GTM: The Generative Topographic Mapping

Presenter
Folami Alamudun

Authors
Chrisopher M. Bishop
Markus Svensen
Christopher K.I. Williams

Introduction
GTM Model
EM Algorithm for GTM
 Summary of learning algorithm
Experimental Results
Relationship to other Models
Discussion
Related work

- Chris Bishop is Chief Research Scientist at Microsoft Research Cambridge, and Professor of Computer Science at the University of Edinburgh.
- He is a Fellow of the Royal Academy of Engineering, a Fellow of the Royal Society of Edinburgh, and a Fellow of Darwin College Cambridge. His research interests include machine learning and its applications.

Related work

What?

- Generative Topographic mapping (GTM) is a novel non-linear latent variable model.

Why?

- GTM seeks an explanation to the behavior of a number of data variables in terms of a smaller number of latent variables.

How?

- GTM allows for a non-linear relationship between latent and observed variables

Introduction

Introduction

- What is a latent variable model?
  - Methodic representation of multidimensional data in fewer dimensions using latent variables
- What are other examples of latent variable models?
  - Factor Analysis
  - Probabilistic Principal Component Analysis.

Introduction
GTM Model
EM Algorithm for GTM
Summary of learning algorithm
Experimental Results
Relationship to other Models
Discussion
Related work

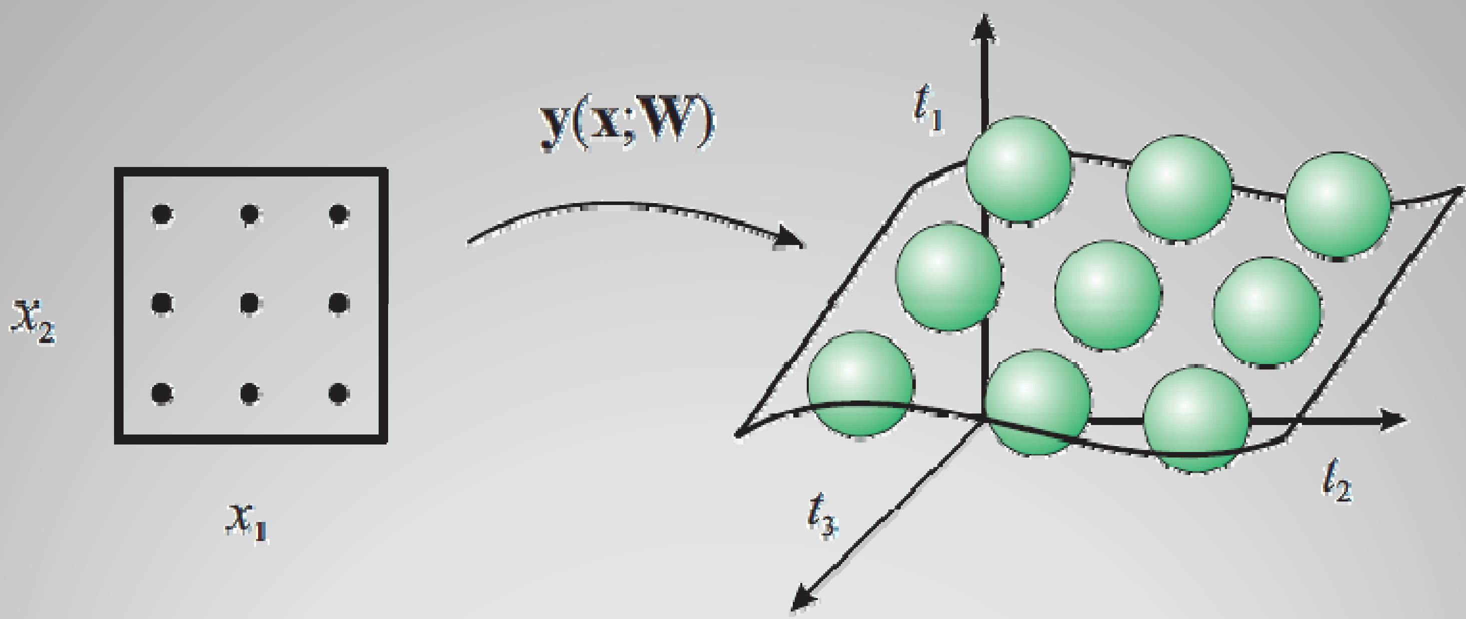
- The goal of GTM is to define a probability distribution, *p(t)*, over D-dimensional space (D-space) $\mathbf{t} = (t_1, t_2, t_3,.., t_D)$ in terms of latent variables $\mathbf{x} = (x_1, x_2, x_3,.., x_L)$.

