Optimal projection to estimate the proportions of the different subsamples in a given mixture sample

Lluís Garrido[1,3], Sergio Gómez[2],
Aurelio Juste[3], Vicens Gaitan[3]

1) Dept. d'Estructura i Constituents de la Matèria,
Facultat de Física, Universitat de Barcelona,
Diagonal 647, E-08028 Barcelona, Spain.
Phone: +34 3 402 11 91 Fax: +34 3 402 11 98
e-mail: garrido@ecm.ub.es

2) Dept. d'Enginyeria Informàtica,
E.T.S.E., Universitat Rovira i Virgili,
Crt. de Salou s/n, E-43006 Tarragona, Spain.
Phone: +34 77 55 96 81 Fax: +34 77 55 97 10
e-mail: sgomez@etse.urv.es

3) Institut de Física d'Altes Energies,
Universitat Autònoma de Barcelona,
E-08193 Bellaterra (Barcelona), Spain.
Phone: +34 3 691 30 12 Fax: +34 3 581 19 38
e-mail: juste@ifae.es gaitan@ifae.es

Abstract

Given a n-dimensional sample composed of a mixture of m subsamples with different probability density functions (p.d.f.), it is possible to build a $(m-1)$-dimensional distribution that carries all the information about the subsample proportions in the mixture sample. This projection can be estimated without an analytical knowlegde of the p.d.f.'s of the different subsamples with the aid, for instance, of neural networks. This way, if $m-1 < n$ it is possible to estimate the proportions of the mixture sample in a lower $(m-1)$-dimensional space without losing sensitivity.

PACS'96: 02.50.Ph, 07.05.Kf, 07.05.Mh

Published in *Computer Physics Communications* **104** (1997) 37–45.

1 Introduction

Very often in Physics and other fields we have a sample of n-dimensional data composed of a mixture of m subsamples, whose proportions have to be estimated without analytical expressions for their corresponding probability density functions (p.d.f.), but being able to generate Monte Carlo events according to them. However, the estimation of these proportions is not simple and very often is done by binning the original n-dimensional space and performing a χ^2 adjustment. This means that, even for a low number of bins per dimension, large amounts of data are necessary since the number of data points needed to fill the bins with enough statistical significance grows exponentially with the number of variables. Thus, it would be of great interest to have a projection that could reduce the dimensionality of the original n-dimensional space without sensitivity loss, since this would mean that the fit in this projected space could be done with a smaller data set.

In Sect. 2 we characterize the optimal projection space which can be found imposing that no information is lost, and in Sect. 3 it is shown how neural networks may be used to implement this projection. Finally, an example is explained in Sect. 4.

2 Optimal choice of the fitting space

Let us suppose that we have a mixture sample $\{\boldsymbol{x}_e,\, e = 1, \ldots, N\}$ where each n-dimensional $\boldsymbol{x}_e$ belongs to one of m different classes (or subsamples) $\{\mathcal{C}_i,\, i = 1, \ldots, m\}$, but we do not know to which one. The p.d.f. of the i-th subsample is $p_i(\boldsymbol{x})$, and its proportion in the mixture sample is α_i (in probability terms, $p_i(\boldsymbol{x}) = p(\boldsymbol{x}|\mathcal{C}_i)$ and $\alpha_i = P(\mathcal{C}_i)$). The maximum likelihood estimator of $\boldsymbol{\alpha}$ is obtained by maximization of the log-likelihood function

$$l_x(\boldsymbol{\alpha}) = \sum_{e=1}^{N} \ln \left(\sum_{i=1}^{m} \alpha_i p_i(\boldsymbol{x}_e) \right) + \lambda \left(\sum_{i=1}^{m} \alpha_i - 1 \right) . \tag{2.1}$$

with respect to $\boldsymbol{\alpha}$. The last term is a lagrangian multiplier term to ensure that

$$\sum_{i=1}^{m} \alpha_i = 1 \, . \tag{2.2}$$

However, eq. (2.1) cannot be used if we do not have analytical expressions for the $p_i(\boldsymbol{x})$, or we do not know how to calculate them for the whole sample. One possibility is to bin the $\boldsymbol{x}$ space and approximate the probability of each bin by using Monte Carlo data, but this is not practical unless the dimensionality of the original space is low, since the number of Monte Carlo data needed grows exponentially with that dimension.

