Guiding Local Regression using Visualisation

Dharmesh M. Maniyar and Ian T. Nabney

Neural Computing Research Group, Aston University, Birmingham B4 7ET, UK,
maniyard@aston.ac.uk,
http://www.ncrg.aston.ac.uk/

Abstract. Solving many scientific problems requires effective regression and/or classification models for large high-dimensional datasets. Experts from these problem domains (*e.g.* biologists, chemists, financial analysts) have insights into the domain which can be helpful in developing powerful models but they need a modelling framework that helps them to use these insights. Data visualisation is an effective technique for presenting data and requiring feedback from the experts. A single global regression model can rarely capture the full behavioural variability of a huge multi-dimensional dataset. Instead, local regression models, each focused on a separate area of input space, often work better since the behaviour of different areas may vary. Classical local models such as Mixture of Experts segment the input space automatically, which is not always effective and it also lacks involvement of the domain experts to guide a meaningful segmentation of the input space. In this paper we addresses this issue by allowing domain experts to interactively segment the input space using data visualisation. The segmentation output obtained is then further used to develop effective local regression models.

1 Introduction

The work presented here was motivated by a problem in the Chemoinformatics domain where there is a need for a computational model that relates physico-chemical properties of compounds with their biological activity. A reliable regression model would allow a screening scientist to predict the biological activity of compounds and then decide which compounds are worth physically testing.

There are many regression techniques available from the statistical and neural computing domains. Broadly they can be divided into global and local regression models. Global models use a single model for the problem which covers the entire input space. Local regression models use a combination of models, each of which applies to a smaller part of the input space.

Because of the quantity and diversity of data points (e.g. a huge chemical compound library), trying to develop a single model to make prediction for all data points (e.g. chemical compounds in a library) is unlikely to succeed. What is more likely to be effective is a group of local models, each of which working on a set of similar data points, in other words, in different regions of the input space. In this paper, we present a *guided* local regression approach which first, with the help of principled visualisation techniques, allows domain experts to create

an *informed segmentation* of the input space. Then, we use that segmentation output to develop local regression models. We compare our results with the results from classical global and local regression models.

The next section briefly describes the Mixture of Experts (ME) model since it is related to the guided regression models we introduce here. Section 3 gives a brief introduction to the Hierarchical Generative Topographic Map (HGTM) which we use for visualisation and segmentation. In Section 4 we present the guided local regression models. The experimental results are reported in Section 5. Finally, the paper ends with a discussion in Section 6.

2 Mixture of Experts (ME)

Jacobs et *al.* introduced the mixture of experts model, which determines a decomposition of the data as a part of the learning process [1]. In this model, all of the expert networks, as well as a gating network, are trained together. The goal of the training procedure is to have the gating network learn an appropriate decomposition of the input space into different regions, while each expert network learns to generate the outputs for input vectors falling within a specific region. The gating network outputs $g_i(\mathbf{x})$ can be regarded as the probability that input $\mathbf{x}$ is attributed to expert i. This probabilistic interpretation is ensured because of the choice of output for the gating network is the softmax activation function:

$$g_i = \frac{\exp(\gamma_i)}{\sum_{j=1}^{M} \exp(\gamma_i)}, \tag{1}$$

where the $\gamma_i (i = 1, 2, ..., M)$ are the outputs of the gating network and M is the number of experts.

The error function for the complete model is given by the negative logarithm of the likelihood with respect to a probability distribution given by a mixture of M Gaussians of the form

$$E = -\sum_{n} \ln \left\{ \sum_{i=1}^{M} g_i(\mathbf{x}^n) \phi_i(\mathbf{t}^n \mid \mathbf{x}^n) \right\}, \tag{2}$$

where $\mathbf{t}$ is the output vector and the $\phi_i(\mathbf{t} \mid \mathbf{x})$ are regression models with Gaussian noise.