- GTM uses a non-linear, parametric function *y(x, W)* which maps every point in the latent space to a point in the data space
  - *from* L-dimensional **L**-space ($x \in \Re^L$)
  - *to* a corresponding point ($y \in \Re^D$) in **D**-space

where:

- $L < D$

GTM Model



GTM Model

- We define probability distribution over **L**-space as p(x), then probability distribution over **D**-space convolved with an isotropic Gaussian noise distribution can be given by:

$$p(\mathbf{t} \mid \mathbf{x}, W, \beta) = \mathcal{N}(y(\mathbf{x}, W), \beta)$$

$$= \left(\frac{\beta}{2\pi}\right)^{\frac{D}{2}} exp\left\{-\frac{\beta}{2}\|y(\mathbf{x}, \boldsymbol{W}) - \mathbf{t}\|^2\right\}$$

  where
  - T is a point in data space; and
  - β^{-1} denotes the noise variance.

GTM Model

- For a given value of *W*, the distribution in D-space is given by:

$$p(\boldsymbol{t} \mid \boldsymbol{W}, \beta) = \int p(\boldsymbol{t} \mid \boldsymbol{x}, \boldsymbol{W}, \beta)\, p(\boldsymbol{x}) dx$$

- For a given dataset of N data points, $\boldsymbol{\mathcal{D}} = (t_1, t_2, t_3, .., t_D)$, the parameter matrix *W*, and inverse matrix β are obtained by maximizing the log likelihood $\mathcal{L}(W, \beta)$ given by:

$$\mathcal{L}(\boldsymbol{W}, \beta) = \ln \prod_{n=1}^{N} p(\boldsymbol{t}_n \mid \boldsymbol{W}, \beta)$$

GTM Model

- For analytical tractability, we use a set of K equally weighted delta functions on a regular grid to represent p(x). The log likelihood function becomes:

$$\mathcal{L}(\mathbf{W},\beta) = \sum_{n=1}^{N} \ln \left\{ \frac{1}{K} \sum_{k}^{K} p(\mathbf{t}_n | \mathbf{x}_k, \mathbf{W}, \beta) \right\}$$

GTM Model

Introduction
GTM Model
EM Algorithm for GTM
Summary of learning algorithm
Experimental Results
Relationship to other Models
Discussion
Related work

EM Algorithm for GTM

The Expectation maximization algorithm works in two steps:

- E-step: Uses W_{old} and β_{old} to calculate the responsibility (posterior probabilities) for each Gaussian component i for all data points t_n

$$\mathcal{L}_{comp}(W, \beta) - \sum_{n=1}^{N} \sum_{i=1}^{K} \mathrm{R}_{in}(W_{\text{old}}, \beta_{\text{old}}) \ln\{p(t_n | x_i, W, \beta)\}$$

$$y_d(\mathbf{x}, \mathbf{W}) = \sum_{m}^{M} \phi_m(\mathbf{x}) w_{md}$$

- *W* is a M x D matrix containing weight and bias parameters

$$\phi_m(\mathbf{x}) = \begin{cases} \exp\left\{ -\frac{\|\mathbf{x}-\boldsymbol{\mu}_m\|^2}{2\sigma^2} \right\} & \text{if } m \leq M_{\mathrm{NL}}, \\ x^l & \text{if } m = M_{\mathrm{NL}} + l,\ l = 1, \ldots, L \\ 1 & \text{if } m = M_{\mathrm{NL}} + L + 1 = M, \end{cases}$$

  - M_{NL} non-linear basis functions in the form of non-normalized Gaussian basis functions.
  - L linear basis functions - for capturing linear trends in the data.
  - One fixed basis function that allows the corresponding weights to act as biases.

