Generative Topographic Mapping

Nonlinear Dimensionality Reduction Seminar
Helsinki University of Technology

László Kozma <lkozma@cc.hut.fi>
• Nonlinear Dimensionality Reduction
  – Distance-preserving methods
  – Topology-preserving methods
    • Predefined-lattice
      – SOM
      – *Generative Topographic Mapping*
    • Data-driven lattice
      – ...
      – ...
      – ...
Generative Topographic Mapping

- Generative model
- Probabilistic method based on Bayesian learning
- Introduced by Bishop, Svensén, et. al. in 1996
- http://www.ncrg.aston.ac.uk/GTM/
GTM in a nutshell

- $R^D$ – data space
- $R^L$ – latent space
- $D > L$

1. probabilistically pick a point in $R^L$
2. map the point to $R^D$ via a nonlinear, smooth function
3. add noise

- probability distribution in $R^L$, smooth function, noise can all be learned through EM-algorithm
GTM is “a principled SOM”

- explicit density model over data
- objective function that quantifies how well the map is trained
- sound, provably convergent optimization method (EM-algorithm)
Generative Topographic Mapping

- Data space: $\mathbb{R}^D$
- Latent space: $\mathbb{R}^L$

- Find a nonlinear, smooth function:
  \[ y(x, W): \mathbb{R}^L \rightarrow \mathbb{R}^D \]
  (for example a MLP, where W-weights)

- $y$ maps an $L$ dimensional space into an $L$-dimensional manifold non-linearly embedded in $D$-dimensions
• \( p(x) \) – probability distribution in latent space
• induces probability distribution in data space
• Convolve distribution with Gaussian noise:

\[
p(t|x, W, \beta) = \mathcal{N}(y(x, W), \beta) \\
= \left(\frac{\beta}{2\pi}\right)^{-D/2} \exp\left\{-\frac{\beta}{2} \sum_\text{d} (t_d - y_d(x, W))^2\right\}
\]

- \( \beta \) – inverse of variance
- \( D \) – dimension of data space
Integrate out the latent variables:

\[ p(t|W, \beta) = \int p(t|x, W, \beta) p(x) \, dx. \]

generally not solvable analytically

choose grid points in latent space:

\[ p(x) = \frac{1}{K} \sum_{k}^{K} \delta(x - x_k), \]

\[ p(t|W, \beta) = \frac{1}{K} \sum_{k}^{K} p(t|x_k, W, \beta). \]
• Likelihood of the model
\[
\mathcal{L} = \prod_{n}^{N} p(t|W, \beta) = \prod_{n}^{N} \left[ \frac{1}{K} \sum_{k}^{K} p(t_n|x_k, W, \beta) \right]
\]

• Log-likelihood:
\[
\ell = \sum_{n}^{N} \ln \left( \frac{1}{K} \sum_{k}^{K} p(t_n|x_k, W, \beta) \right)
\]

• Maximize it with respect to $\beta$ and $W$
• For example with gradient descent
• Mixture of Gaussians: use EM-algorithm
EM - algorithm

- **E-step:**
  - responsibility of latent point $x_k$ for data point $t_n$
  
  $$r_{kn} = p(x_k|t_n, W, \beta) = \frac{p(t_n|x_k, W, \beta)p(x_k)}{\sum_{k'} p(t_n|x_{k'}, W, \beta)p(x_{k'})}$$
  - $p(x_k)$ constant (1/K)

- **M-step:**
  - $r_{kn}$ used as weights to update $\beta$ and $W$
  - “move each component of the mixture towards data points for which it is most responsible”
The nonlinear function $y$

- choice important if we want to preserve topology
- linear combination of linear and non-linear basis functions

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

- $L$ linear basis functions can be initialized using PCA
- non-linear basis functions typically Gaussian kernels
- nr. of basis functions $\sim$ nr. of grid points
Initialization

- Latent space dimension (1 or 2)
- Prior distribution in latent space (grid points)
- Center and width of Gaussian basis functions
- Weights $W$:
  - can be chosen randomly, such that variance over $y$ equals variance of test data
  - if $y$ has linear components, they can be initialized with PCA
  - non-linear component-weights can be set to zero or to small random values
- Noise variance: $1/\beta$ (at least the length of $(L+1)$th PC)
Algorithm

Pick latent space dimension, grid points
Choose basis functions
Initialize $W, \beta$
repeat
  E-step
  M-step
until Convergence
Example...

iteration 0,1
Example...

iteration 2,4
Example...

iteration 8,15
Dimension Reduction

• Suppose we found suitable $W^*$ and $\beta^*$

• We have a probability distribution in data space: $p(t|x_k)$ $k=1,2,3,...,K$

• Prior distribution in latent space: $p(x_k)=1/K$

• Use Bayes-theorem:

$$p(x_k|t) = \frac{p(t|x_k, W^*, \beta^*)p(x_k)}{\sum_{k'} p(t_n|x_{k'}, W^*, \beta^*)p(x_{k'})}$$
Dimension Reduction

- posterior-mode projection:

\[ x_n^{\text{mode}} = \arg\max_{x_k} p(x_k | t_n) \]

- posterior-mean projection:

\[ x_n^{\text{mean}} = \sum_{k}^{K} x_k p(x_k | t_n) \]
Results
GTM summary

• Advantages:
  – In addition to finding $\hat{x}$ for given y, it can also approximate $\hat{p}(x|y)$
  – Easy to generalize to new points
  – Optimizes well-defined function (log-likelihood)
  – EM maximizes log-likelihood monotonically, converges after few iterations

• Disadvantages:
  – Inefficient for more than 2 latent dimensions
  – Doesn't estimate intrinsic dimension
  – Limited mapping power: kernel centers, variances fixed, only weights adjusted