When the trained network is used to make predictions, the input vector is presented to the gating network and all of the expert networks. The output vector of a ME is the weighted mean (with weighting given by the gating network outputs) of the expert outputs:

$$\mathbf{y}(\mathbf{x}) = \sum_{i=1}^{M} g_i(\mathbf{x}) \phi_i(\mathbf{x}). \tag{3}$$

The mixture of experts network is trained by minimising the error function (2) simultaneously with respect to the weights in all of the expert networks and in

the gating network. The standard choices for gating and expert networks are generalised linear models (GLM) and multi-layer perceptrons (MLP).

3 Hierarchical Generative Topographic Map (HGTM)

The HGTM [2] is a probabilistic model that provides a hierarchical visualisation of data. It arranges a set of GTMs [3] and their corresponding plots in a tree structure $\mathcal{T}$. The GTM models a probability distribution in the high-dimensional data space by means of a low-dimensional (usually 2-dimensional) latent space.

- In GTM, the non-linear transformation, $f : \mathcal{H} \Rightarrow \mathcal{D}$, from the latent space to the data space is defined using a Radial Basis Function (RBF) network with weights $\mathbf{W}$. The density in the latent space is defined as a sum of delta functions centred on nodes $\mathbf{k}_i$. The unconditional probability of a data point $\mathbf{x}$ is given by a mixture

$$p(\mathbf{x} \mid \mathbf{W}, \beta) = \frac{1}{M} \sum_{i=1}^{M} p(\mathbf{x} \mid \mathbf{k}_i, \mathbf{W}, \beta), \tag{4}$$

  where the ith component density is a Gaussian distribution whose mean is the image of $\mathbf{k}_i$ under f with inverse variance β.
- Bayes' theorem is used to invert the transformation f. The posterior probability $R_{i,n}$ (responsibility) that the ith Gaussian generated the point $\mathbf{x}_n$, is given by

$$R_{i,n} = \frac{P(\mathbf{x}_n \mid \mathbf{k}_i, \mathbf{W}, \beta)}{\sum_{j=1}^{C} P(\mathbf{x}_n \mid \mathbf{x}_j, \mathbf{W}, \beta)} \tag{5}$$

  In order to visualise a whole dataset in a single plot, the latent space representation of the point $\mathbf{x}_n$ is taken to be the mean, $\sum_{i=1}^{C} R_{i,n}\mathbf{k}_i$, of the posterior distribution on $\mathcal{H}$ where C is total number of latent space centres.

An example HGTM structure is shown in the Figure 1. In this section we give a general formulation of hierarchical GTM, more details can be found in [2].

The *Root* of the hierarchy is at level 1, i.e. *Level(Root)* $= 1$. Children of a model $\mathcal{N}$ with *Level*$(\mathcal{N}) = \ell$ are at level $\ell + 1$, i.e. *Level*$(\mathcal{M}) = \ell + 1$, for all $\mathcal{M} \in$ *Children*$(\mathcal{N})$. Each model $\mathcal{M}$ in the hierarchy, except for *Root*, has an associated non-negative parent-conditional mixture coefficient, or prior $\pi(\mathcal{M} \mid \mathit{Parent}(\mathcal{M}))$. The priors satisfy the consistency condition: $\sum_{\mathcal{M} \in \mathit{Children}(\mathcal{N})} \pi(\mathcal{M} \mid \mathcal{N}) = 1$. Unconditional priors for the models are recursively calculated as: $\pi(\mathit{Root}) = 1$, and for all other models

$$\pi(\mathcal{M}) = \prod_{i=2}^{\mathit{Level}(\mathcal{M})} \pi(\mathit{Path}(\mathcal{M})_i \mid \mathit{Path}(\mathcal{M})_{i-1}), \tag{6}$$