EM Algorithm for GTM

- M-step calculates W_{new} and β_{new} from the maximized log likelihood $\mathcal{L}(W,\beta)$ equations given by:
  - W_{new} from:

$$\Phi^T G_{old} \Phi W_{new}^T = \Phi^T R_{old} T$$

  - β_{new} from

$$\frac{1}{\beta_{new}} = \frac{1}{ND} \sum_{n=1}^{N} \sum_{i=1}^{K} \mathrm{R}_{in}(W_{old}, \beta_{old}) \| W_{new} \Phi(x_i) - t_n \|^2$$

  - **T** is an N x D matrix with elements T_{nk};
  - R is a K x N matrix with elements R_{in};
  - G is a K x K diagonal matrix: $G_{ii} = \sum_{n=1}^{N} R_{in}(W, \beta)$

EM Algorithm for GTM

- Introduction
- GTM Model
- EM Algorithm for GTM
- **Summary of learning algorithm**
- Experimental Results
- Relationship to other Models
- Discussion
- Related work

Summary of GTM algorithm

- Generate the grid of latent points $\{x_k\}$ $k = 1,\ldots,K$
- Generate the grid of basis function centres $\{\mu_m\}$ $m = 1,\ldots, M$
- Select the basis function width σ
- Compute the matrix of basis function activations Φ
- Initialize W randomly or using PCA
- Initialize β
- Train by alternating between E-step and M-step.
  - Evaluate log likelihood at the end of each cycle for convergence.

Introduction
GTM Model
EM Algorithm for GTM
Summary of learning algorithm
Experimental Results
Relationship to other Models
Discussion
Related work

- A potential application for GTM is visualization

- By calculating W^* and β^*, GTM defines the probability distribution in the data space conditioned on the latent variable, $p(t|x_k)$, $k = 1,\ldots,K$.

- Bayes Theorem can be used to calculate the corresponding posterior distribution in latent space for any point in data space, $p(x_k|t)$.

Experimental Results

- We can plot $p(x_k|t)$ against x_k;

- Alternatively, for each data point t_n, we can plot the entire data set by calculating:
  - the posterior mode projection of the distribution:

$$x_n^{mode} = argmax_{x_k} p(x_k|t_n)$$

OR

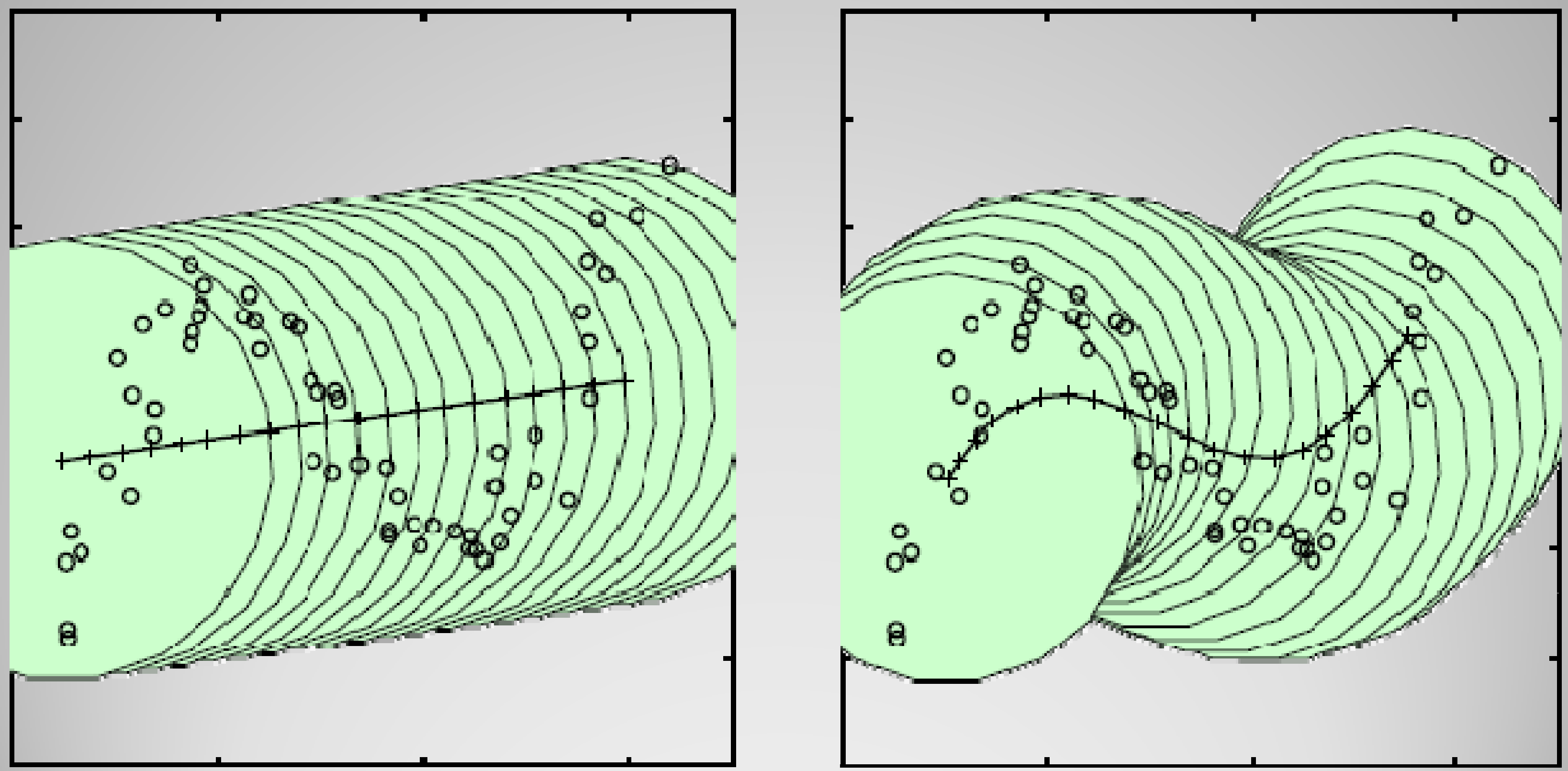
  - the posterior mean projection of the distribution:

$$x_n^{mean} = \sum_{k}^{K} x_k p(x_k|t_n)$$

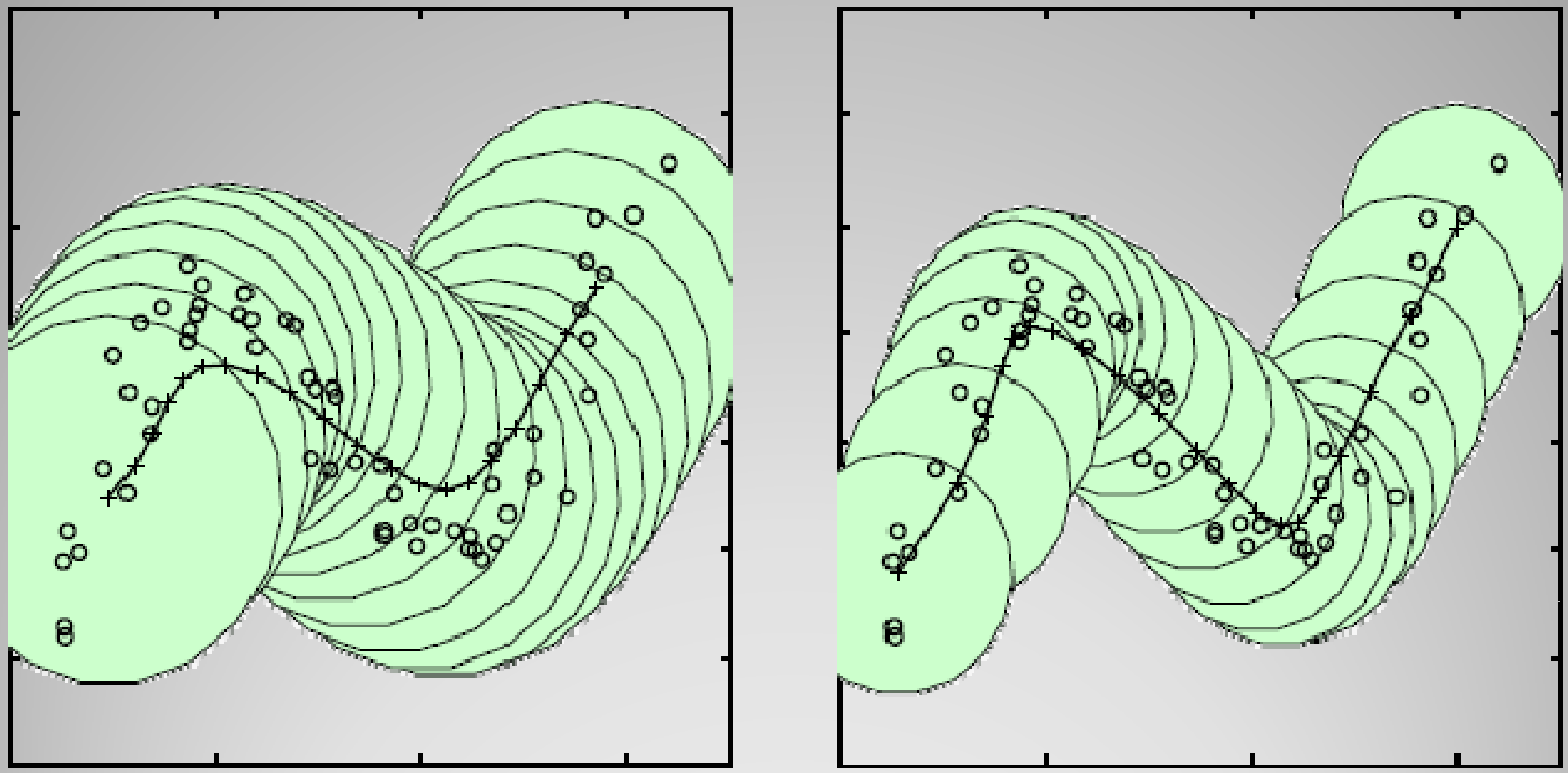
Experimental Results

The following illustrates the GTM learning process:

- GTM 1-D latent variable learns to model a 2-D curved line.
- The plots show the density model in data space.