Let $(\boldsymbol{y}, \boldsymbol{r}) = F(\boldsymbol{x})$ be an invertible transformation with $\dim(\boldsymbol{y}) + \dim(\boldsymbol{r}) = n$, and let us call f the first components of F such that $\boldsymbol{y} = f(\boldsymbol{x})$. The p.d.f. of the i-th class as a function of the $\boldsymbol{y}$ variables can then be written as

$$q_i(\boldsymbol{y}) = \int d\boldsymbol{r}\, p_i(F^{-1}(\boldsymbol{y}, \boldsymbol{r})) J_F(\boldsymbol{y}, \boldsymbol{r}) \, , \tag{2.3}$$

where $J_F(\boldsymbol{y}, \boldsymbol{r})$ is the jacobian of the F transformation. The goal is to find a set of variables $\boldsymbol{y}$ with $\dim(\boldsymbol{y}) < \dim(\boldsymbol{x})$, not depending on $\boldsymbol{\alpha}$, for which the log-likelihood function on the transformed sample $\{\boldsymbol{y}_e = f(\boldsymbol{x}_e), e = 1, \ldots, N\}$,

$$l_y(\boldsymbol{\alpha}) = \sum_{e=1}^{N} \ln \left(\sum_{i=1}^{m} \alpha_i q_i(\boldsymbol{y}_e) \right) + \lambda \left(\sum_{i=1}^{m} \alpha_i - 1 \right) , \tag{2.4}$$

is the same as $l_x(\boldsymbol{\alpha})$, up to a constant not depending on $\boldsymbol{\alpha}$. This will assure not only that the expected value of the $\boldsymbol{\alpha}$ estimator will be the same, but also its variance will be as well; in other words, the fit sensitivity will not be reduced.

Deriving l_x and l_y with respect to α_k and equalling the two expressions yields

$$\sum_{e=1}^{N} \left(\frac{1}{\sum_{i=1}^{m} \alpha_i \frac{p_i(\boldsymbol{x}_e)}{p_k(\boldsymbol{x}_e)}} - \frac{1}{\sum_{i=1}^{m} \alpha_i \frac{q_i(\boldsymbol{y}_e)}{q_k(\boldsymbol{y}_e)}} \right) = 0 . \tag{2.5}$$

The only way to assure the above result (independently of the data sample under consideration) for the whole $\boldsymbol{\alpha}$ range is to impose that

$$\frac{p_i(\boldsymbol{x})}{p_k(\boldsymbol{x})} = \frac{q_i(\boldsymbol{y})}{q_k(\boldsymbol{y})} , \ \forall \boldsymbol{y} = f(\boldsymbol{x}) , \ \forall i, k = 1, \ldots, m . \tag{2.6}$$

Not all the above equalities are independent. Fixing k to an arbitrary class (say $k = m$) results in the set of $m - 1$ equations

$$p_i^*(\boldsymbol{x}) \equiv \frac{p_i(\boldsymbol{x})}{p_m(\boldsymbol{x})} = \frac{q_i(\boldsymbol{y})}{q_m(\boldsymbol{y})} \equiv q_i^*(\boldsymbol{y}) , \ \forall \boldsymbol{y} = f(\boldsymbol{x}) , \ i = 1, \ldots, m-1 . \tag{2.7}$$

Equations $z_i = p_i^*(\boldsymbol{x})$, $i = 1, \ldots, m-1$, map the n-dimensional vector $\boldsymbol{x}$ to the $(m-1)$-dimensional space of $\boldsymbol{z}$. If $n < m-1$, the $\boldsymbol{z}$ vector spans a n-dimensional manifold embedded in the $(m-1)$-dimensional space, but this case is of no interest. Let us then assume that $n \geq m-1$. Due to eqs. (2.7), it must be possible to perform this transformation in two steps: first, mapping the $\boldsymbol{x}$ vector to the $\boldsymbol{y}$ space (still with unknown dimension) using that $\boldsymbol{y} = f(\boldsymbol{x})$, and then to the $\boldsymbol{z}$ space through $z_i = q_i^*(\boldsymbol{y})$. The dimension of $\boldsymbol{y}$ cannot be lower than the dimension of $\boldsymbol{z}$. Thus, the most favourable case (such that minimizes the dimension of the projection space) is when the $\boldsymbol{y}$ vector is also $(m-1)$-dimensional.