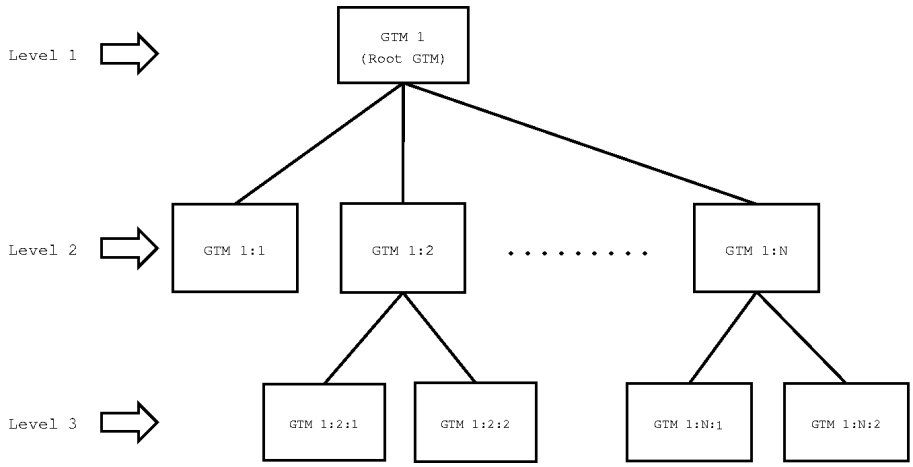

Fig. 1. Example plot structure for HGTM. Each model corresponds to a visualisation.

where $Path(\mathcal{M}) = (Root, ..., \mathcal{M})$ is the N-tuple ($N = Level(\mathcal{M})$) of nodes defining the path in $\mathcal{T}$ from *Root* to $\mathcal{M}$.

The distribution given by the hierarchical model is a mixture of leaf models of $\mathcal{T}$,

$$P(\mathbf{x} \mid \mathcal{T}) = \sum_{\mathcal{M} \in Leaves(\mathcal{T})} \pi(\mathcal{M}) P(\mathbf{x} \mid \mathcal{M}). \tag{7}$$

Non-leaf models not only play a role in the process of creating the hierarchical model, but in the context of data visualization can be useful for determining the relationship between related subplots in the hierarchy.

The hierarchical GTM is trained using the EM algorithm to maximize its likelihood with respect to the data sample $\varsigma = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$. Training of a hierarchy of GTMs proceeds in a recursive fashion. First, a base (*Root*) GTM is trained and used to visualise the data. Then the user identifies interesting regions on the visualization plot that they would like to model in greater detail. In particular, the user chooses a collection of points, $c_i \in \mathcal{H}$, by clicking on the plot. These points are used to initialise the next level of GTMs. Voronoi compartments [4] are defined in the data space by the mapped points $f_{Root}(c_i) \in \mathcal{D}$, where f_{Root} is the map of the *Root* GTM. The child GTMs are initialised by local PCA in the corresponding Voronoi compartments. After training the child GTMs and seeing the lower level visualization plots, the user may decide to proceed further and model in greater detail some portions of the lower level plots, etc. At each stage of the construction of an hierarchical GTM, the EM algorithm alternates between the E- and M-steps until convergence is satisfactory (typically after 10-20 iterations).

We can calculate magnification factors using the Jacobian of the GTM map f [5]. Magnification factor plots are used to observe the amount of stretching

in a GTM manifold on different parts of the latent space which helps in outlier detection and cluster separation. Tiňo *et. al.* [6] derived a closed-form formula for directional curvature of the GTM projection manifold. Directional curvature plots allow the user to observe folding in the GTM manifold. Magnification factors and directional curvatures help the user to decide where to place submodels.

We have developed an interactive software tool which allows a user to see the magnification factor and directional curvature plots with the actual HGTM visualisation. The software also provides a *parallel coordinate* facility to let the user explore patterns of a few neighbouring points (determined using Euclidean distance) from the point selected by the user in the latent space. This is useful for understanding different regions of the latent space as the user can observe the corresponding data space patterns. The tool can be used by domain experts to understand and segment vast data.

4 Guided Local Regression Models

The divide-and-conquer approach used in ME discussed in Section 2 can particularly prove useful in modeling diversities in the input-output mapping. One of the most important issues in applying a divide-and-conquer strategy is to find the different regions to divide the input space. Doing it automatically as in ME might not be effective for a complex dataset.