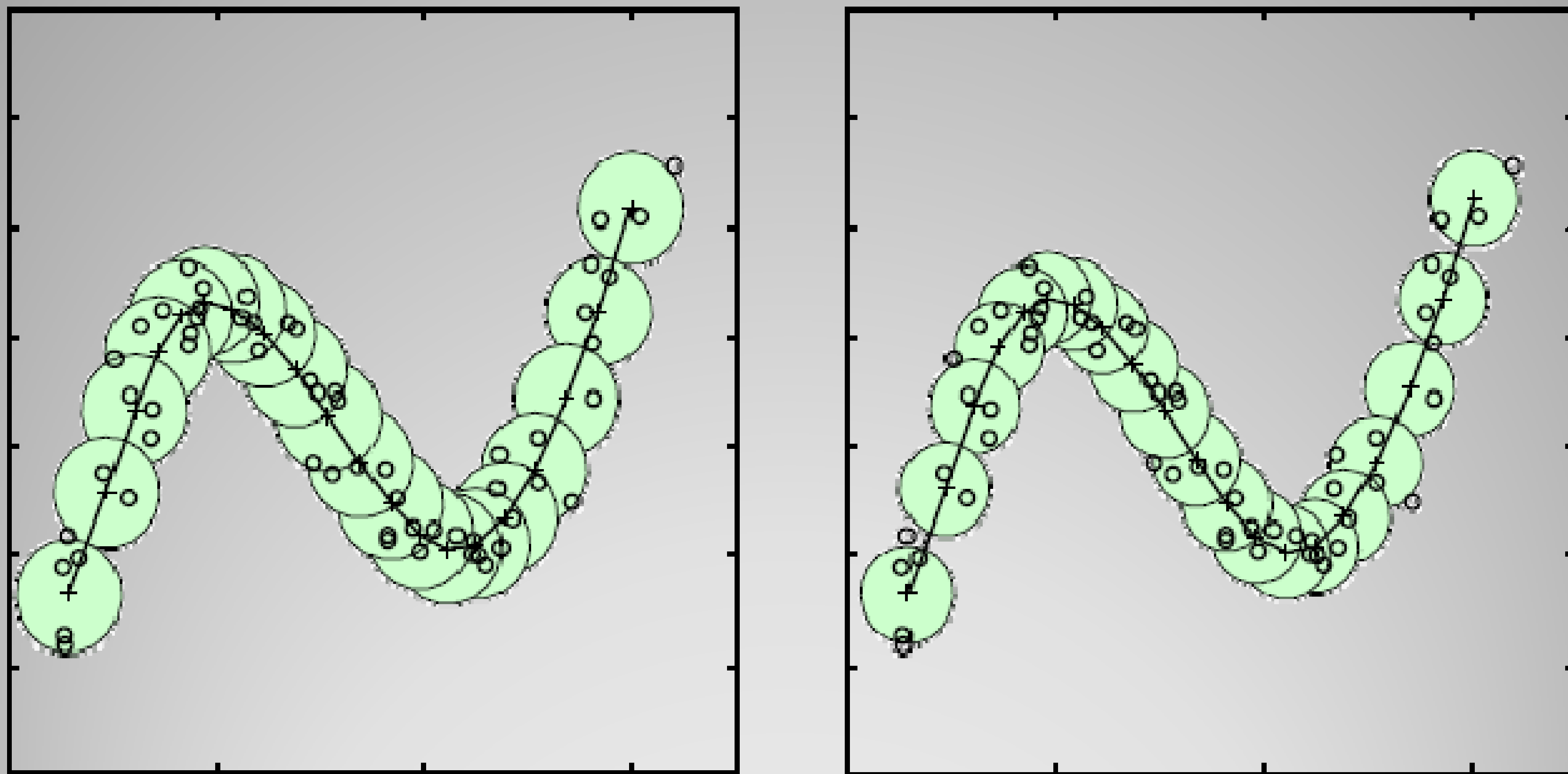


Experimental Results



Density model in data space after 2nd and 4th iteration.