A useful solution to eqs. (2.7), which generalizes that of [4], is

$$y_i(\boldsymbol{x}) \equiv \frac{\alpha_i^0 p_i(\boldsymbol{x})}{\sum_{j=1}^{m} \alpha_j^0 p_j(\boldsymbol{x})} , \ i = 1, \ldots, m-1 , \tag{2.8}$$

where $\{\alpha_i^0\}$ are arbitrary positive numbers fulfiling

$$\sum_{j=1}^{m} \alpha_j^0 = 1 . \tag{2.9}$$

More precisely, taking into account projection (2.8) and that $\boldsymbol{x} = F^{-1}(\boldsymbol{y}, \boldsymbol{r})$,

$$\frac{\alpha_i^0 q_i(\boldsymbol{y})}{\sum_{j=1}^{m} \alpha_j^0 q_j(\boldsymbol{y})} = \frac{\alpha_i^0 \int d\boldsymbol{r}\, p_i(\boldsymbol{x}) J_F(\boldsymbol{y}, \boldsymbol{r})}{\int d\boldsymbol{r} \left(\sum_{j=1}^{m} \alpha_j^0 p_j(\boldsymbol{x}) \right) J_F(\boldsymbol{y}, \boldsymbol{r})} = y_i(\boldsymbol{x})\,, \tag{2.10}$$

it is clear that eqs. (2.7) are satisfied, since

$$\frac{y_i/\alpha_i^0}{y_m/\alpha_m^0} = \frac{p_i(\boldsymbol{x})}{p_m(\boldsymbol{x})} = \frac{q_i(\boldsymbol{y})}{q_m(\boldsymbol{y})}\,,\ \forall \boldsymbol{y} = f(\boldsymbol{x})\,,\ i = 1, \ldots, m-1\,, \tag{2.11}$$

where we have introduced for convenience a new dependent variable

$$y_m \equiv 1 - \sum_{i=1}^{m-1} y_i = \frac{\alpha_m^0 p_m(\boldsymbol{x})}{\sum_{j=1}^{m} \alpha_j^0 p_j(\boldsymbol{x})}\,. \tag{2.12}$$

In addition, if we transform our variables $\boldsymbol{y}$ to other new variables $\boldsymbol{y'} = g(\boldsymbol{y})$ through any arbitrary invertible transformation g, the new variables will also be solutions to eqs. (2.7), since

$$\frac{q_i'(\boldsymbol{y'})}{q_m'(\boldsymbol{y'})} = \frac{q_i'(\boldsymbol{y'}) J_g(\boldsymbol{y'})}{q_m'(\boldsymbol{y'}) J_g(\boldsymbol{y'})} = \frac{q_i(\boldsymbol{y})}{q_m(\boldsymbol{y})}\,,\ \forall \boldsymbol{y'} = g(\boldsymbol{y})\,,\ i = 1, \ldots, m-1\,. \tag{2.13}$$

For instance, the variables $\boldsymbol{y}$ and $\boldsymbol{z}$ defined above are related by an invertible transformation provided that the proportions and p.d.f. at any point $\boldsymbol{x}$ of all the m classes are not zero. Thus, it can be said that the most general solution to eqs. (2.7) is any invertible transformation of variables $z_i = p_i^*(\boldsymbol{x})$, as one would naively expect from that equation.