One of the main differences between the mixture of experts and the guided regression models presented in this section, is the way of segmenting the input space. In ME, the gating network learns a decomposition of the input space into different regions with the training of expert models, while in the guided local regression models, we let the domain experts interactively decide the decomposition of the input space using a visualisation algorithm and other visualisation aids, such as magnification factors, directional curvature and parallel coordinates. Thus the segmentation process here is not *automatic* as in ME but it is *guided* by the domain experts.

In this paper we only use a 2-level HGTM tree structure for simplicity, but the results can be extended to an HGTM of any depth. Consider an HGTM tree structure, $\mathcal{T}$, as in Figure 1.

Model responsibilities, $\mathbf{R}$, corresponding to all the models, $\mathcal{M}_i$, $i = 1, \ldots, M$, in the HGTM tree structure, $\mathcal{T}$, are calculated as follows:

$$R_{i,n} = P(\mathcal{M}_i \mid \mathit{Parent}(\mathcal{M}_i), \mathbf{x}_n) = \frac{\pi(\mathcal{M}_i \mid \mathit{Parent}(\mathcal{M}_i))P(\mathbf{x}_n \mid \mathcal{M}_i)}{\sum_{\mathcal{N} \in [\mathcal{M}_i]} \pi(\mathcal{N} \mid \mathit{Parent}(\mathcal{M}_i))P(\mathbf{x}_n \mid \mathcal{N})}, \tag{8}$$

where $[\mathcal{M}_i] = \mathit{Children}(\mathit{Parent}(\mathcal{M}_i))$.

Imposing $P(\mathit{Root} \mid \mathbf{x}_n) = 1$, the unconditional (on parent) model responsibilities are recursively determined by the formula:

$$P(\mathcal{M} \mid \mathbf{x}_n) = P(\mathcal{M} \mid \mathit{Parent}(\mathcal{M}), \mathbf{x}_n)P(\mathit{Parent}(\mathcal{M}) \mid \mathbf{x}_n). \tag{9}$$

The model responsibility matrix, $\mathbf{R}$, has the property

$$\sum_{i=1}^{M} R_{i,n} = 1 \qquad \forall\ n. \tag{10}$$

Equation (10) confirms the *soft* segmentation of the input space we obtain from the HGTM model. It is similar to the segmentation derived from the softmax function in the trained gating network in the ME (eq. 1). The soft segmentation obtained using HGTM is non-linear, so the segmentation regions can have an arbitrary shape. The individual experts can arbitrarily be linear or non-linear regression models. The trained HGTM model is then used to train local regression model, which we name as Guided Mixture of Experts (GME), as specified in Procedure 1. Notice that in step 2, for the training of a local expert, using the model responsibility obtained by the trained HGTM model, we select only those data points which belong to a particular local region. It means that only those data points which lie in a particular local region are used to train the expert responsible for modelling that region. In the work presented here, during the training of a local expert, we do not weight data points with their corresponding model responsibility. One of our future extensions will be to use responsibilities for weighting during the training. We have already implemented a weighted Generalised Linear Model.

Procedure 1 (Training). *1. Using a previously trained HGTM visualisation model, calculate the model responsibility matrix,* $\mathbf{R}$*, for all the training points for all leaves (eq. 8).*

2. *Train an expert regression model corresponding to each leaf node. Each expert,* $\phi_i(\mathbf{t} \mid \mathbf{x})$*, is trained individually on all the training points,* $\mathbf{x}_n$*, for which* $R_{i,n}$ *is greater than a threshold. Different thresholds can be tried and validated.*
3. *During the training of each expert,* $\phi_i(\mathbf{t} \mid \mathbf{x})$*, possible best architecture is selected through validation on the local points it is responsible for.*

While making predictions for new inputs, in ME, we present inputs to all of the experts and the gating network. The outputs of the experts are weighted by the output of the gating network and summed (eq. 3). In the GME, the inputs are first presented to a trained HGTM visualisation model and responsibilities for each expert are calculated using (eq. 8). Then the output of each expert is weighted by the corresponding responsibility and finally summed as shown in Figure 2. The prediction (testing) procedure, using a trained GME model, is given below:

Procedure 2 (Testing). *1. Calculate the model responsibility matrix,* $\mathbf{R}$*, for all the testing points using the trained HGTM model stored with the trained GME model.*

2. *Each trained expert is presented with all the inputs (see Figure 2). All experts produce the outputs for the input point,* $\mathbf{x}_n$*, which are then weighted by the corresponding model responsibilities and summed to get the final output for*

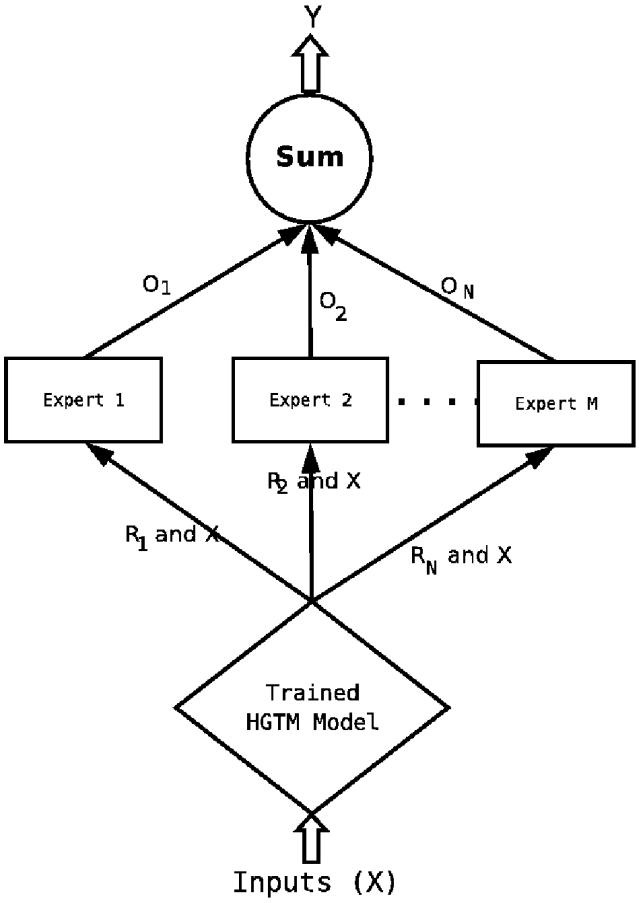

Fig. 2. Architecture of Guided Mixture of Experts (GME)

.

that particular input.

$$\mathbf{y}^n = \sum_{i=1}^{M} R_{i,n}\phi_i(\mathbf{t}^n \mid \mathbf{x}^n), \tag{11}$$

where $\phi_i(\mathbf{t}^n \mid \mathbf{x}^n)$ *is the output from the trained expert* i*.*

5 Results

Two experiments were carried out: one with a synthetic dataset and one with Chemoinformatics data.

5.1 Synthetic Dataset

The data set consisted of around 2900 points, $\mathbf{x} = (x_1, x_2, x_3)^T$ lying on a two-dimensional manifold in the three-dimensional Euclidean space. The manifold is shown in Figure 3 and is described by the equation

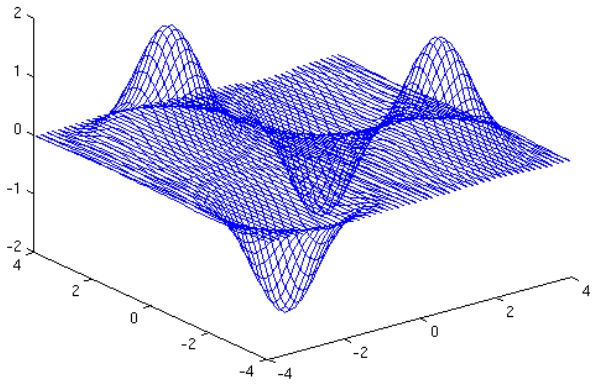

Fig. 3. A two-dimensional manifold in three-dimensional Euclidean space

$$x_3 = 2 \sum_{c_1, c_2 \in \{-2,2\}} \exp\{-(x_1 - c_1)^2 - (x_2 - c_2)^2\}, \quad (x_1, x_2) \in [-4, 4]^2. \tag{12}$$

