition Letters*, 16, 105–111.
- Decaestecker, C., Lopes, M.-B., Gordower, L., Camby, I., Cras, P., Martin, J.-J., Kiss, R., VandenBerg, S., & Salmon, I. (1997). Quantitative chromatin pattern description in feulgen-stained nuclei as a diagnostic tool to characterise the oligodendroglial and astroglial components in mixed oligoastrocytomas. *Journal of Neuropathology and Experimental Neurology*, 56, 391–402.
- Dempster, A., Laird, N., & Rubin, D. (1977). Maximum likelihood from

- incomplete data via the em algorithm (with discussion). *Journal of the Royal Statistical Society B*, 39, 1–38.
- Ghahramani, Z., & Jordan, M. (1994). Supervised learning from incomplete data via an em algorithm. *Advances in Neural Information Processing Systems* (pp. 120–127). Morgan Kaufmann Publishers, Inc.
- Hand, D. (1981). *Discrimination and classification*. John Wiley & Sons.
- Kish, L., & Frankel, M. (1974). Inference from complex samples (with discussion). *Journal of the Royal Statistical Society B*, 61, 1–37.
- Lawless, J., Kalbfleisch, J., & Wild, C. (1999). Semiparametric methods for response-selective and missing data problems in regression. *Journal of the Royal Statistical Society B*, 61, 413–438.
- McLachlan, G. (1992). *Discriminant analysis and statistical pattern recognition*. John Wiley & Sons.
- McLachlan, G., & Basford, K. (1988). *Mixture models. inference and applications to clustering*. Marcel Dekker.
- McLachlan, G., & Krishnan, T. (1997). *The em algorithm and extensions*. Wiley.
- Melsa, J., & Cohn, D. (1978). *Decision and estimation theory*. McGraw-Hill.
- Miller, D., & Uyar, S. (1997). A mixture of experts classifier with learning based on both labeled and unlabeled data. *Advances in Neural Information Processing Systems*, 9, 571–578.

- Mood, A., Graybill, F., & Boes, D. (1974). *Introduction to the theory of statistics*. McGraw-Hill. 3rd edition.
- Nigam, K., McCallum, A., Thrun, S., & Mitchell, T. (2000). Text classification from labeled and unlabeled documents using em. *Machine Learning*, 39, 103–134.
- Papoulis, A. (1991). *Probability, random variables, and stochastic processes*. McGraw-Hill. 3rd edition.
- Richard, M., & Lippmann, R. (1991). Neural network classifiers estimate bayesian a posteriori probabilities. *Neural Computation*, 2, 461–483.
- Saerens, M. (2000). Building cost functions minimizing to some summary statistics. *IEEE Transactions on Neural Networks*, 11, 1263–1271.
- Scott, A., & Wild, C. (1997). Fitting regression models to case-control data by maximum likelihood. *Biometrika*, 84, 57–71.
- Shahshahani, B., & Landgrebe, D. (1994). The effect of unlabeled samples in reducing the small sample size problem and mitigating the hugues phenomenon. *IEEE Transactions on Geoscience and Remote Sensing*, 32, 1087–1095.
- Towell, G. (1996). Using unlabeled data for supervised learning. *Advances in Neural Information Processing Systems*, 8, 647–653.

A Appendix: Derivation of the EM algorithm

Our derivation of the iterative process (9) closely follows the estimation of mixing proportions of densities (see McLachlan and Krishnan, 1997). Indeed, $p(\mathbf{x}|\omega_i)$ can be viewed as a probability density defined by equation (1).

The EM algorithm supposes that there exists a set of unobserved data that, in the present case, are defined as the class labels of the observations of the new dataset. In order to pose the problem as an incomplete-data one, associated to the new observed data, $\mathbf{X}_1^N = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N)$, we introduce, as the unobservable data, $\mathbf{Z}_1^N = (\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_N)$, where each vector \mathbf{z}_k is associated to one of the n mutually exclusive classes: \mathbf{z}_k will represent the *class label* $\in (\omega_1, \dots, \omega_n)$ of the observation \mathbf{x}_k . More precisely, each \mathbf{z}_k will be defined as an *indicator vector*: if z_{ki} is the component i of vector \mathbf{z}_k , then $z_{ki} = 1$ and $z_{kj} = 0$ for each $j \neq i$ if and only if the class label associated to observation \mathbf{x}_k is ω_i . For instance if the observation \mathbf{x}_k is assigned to class label ω_i , then $\mathbf{z}_k = [0, \dots, 0, \underset{i-1}{1}, \underset{i}{1}, \underset{i+1}{0}, \dots, 0]^T$.