The advantage of projection (2.8) is that the new variables $\boldsymbol{y}$ directly represent the Bayesian a-posteriori probability of the different classes for some a-priori probabilities (subsamples proportions) $\boldsymbol{\alpha}^0$, which need not be the same as the unknown $\boldsymbol{\alpha}$ of the mixture sample that we are trying to estimate. That is,

$$y_i(\boldsymbol{x}) = P_0(\mathcal{C}_i|\boldsymbol{x})\,,\ i = 1, \ldots, m-1\,. \tag{2.14}$$

Therefore, if we know how to generate Monte Carlo according to $p_i(\boldsymbol{x})$ for some proportions $\boldsymbol{\alpha}^0 = \boldsymbol{\alpha}^{\mathrm{MC}}$, and also how to estimate the corresponding a-posteriori probabilities $P_0(\mathcal{C}_i|\boldsymbol{x}) = P_{\mathrm{MC}}(\mathcal{C}_i|\boldsymbol{x})$, then we will be able to make the projection (2.14) and fit the $\boldsymbol{\alpha}$ on this lower dimensional space.

Finally, the relationship between the log-likelihood expressions in terms of variables $\boldsymbol{x}$ and $\boldsymbol{y}$ is

$$l_x(\boldsymbol{\alpha}) = l_y(\boldsymbol{\alpha}) + \sum_{e=1}^{N} \frac{p_m(\boldsymbol{x}_e)}{q_m(\boldsymbol{y}_e)}\,, \tag{2.15}$$

which guarantees that the maximum and the fit sensitivity is the same in both original and projected spaces.

3 Implementation with Neural Networks

It is well known that feed-forward neural networks (see e.g. [7, 9] for an introduction) can approximate any sufficiently well-behaved function provided that the number of units is large enough (see refs. [1, 2, 6, 8] for several theorems on the approximation of functions with neural networks). Moreover, it can be proved that, given a training sample $\{(\boldsymbol{x}_e, \boldsymbol{d}(\boldsymbol{x}_e)),\, e = 1, \ldots, N\}$, where

$$\boldsymbol{d}(\boldsymbol{x}_e) \equiv (\, \overbrace{0}^{1}\, , \ldots , \overbrace{0}^{i-1}\, , \overbrace{1}^{i}\, , \overbrace{0}^{i+1}\, , \ldots , \overbrace{0}^{m}\,) \text{ if } \boldsymbol{x}_e \in \mathcal{C}_i \,, \tag{3.1}$$

the minimum of the functional

$$E[\boldsymbol{o}] \equiv \frac{1}{2N} \sum_{e=1}^{N} \sum_{j=1}^{m} \left(o_j(\boldsymbol{x}_e) - d_j(\boldsymbol{x}_e)\right)^2 \,, \tag{3.2}$$

with respect to the unconstrained functions $\boldsymbol{o}(\boldsymbol{x})$ is achieved when these are the a-posteriori probabilities of each class [5, 10, 11, 14], i.e.

$$o_j^{(\min)}(\boldsymbol{x}) = P(\mathcal{C}_j|\boldsymbol{x}) \,,\ j = 1, \ldots, m \,. \tag{3.3}$$

Hence, if we are able to generate Monte Carlo data $\{(\boldsymbol{x}_e, \boldsymbol{d}(\boldsymbol{x}_e)),\, e = 1, \ldots, N\}$ according to certain subsample proportions $\boldsymbol{\alpha}^{\mathrm{MC}}$, and we apply for instance back-propagation (see [12, 13, 15]) to learn this data (i.e. minimize (3.2)), the resulting neural network $o_j^{(\mathrm{NN})}(\boldsymbol{x})$ would be a good approximation to (3.3), which in this case is, in fact, the projection function we were looking for:

$$o_j^{(\mathrm{NN})}(\boldsymbol{x}) = P_{\mathrm{MC}}(\mathcal{C}_j|\boldsymbol{x}) = y_j(\boldsymbol{x}) \,,\ j = 1, \ldots, m \,. \tag{3.4}$$

It must be taken into account that the above m functions are not independent because their sum must be one. However, since the minimization of (3.2) may produce a neural network solution with some amount of error, the sum may not be one. Two possible solutions are, thus, to discard one of the output units, or to normalize the output of each output unit by the sum of the outputs of all of them.

Now, the $\boldsymbol{\alpha}$ proportions can be determined by performing a simple binned log-likelihood fit (see [3]) of eq. (2.4) in the projected space of the $\boldsymbol{y}$ variables. By reducing the dimensionality of the problem from n to $m - 1$ we have won the fact that less data is needed when binning the projected space.