To have a different mapping in each "hump", we define the following functions:

$$\begin{aligned}
y &= x_1 - x_2^2 - x_3 \quad \forall x_1, x_2, x_3 \quad 0 < x_1 < 4,\ 0 < x_2 < 4, \text{ and } -2 < x_3 < 0,\\
y &= x_1^2 + x_2 + x_3 \quad \forall x_1, x_2, x_3 \quad -4 < x_1 < 0,\ 0 < x_2 < 4, \text{ and } 0 < x_3 < 2,\\
y &= x_1 + x_2 - x_3^2 \quad \forall x_1, x_2, x_3 \quad -4 < x_1 < 0,\ -4 < x_2 < 0, \text{ and } -2 < x_3 < 0,\\
y &= x_1 - x_2^2 + x_3^2 \quad \forall x_1, x_2, x_3 \quad 0 < x_1 < 4,\ -4 < x_2 < 0, \text{ and } 0 < x_3 < 2.
\end{aligned}$$

From the total dataset of around 2900 data points, 80% of the points were used as the training set and rest were kept aside for testing. 20% of the training set was used for validation to choose the model architecture. Figure 4 shows a trained HGTM output on the testing set of the synthetic dataset.

We trained models with different complexities for MLP, ME and GME. The validation set error was calculated for all the models and, for each architecture,

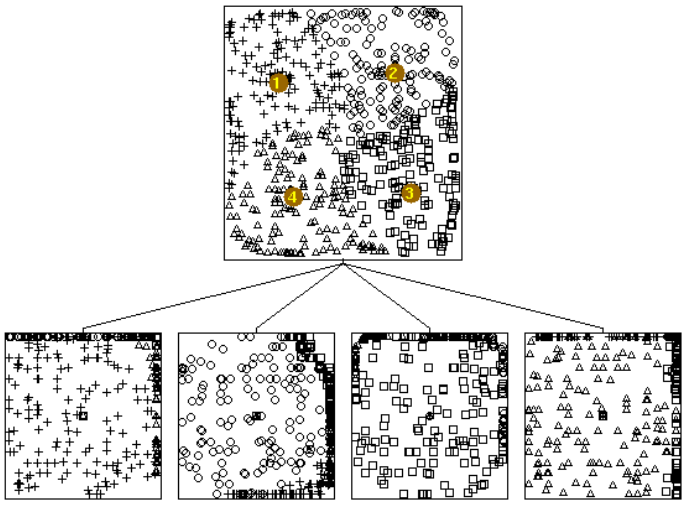

Fig. 4. HGTM visualisation output for the testing set of synthetic data

the model with the minimum validation set error was selected. The selected model from a given class was then trained on the whole training set (including the validation set).

To analyse the properties of the input-space segmentation obtained from ME and GME, we measure its average entropy [7]. The average entropy was calculated as below:

$$H = -\frac{1}{M}\sum_{m=1}^{M}\frac{1}{N}\sum_{n=1}^{N} P_m(\mathbf{x}_n)\log P_m(\mathbf{x}_n), \tag{13}$$

where $P_m(\mathbf{x}_n)\log P_m(\mathbf{x}_n)$ is defined as 0 if $P_m(\mathbf{x}_n) = 0$. For ME, $P_m(\mathbf{x}_n)$ is the output of the gating network for the mth expert, and input point x_n, while for GME, $P_m(\mathbf{x}_n)$ is the model responsibility, $R_{m,n}$. For the selected architectures, the average entropy values for ME and GME were obtained as 0.0621 and 0.0058 respectively. These values reveal that the ME gives a comparatively soft segmentation with more overlaps, while the GME provides a harder segmentation which separates the input space in to distinct different regions with little overlap which is also easier for the domain experts to interpret.

Table 1 presents the normalised mean squared error (NMSE) [8] we obtained for the training and the test sets. The 4th column in Table 1 displays the t-test significance value compared with the result of the GME. The t-test assesses whether the means of two groups are statistically different from each other [9]. The smaller the value, the more significant the difference between the means. Information about which model architecture was selected, using the validation set, is given in the last column. We note that the GME result is significantly better than the MLP and ME.