Let us denote by $\boldsymbol{\pi} = [p(\omega_1), p(\omega_2), \dots, p(\omega_n)]^T$ the vector of a priori probabilities (the parameters) to be estimated. The likelihood of the *complete data* (for the new data set) is

$$\begin{aligned} L(\mathbf{X}_1^N, \mathbf{Z}_1^N | \boldsymbol{\pi}) &= \prod_{k=1}^N \prod_{i=1}^n [p(\mathbf{x}_k, \omega_i)]^{z_{ki}} \\ &= \prod_{k=1}^N \prod_{i=1}^n [p(\mathbf{x}_k | \omega_i) p(\omega_i)]^{z_{ki}} \end{aligned} \quad (14)$$

where $p(\mathbf{x}_k | \omega_i)$ is constant (it does not depend on the parameter vector $\boldsymbol{\pi}$) and the $p(\omega_i)$ probabilities are the parameters to be estimated.

The log-likelihood is

$$\begin{aligned}
l(\mathbf{X}_1^N, \mathbf{Z}_1^N | \boldsymbol{\pi}) &= \log [L(\mathbf{X}_1^N, \mathbf{Z}_1^N | \boldsymbol{\pi})] \\
&= \sum_{k=1}^N \sum_{i=1}^n z_{ki} \log [p(\omega_i)] + \sum_{k=1}^N \sum_{i=1}^n z_{ki} \log [p(\mathbf{x}_k | \omega_i)] \quad (15)
\end{aligned}$$

Since the \mathbf{Z}_1^N data are unobservable, during the E-step, we replace the log-likelihood function by its conditional expectation over $p(\mathbf{Z}_1^N | \mathbf{X}_1^N, \boldsymbol{\pi})$: $E_{\mathbf{Z}_1^N} [l | \mathbf{X}_1^N, \boldsymbol{\pi}]$. Moreover, since we need to know the value of $\boldsymbol{\pi}$ in order to compute $E_{\mathbf{Z}_1^N} [l | \mathbf{X}_1^N, \boldsymbol{\pi}]$ (the expected log-likelihood), we use, as a current guess for $\boldsymbol{\pi}$, the current value (at iteration step s) of the parameter vector, $\hat{\boldsymbol{\pi}}^{(s)} = [\hat{p}^{(s)}(\omega_1), \hat{p}^{(s)}(\omega_2), \dots, \hat{p}^{(s)}(\omega_n)]^T$:

$$\begin{aligned}
Q(\boldsymbol{\pi}, \hat{\boldsymbol{\pi}}^{(s)}) &= E_{\mathbf{Z}_1^N} [l(\mathbf{X}_1^N, \mathbf{Z}_1^N | \boldsymbol{\pi}) | \mathbf{X}_1^N, \hat{\boldsymbol{\pi}}^{(s)}] \\
&= \sum_{k=1}^N \sum_{i=1}^n E_{\mathbf{Z}_1^N} [z_{ki} | \mathbf{x}_k, \hat{\boldsymbol{\pi}}^{(s)}] \log [p(\omega_i)] \\
&\quad + \sum_{k=1}^N \sum_{i=1}^n E_{\mathbf{Z}_1^N} [z_{ki} | \mathbf{x}_k, \hat{\boldsymbol{\pi}}^{(s)}] \log [p(\mathbf{x}_k | \omega_i)] \quad (16)
\end{aligned}$$

where we assumed that the complete-data observations $\{(\mathbf{x}_k, \mathbf{z}_k), k = 1, \dots, N\}$ are independent. We obtain for the expectation of the unobservable data