Since low number of events per bin might be possible, Poisson statistics per bin is assumed and the corresponding log-likelihood function is given by

$$l_y(\boldsymbol{\alpha}) = \sum_{b=1}^{B} \ln \frac{e^{-N_b(\boldsymbol{\alpha})} (N_b(\boldsymbol{\alpha}))^{n_b}}{n_b!} \,, \tag{3.5}$$

where n_b and N_b are respectively the number of observed and expected events in the b-th bin, and B is the total number of bins in the projected $\boldsymbol{y}$ space. N_b can be expressed in terms of the unknown $\boldsymbol{\alpha}$ proportions as

$$N_b(\boldsymbol{\alpha}) = N \sum_{i=1}^{m} \alpha_i P_b(\mathcal{C}_i) \,, \tag{3.6}$$

where N is the total number of available events, and $P_b(\mathcal{C}_i)$ is the probability of the i-th class events at the b-th bin. It can be approximated making use of the Monte Carlo data:

$$P_b(\mathcal{C}_i) = \int_{\mathrm{bin}(b)} d\boldsymbol{y}\, q_i(\boldsymbol{y}) \approx \frac{N_b^{\mathrm{MC}}(\mathcal{C}_i)}{N^{\mathrm{MC}}(\mathcal{C}_i)} \tag{3.7}$$

Here, $N_b^{\mathrm{MC}}(\mathcal{C}_i)$ is the number of Monte Carlo events of class $\mathcal{C}_i$ at the b-th bin, whereas $N^{\mathrm{MC}}(\mathcal{C}_i)$ is the total number of i-th class Monte Carlo events.

If the number of events per bin is large enough (let us say greater than 10 events/bin) gaussian fluctuations can be assumed, so the proportions can be estimated by minimizing the usual χ^2 function

$$\chi^2(\boldsymbol{\alpha}) = \sum_{b=1}^{B} \frac{(n_b - N_b(\boldsymbol{\alpha}))^2}{N_b(\boldsymbol{\alpha})}\,, \tag{3.8}$$

instead of maximizing the previous expression (3.5).

4 Example and results

In order to show the power of our projection method we have studied an example which is difficult to solve using the standard techniques. The problem consists in the determination of the proportions of 3 different 10-dimensional normally distributed subsamples,

$$p_i(\boldsymbol{x}) = \frac{1}{\left(\sqrt{2\pi}\right)^{10}\sqrt{\det(V_i)}} \exp\left(-\frac{1}{2}\boldsymbol{x}^T V_i^{-1}\boldsymbol{x}\right)\,,\ i = 1,2,3, \tag{4.1}$$

whose 10×10 covariance matrices V_i have ones in their diagonals and zeros otherwise, except for V_2 that has $+20\%$ correlations between its first three variables, and V_3 with those correlations equal to -20%. The difficulty associated to the tiny difference between the 3 classes is enlarged by the high dimensionality of the data, which makes the binning of the original space impossible. However, since our problem has 3 classes, our projection reduces the dimension of the fitting space from 10 to only 2.

First, we prepared a 750000 Monte Carlo data set, with $1/3 : 1/3 : 1/3$ proportions of each class, to train the neural network which had to perform the projection. The neural network used had 10 input units, 2 hidden layers with 7 and 6 units respectively, and 2 output units which corresponded to classes $\mathcal{C}_1$ and $\mathcal{C}_2$. After the training step, the neural net was able to detect the slight differences between the 3 distributions. In Fig. 1 we have plotted the difference of the 2-dimensional distributions in the projected space as expected from the analytical expressions available for this example ($q_3(\boldsymbol{y}) - q_1(\boldsymbol{y})$ and $q_2(\boldsymbol{y}) - q_1(\boldsymbol{y})$), and the ones obtained by the neural network ($q_3(\boldsymbol{o}) - q_1(\boldsymbol{o})$ and $q_2(\boldsymbol{o}) - q_1(\boldsymbol{o})$). As it can be seen, the neural network has been able to find the dirretens between the distributions in the correct place and about of the same magnitude, proving the goodness of the estimation of the $\boldsymbol{y}$ projection.