Table 1. Regression results for the synthetic dataset

Model	Training NMSE	Testing NMSE	P-value	Architecture
MLP	0.1009	0.0968	7.5816e-48	$N_{hid} = 21$
ME	0.0433	0.0466	0.0021	$N_{experts} = 11$
GME	0.0234	0.0227	-	$N_{experts} = 4$

5.2 Chemoinformatics Data

The second experiment was carried out on a real life problem in the Chemoinformatics domain where we need to predict the biological activity of chemical compounds, for a particular target, from 11 physicochemical properties of the compounds. The dataset (of around 20700 chemical compounds selected randomly from around 1000000 compounds) was divided equally into training and testing sets. 20% of the training set was kept aside for validation to choose the model architecture. Figure 5 presents the HGTM visualisation output for a subset (random 600 compounds, 300 active and 300 inactive) of testing set.

Table 2. Regression results for biological activity prediction

Model	Training NMSE	Testing NMSE	P-value	Architecture
MLP	0.8439	0.8458	0.0129	$N_{hid} = 25$
ME	0.8370	0.8405	0.0200	$N_{experts} = 12$
GME	0.8104	0.8214	-	$N_{experts} = 7$

The results are presented in Table 2. For the selected architectures, the average entropy values for ME and GME were obtained as 0.1953 and 0.0298 respectively which demonstrates better segmentation obtained by GME. The GME result is better than the two models, though only at a level of 2%.

6 Discussion

Our approach of using visualisation output to develop guided local regression models has given better results than the classical ME. That is in line with our assumption that the segmentation obtained from principled visualisation algorithms, such as HGTM, can be sensibly used for the development of new local regression models.

The advantage of the approach is that the informed segmentation is obtained with the help of domain experts who have some understanding of the data in this way, domain experts are more involved in the model development process. The disadvantage of GME is that, as a 2 stage process, it requires user interactions and thus it takes comparatively more time to develop a new model.