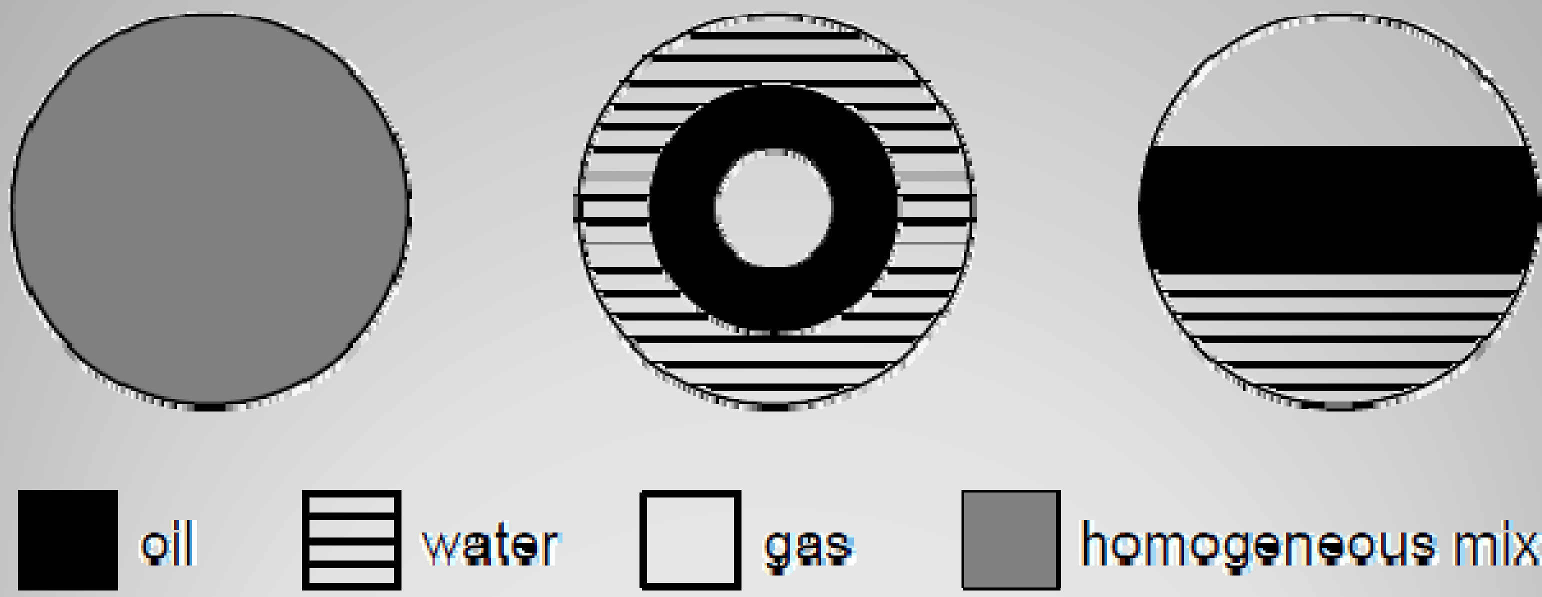
Experimental Results



Density model in data space after 8^{th} and 15^{th} iteration

Experimental Results

- The following illustrates GTM on 3-phase pipe flow data.
- GTM uses synthetically generated data simulating flow in a pipeline transporting a mixture of gas, oil and water.