Once the projection was found, we applied it to two different mixture samples, one with proportions $1/3 : 1/3 : 1/3$ and the other with proportions $0.4 : 0.2 : 0.4$. Then, we calculated the estimated proportions given by binned likelihood fits to the projected spaces, and compared them with the estimations which can be derived by the maximization of (2.1) (this can be done in this example because we know explicitly the p.d.f.'s). The results are summarized in Tables 1 and 2, and Figs. 2 and 3 represent the one sigma contour of these estimations of α_2 and α_3 for the two mixture samples, showing a good agreement between them.

5 Conclusions

We have found the optimal projections which can be done to estimate the proportions of the subsamples of a given mixture sample, in such a way that the maximum likelihood estimator is the same and the sensitivity is not modified. These projections reduce the dimensionality of the problem from n (dimension of the data) to $m-1$ (where m is the number of classes which form the mixture sample), and they can be performed with neural networks trained on Monte Carlo data.

6 Acknowledgements

This research has been partly supported by the 'Comissionat per Universitats i Recerca de la Generalitat de Catalunya' and by EU under contract number CHRX-CT92-0004.

References

[1] A.R. Barron, IEEE Trans. Information Theory 39 (1993) 930.

[2] G. Cybenko, Math. Contr. Signals, Syst. 2 (1989) 303.

[3] W.T. Eadie, D. Drijard, F.E. James, M.Roos and B. Sadoulet, Statistical Methods in experimental Physics, (North-Holland, 1971).

[4] Ll. Garrido, V. Gaitan and M. Serra-Ricart, Comput. Phys. Commun. 84 (1994) 297.

[5] Ll. Garrido and S. Gómez, Int. J. of Neural Systems 7 (1996) 19.

[6] R. Hecht-Nielsen, Neurocomputing, (Addison-Wesley, Reading MA, 1991).

[7] J.A. Hertz, A. Krogh and R.G. Palmer, Introduction to the theory of neural computation, (Addison-Wesley, Redwood City CA, 1991).

[8] K. Hornik, M. Stinchcombe and H. White, Neural Networks 2 (1989) 359.

[9] B. Müller and J. Reinhardt, Neural networks: an introduction, (Springer-Verlag, Berlin, 1991).

[10] A. Papoulis, Probability, random variables and stochastic processes, (McGraw-Hill, New York, 1965).

[11] D.W. Ruck, S.K. Rogers, M. Kabrisky, M.E. Oxley and B.W. Suter, IEEE Trans. Neural Networks 1 (1990) 296.

[12] D.E. Rumelhart, G.E. Hinton and R.J. Williams, Nature 323 (1986) 533.

[13] D.E. Rumelhart, G.E. Hinton and R.J. Williams, in: Parallel Distributed Processing, eds. D.E. Rumelhart and J.L. McClelland, (MIT Press, Vol. 1, Cambridge MA, 1986) p. 318.

[14] E.A. Wan, IEEE Trans. Neural Networks 1 (1990) 303.

[15] P. Werbos, Beyond regression: new tools for prediction and analysis in the behavioral sciences, (Ph.D. thesis, Harvard University, 1974).

Figure captions

- **Figure 1:** Difference between the distributions in the projected space, the expected ones above and the obtained with the neural network below.
- **Figure 2:** Contours of one sigma level, for both proportions simultaneously, estimated by a log-likelihood fit to the true p.d.f.'s (solid line) and by a binned log-likelihood fit in the bidimensional neural network output space (dashed line). The real proportions of each class are $1/3 : 1/3 : 1/3$.
- **Figure 3:** Contours of one sigma level, for both proportions simultaneously, estimated by a log-likelihood fit to the true p.d.f.'s (solid line) and by a binned log-likelihood fit in the bidimensional neural network output space (dashed line). The real proportions of each class are $0.4 : 0.2 : 0.4$.