$$\begin{aligned}
E_{\mathbf{Z}_1^N} [z_{ki} | \mathbf{x}_k, \hat{\boldsymbol{\pi}}^{(s)}] &= 0 \cdot p(z_{ki} = 0 | \mathbf{x}_k, \hat{\boldsymbol{\pi}}^{(s)}) + 1 \cdot p(z_{ki} = 1 | \mathbf{x}_k, \hat{\boldsymbol{\pi}}^{(s)}) \\
&= p(z_{ki} = 1 | \mathbf{x}_k, \hat{\boldsymbol{\pi}}^{(s)}) \\
&= \hat{p}^{(s)}(\omega_i | \mathbf{x}_k) \\
&= \frac{\frac{\hat{p}^{(s)}(\omega_i)}{\hat{p}_t(\omega_i)} \hat{p}_t(\omega_i | \mathbf{x}_k)}{\sum_{j=1}^n \frac{\hat{p}^{(s)}(\omega_j)}{\hat{p}_t(\omega_j)} \hat{p}_t(\omega_j | \mathbf{x}_k)} \quad (17)
\end{aligned}$$

where we used the equation (4) at the last step. The expected likelihood is therefore

$$\begin{aligned} Q(\boldsymbol{\pi}, \hat{\boldsymbol{\pi}}^{(s)}) &= \sum_{k=1}^N \sum_{i=1}^n \hat{p}^{(s)}(\omega_i | \mathbf{x}_k) \log [p(\omega_i)] \\ &\quad + \sum_{k=1}^N \sum_{i=1}^n \hat{p}^{(s)}(\omega_i | \mathbf{x}_k) \log [p(\mathbf{x}_k | \omega_i)] \end{aligned} \quad (18)$$

where $\hat{p}^{(s)}(\omega_i | \mathbf{x}_k)$ is given by (17).

For the M-step, we compute the maximum of $Q(\boldsymbol{\pi}, \hat{\boldsymbol{\pi}}^{(s)})$ (equation 18) with respect to the parameter vector $\boldsymbol{\pi} = [p(\omega_1), p(\omega_2), \dots, p(\omega_n)]^T$. The new estimate at time step $(s+1)$ will therefore be the value of the parameter vector $\boldsymbol{\pi}$ that maximizes $Q(\boldsymbol{\pi}, \hat{\boldsymbol{\pi}}^{(s)})$. Since we have the constraint, $\sum_{i=1}^n p(\omega_i) = 1$, we define the Lagrange function as

$$\begin{aligned} \ell(\boldsymbol{\pi}) &= Q(\boldsymbol{\pi}, \hat{\boldsymbol{\pi}}^{(s)}) + \lambda \left[1 - \sum_{i=1}^n p(\omega_i) \right] \\ &= \sum_{k=1}^N \sum_{i=1}^n \hat{p}^{(s)}(\omega_i | \mathbf{x}_k) \log [p(\omega_i)] + \sum_{k=1}^N \sum_{i=1}^n \hat{p}^{(s)}(\omega_i | \mathbf{x}_k) \log [p(\mathbf{x}_k | \omega_i)] \\ &\quad + \lambda \left[1 - \sum_{i=1}^n p(\omega_i) \right] \end{aligned} \quad (19)$$

By computing $\frac{\partial \ell(\boldsymbol{\pi})}{\partial p(\omega_j)} = 0$, we obtain

$$\sum_{k=1}^N \hat{p}^{(s)}(\omega_j | \mathbf{x}_k) = \lambda p(\omega_j) \quad (20)$$

for $j = 1, \dots, n$. If we sum this equation over j , we obtain the value of the Lagrange parameter, $\lambda = N$, so that

$$p(\omega_j) = \frac{1}{N} \sum_{k=1}^N \hat{p}^{(s)}(\omega_j | \mathbf{x}_k) \quad (21)$$

and the next estimate of $p(\omega_i)$ is therefore

$$\hat{p}^{(s+1)}(\omega_i) = \frac{1}{N} \sum_{k=1}^N \hat{p}^{(s)}(\omega_i | \mathbf{x}_k) \quad (22)$$

So that the equations (17) (E-step) and (22) (M-step) are repeated until the convergence of the parameter vector $\boldsymbol{\pi}$. The overall procedure is summarized in (9). It can be shown that this iterative process increases the likelihood (6) at each step (see for instance Dempster et al., 1977; McLachlan and Krishnan, 1997). ■