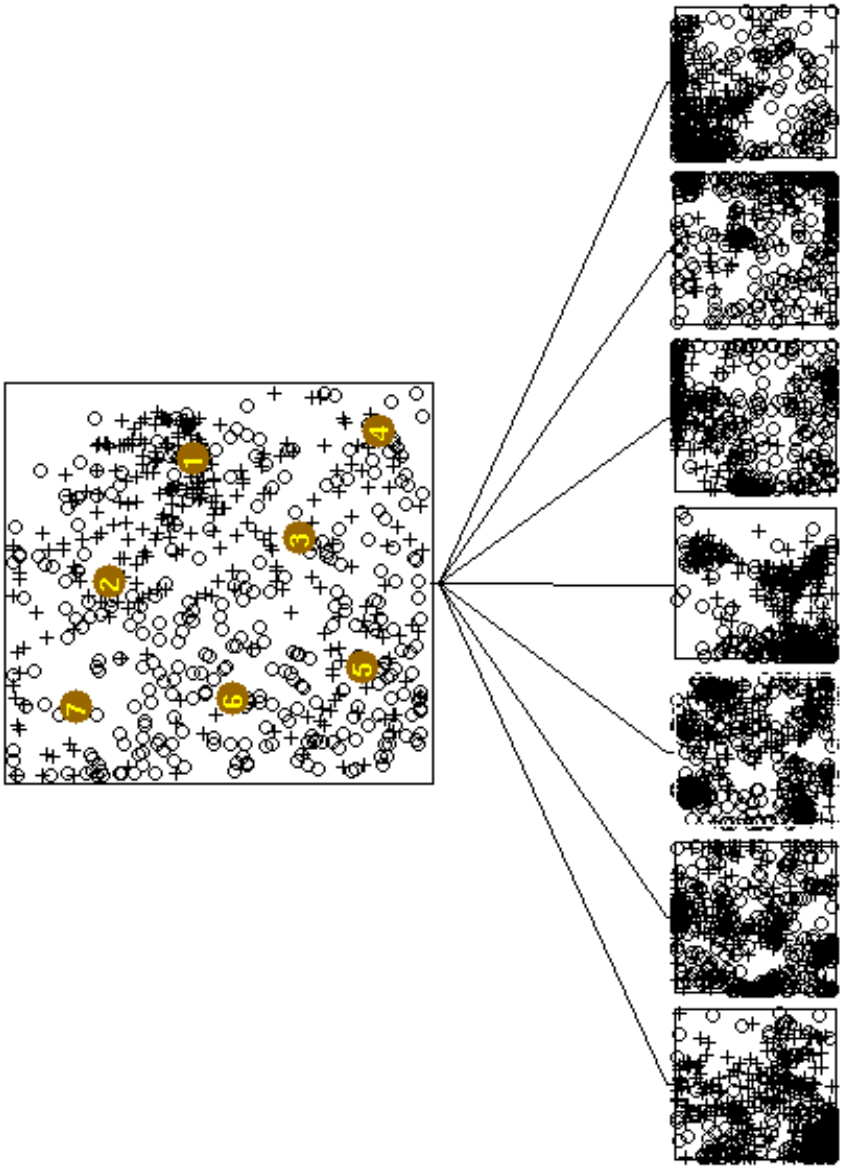

Fig. 5. HGTM output for the subset of the testing set of chemoinformatics data

Overall, for the synthetic dataset, the local regression models gave better results than the global regression model. We believe that the principal local regression models, such as ME and GME, will perform well for high dimensional diverse datasets generally found in drug discovery and bioinformatics domains.

However, the experiments on chemical compounds gave NMSE of more than 0.8 which is not satisfactory for practical use. The relatively high value of NMSE indicates that the models are close to predicting "in the mean" [8]. After discussion with screening scientists, it was realised that the descriptors (11 physicochemical properties) used to predict the biological activities do not contain enough information to make a robust prediction. Using structure information of chemical compounds should help in improving regression models performance since the pharmacophore[1] of compounds plays an important role in making a compound active for a target [10].

Acknowledgements

DM is grateful to Pfizer Central Research for their financial support. We thank Bruce S. Williams and Andreas Sewing for useful discussions on the results with the Chemioinformatics data.

References

1. R. A. Jacobs, M. I. Jordan, S. J. Nowlan, G. E. Hinton, Adaptive mixture of local experts, Neural Computation 3 (1991) 79–87.
2. P. Tiňo, I. T. Nabney, Constructing localized non-linear projection manifolds in a principled way: hierarchical generative topographic mapping., IEEE T. Pattern Analysis and Machine Intelligence 24 (2002) 639–656.
3. C. M. Bishop, M. Svensén, C. K. I. Williams, GTM: The generative topographic mapping, Neural Computation 10 (1998) 215–234.
4. F. Aurenhammer, Voronoi diagrams - survey of a fundamental geometric data structure", ACM Computing Surveys 3 (1991) 345–405.
5. C. M. Bishop, M. Svensén, C. K. I. Williams, Magnification factors for the GTM algorithm, Proceedings IEE Fifth International Conference on Artificial Neural Networks (1997) 64–69.
6. P. Tiňo, I. T. Nabney, Y. Sun, Using directional curvatures to visualize folding patterns of the GTM projection manifolds, Artificial Neural Networks - ICANN (eds) G. Dorffner, H. Bischof and K. Hornik (2001) 421–428.
7. R. Ellis, Entropy, Large Deviations, and Statistical Mechanics, Springer-Verlag, New York, 1985.
8. C. M. Bishop, Neural Networks for Pattern Recognition, 1st Edition, Oxford University Press, 1995.
9. N. Weiss, Elementary Statistics, 3rd Edition, Addison Wesley, 1996.
10. A. C. Good, S. R. Krystek, J. S. Mason, High-throughput and virtual screening: core lead discovery techologies move towards integration, Drug Discovery Today 5 (2000) S61–S69.

[1] A specific arrangement of chemical groups in a compound that are essential for recognition by a target.