Method	α_2	α_3	Correlation
Likelihood fit to true p.d.f.'s	0.3267 ± 0.0097	0.3205 ± 0.0089	0.822
Binned Likelihood fit to $\boldsymbol{o}(\boldsymbol{x})$	0.3288 ± 0.0105	0.3230 ± 0.0097	0.845

Table 1: Estimated proportions for classes $\mathcal{C}_2$ and $\mathcal{C}_3$ and correlation between them using the two methods described in the text. The real proportions are $1/3 : 1/3 : 1/3$ over 300000 data.

Method	α_2	α_3	Correlation
Likelihood fit to true p.d.f.'s	0.1929 ± 0.0098	0.3874 ± 0.0095	0.817
Binned Likelihood fit to $\boldsymbol{o}(\boldsymbol{x})$	0.1933 ± 0.0106	0.3870 ± 0.0104	0.841

Table 2: Estimated proportions for classes $\mathcal{C}_2$ and $\mathcal{C}_3$ and correlation between them using the two methods described in the text. The real proportions are $0.4 : 0.2 : 0.4$ over 250000 data.

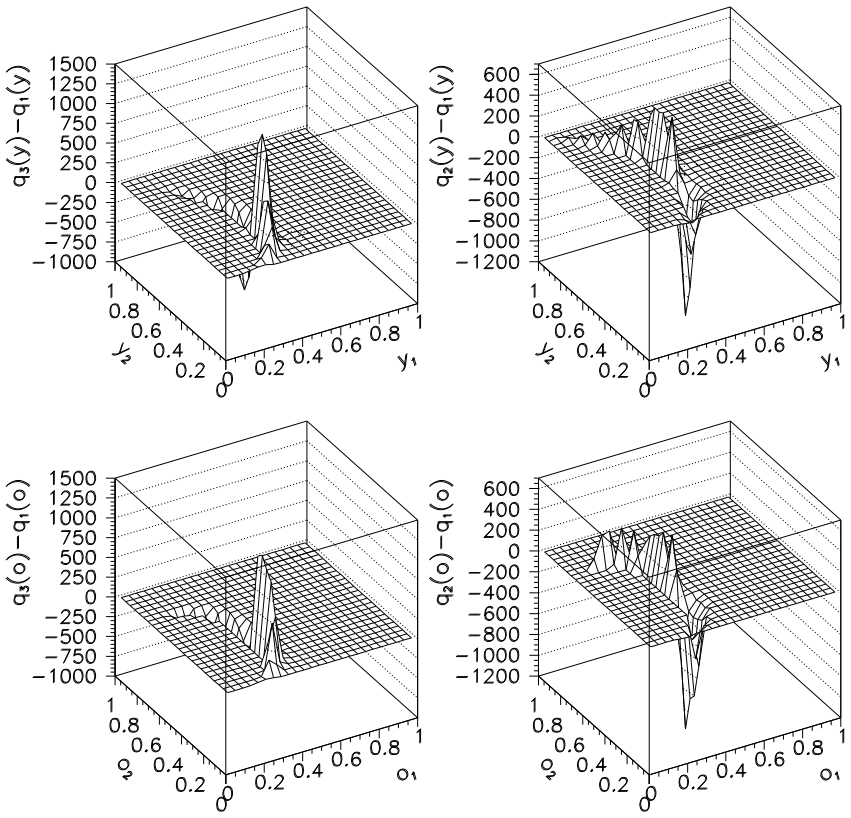

Figure 1: Difference between the distributions in the projected space, the expected ones above and the obtained with the neural network below.