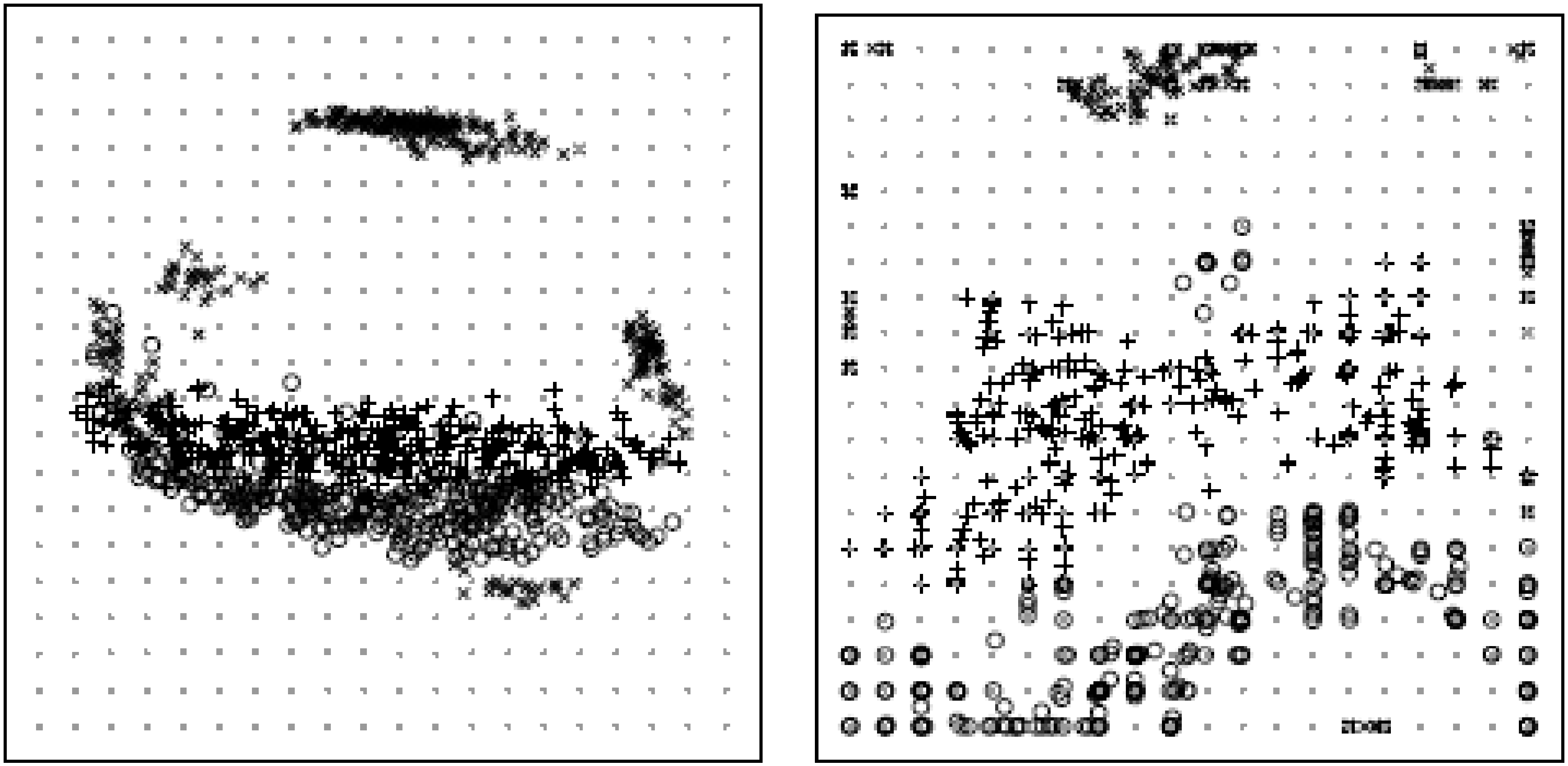


A cross sectional view of three different configurations

Left to right: Homogeneous, Annular and Stratified

Experimental Results

Experimental Results



Left: posterior-mean projection of the data in latent space of the PCA initialized GTM before training.

Right: Corresponding plot after training.

Introduction
GTM Model
EM Algorithm for GTM
Summary of learning algorithm
Experimental Results
Relationship to other Models
Discussion
Related work

SOM	GTM
Doesn’t optimize an objective function. Such a function doesn’t exist	It optimizes an objective function (log-likelihood function).
There is no general guarantee that the algorithm will converge	The EM-algorithm is guaranteed to converge to a maxima of the likelihood function.
There is no theoretical framework based on which appropriate values for the model parameters can be chosen	Uses Bayesian statistical theory to derive methods for treating model parameters
There is no logical means for comparing one SOM model to another or to different architectures.	The likelihood is a measure that can serve as a bases for comparing GTM model to other generative models
The mapping from topographic space to data space in the original SOM is only defined at locations of the nodes.	GTM defines a continuous manifold in the data space.

Relationship to other Models

- *Elastic Net Algorithm*:
  - A Gaussian mixture that encourages centers to follow a locally 1-D, globally cyclic structure.
  - Does not define a continuous data space manifold
- *Principal Curves:*
  - Similar to SOM. Projects each data point to a single point on the curve.
  - Uses Gaussian mixture equal to number of data points and a well defined likelihood function trained by the EM algorithm.

Relationship to other Models

Introduction
GTM Model
EM Algorithm for GTM
Summary of learning algorithm
Experimental Results
Relationship to other Models
Discussion
Related work

Benefits to the probabilistic model used in GTM:

- Handling missing data values by simple modification of the EM algorithm
- GTM can be used for visualization of data from the modeled distribution
- GTM models can be combined:
  - $P(t) = \Sigma_r P(r)p(t|r)$
    - $P(t|r)$ represents the r^{th} model.
- GTM can be generalized, extended and adapted within the framework of probability theory

Discussion

Introduction
GTM Model
EM Algorithm for GTM
Summary of learning algorithm
Experimental Results
Relationship to other Models
Discussion
Related work

- Ata Kaban, *A Scalable Generative Topographic Mapping for Sparse Data Sequences*, Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'05) - Volume I, p.51-56, April 04-06, 2005.
- Grimmelstein, M. and Urfer, W.W. (2005), *Analyzing protein data with the generative topographic mapping approach*.

Related work