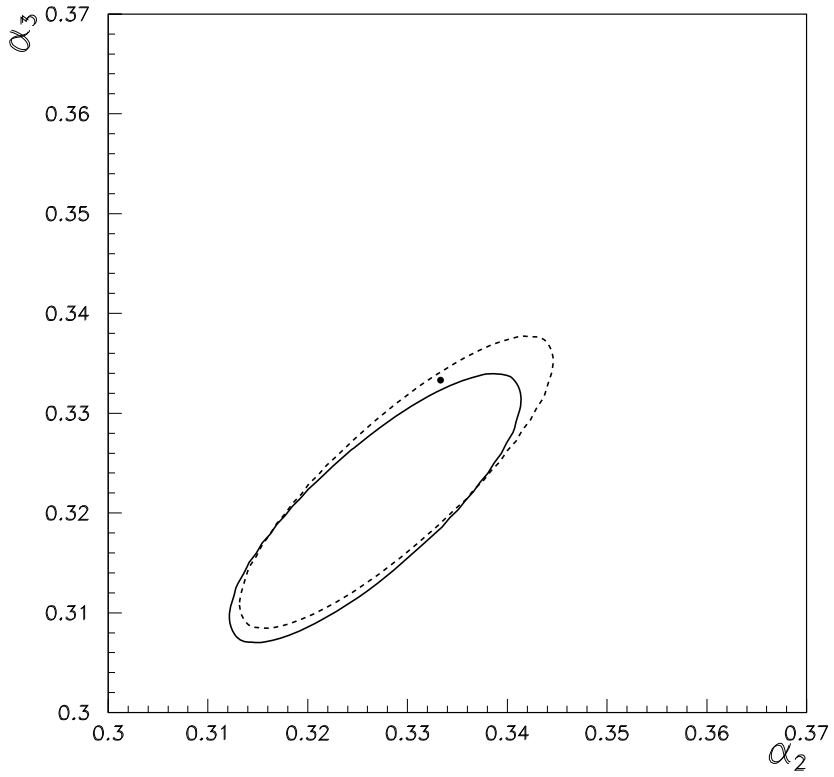

Figure 2: Contours of one sigma level, for both proportions simultaneously, estimated by a log-likelihood fit to the true p.d.f.'s (solid line) and by a binned log-likelihood fit in the bidimensional neural network output space (dashed line). The real proportions of each class are $1/3 : 1/3 : 1/3$.

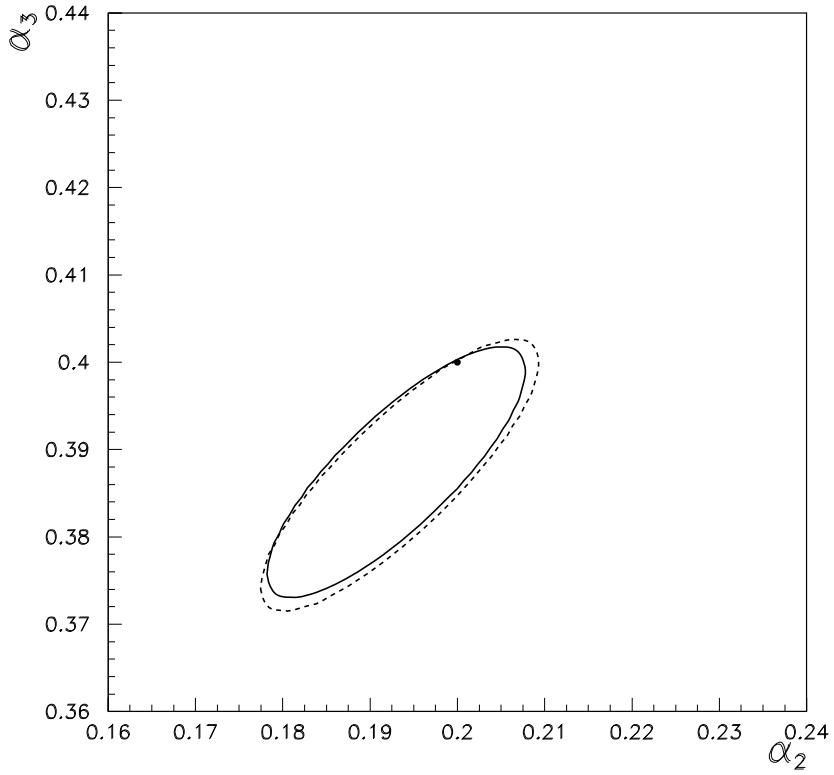

Figure 3: Contours of one sigma level, for both proportions simultaneously, estimated by a log-likelihood fit to the true p.d.f.'s (solid line) and by a binned log-likelihood fit in the bidimensional neural network output space (dashed line). The real proportions of each class are $0.4 : 0.2 : 0.4$.