GP-select: Accelerating EM using adaptive subspace preselection

Jacquelyn A. Shelton[1], Jan Gasthaus[2,3], Zhenwen Dai[4], Jörg Lücke[5], and Arthur Gretton[2]
[1]Technical University Berlin, [2]UCL Gatsby and CSML, [3]Amazon, [4]University of Sheffield, [5]University of Oldenburg

Highlights/Abstract

- Novel nonparametric procedure for fast inference in generative graphical models with large # of latent states, e.g. $\#\textit{latent states}^{\#\textit{latent variables}}$
- Idea: meta-algorithm for EM, iterative latent variable preselection – alternate between learning a 'selection function' (reveal the relevant latent variables) and using the result for a compact approx of the posterior distribution for EM
- How: learn selection function entirely from the observed data and current EM state via Gaussian process regression – earlier approaches used expensive manually-designed selection functions for each problem setting – our approach is fully automatic and flexible
- Experiments suggest GP-select to play a crucial role for inference in complex hierarchical models (e.g. [1]) where the relationship between inputs / outputs is complex and thus hand-derived selection functions are expensive (or impossible)

Variable selection for accelerated inference

Notation:

- Observed data: $\mathbf{y}^{(n)} = (y_1^{(n)}, \ldots, y_D^{(n)})^{\mathrm{T}}$, N observations of D dimensions
- Binary latent variables: $\mathbf{s}^{(n)} = (s_1^{(n)} \ldots, s_H^{(n)})^{\mathrm{T}} \in \{0,1\}^H$, H latent dims
- Reduced latent space: H'-dimensional, where $H' \ll H$ dimensions.
- Prior distribution over latent variables is $p(\mathbf{s}|\theta)$, likelihood of the data is $p(\mathbf{y}|\mathbf{s},\theta) \rightarrow$ Posterior distribution over latent variables:

$$p(\mathbf{s}^{(n)}|\mathbf{y}^{(n)},\Theta) = \frac{p(\mathbf{s}|\Theta)\, p(\mathbf{y}|\mathbf{s},\Theta)}{\sum_{\mathbf{s}'^{(n)}} p(\mathbf{s}'|\Theta)\, p(\mathbf{y}|\mathbf{s}',\Theta)} \tag{1}$$

Selection via Expectation Truncation (ET) [2] in EM

- Posterior distribution (1) approximated by a truncated posterior distribution, computed with support reduced to $\mathcal{K}_n$:

$$p(\mathbf{s}^{(n)}|\mathbf{y}^{(n)},\Theta) \approx q_n(\mathbf{s}^{(n)};\Theta) = \frac{p(\mathbf{s}^{(n)},\mathbf{y}^{(n)}|\,\Theta)\,\delta(\mathbf{s}^{(n)} \in \mathcal{K}_n)}{\sum_{\mathbf{s}'^{(n)} \in \mathcal{K}_n} p(\mathbf{s}'^{(n)},\mathbf{y}^{(n)}|\,\Theta)} \tag{2}$$

  - where $\mathcal{K}_n$ contains the latent states of the H' relevant variables for data point $\mathbf{y}^{(n)}$, and $\delta(\mathbf{s} \in \mathcal{K}_n) = 1$ if $\mathbf{s} \in \mathcal{K}_n$, else 0,
  - $\mathcal{K}_n$ should contain most of the probability mass $p(\mathbf{s}\,|\,\mathbf{y})$, and
  - $\mathcal{K}_n$ should be significantly smaller than full latent space

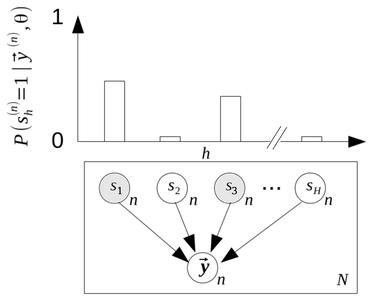
ET with affinity

- Constructing a selection function first, rank the latent variables according to an *affinity function* $f_h(\mathbf{y}^{(n)}) : \mathrm{R}^D \mapsto \mathrm{R}$ which directly reflects the relevance of latent variable s_h.
- A natural choice of selection function is the one that *approximates the marginal posterior probability* of each variable, e.g. learn f as follows:

$$f_h(\mathbf{y}^{(n)}) \approx p_h^{(n)} \equiv p(s_h^{(n)} = 1|\mathbf{y}^{(n)},\Theta) \tag{3}$$

$\rightarrow$ Use the affinity function to select relevant variables: marginal posterior probability p_h exceeds a threshold

Latent Variable Preselection: affinity and GP-Select



Affinity (approx marg post prob) to highlight most relevant latent variables

- Sort and reduce full indices to H' most rel. variables' set – define $\gamma\,(\hat{\mathbf{p}}^{(n)})$ to output the H' selected variable indices I for the nth data point
- Define subset of the H'-dimensional relevant latent states $\mathcal{K}_n$ with $\mathcal{I}(I)$
- All non-relevant variable states s_h for all variables $h \notin I$ are set to 0 in Eq. (2)
- Using $\mathbf{f}$, $\mathcal{I}$, and γ, we can define a *selection function* $\mathcal{S} : \mathrm{R}^D \mapsto 2^{\{1,\ldots,H\}}$ to select subsets $\mathcal{K}_n$ per data point $\mathbf{y}^{(n)}$ for the affinity based selection function:

$$\mathcal{S}(\mathbf{y}^{(n)}) \;=\; \mathcal{I}\left[\gamma\left[\mathbf{f}(\mathbf{y}^{(n)})\right]\right] \;=\; \mathcal{K}_n \tag{4}$$

GP-select: Learn affinity with GP regression

- Previous work: selection function $\mathcal{S}$ was deterministic and derived by hand for each model using upper bounds or noiseless limits [3,4]
- We generalize and automatize this approach: learn Ss with GP regression
- Define $f_h(\mathbf{y}^{(n)}) \sim \mathrm{GP}\,(0,\, k(\cdot,\cdot))$, where $k(\cdot,\cdot)$ is the covariance kernel and flexibly parameterizable to represent the relationship between variables
- Before each E-step: train GP on p_h from prev. EM iteration (where $p_h = \langle s_h \rangle$): $\mathcal{D} = \{(\mathbf{y}^{(n)}, \langle \mathbf{s} \rangle^{(n)}_{q_n(\mathbf{s})}) | n = 1, \ldots, N\}$
- Compute predicted mean of GP using leave-one-out (LOO) prediction:

$$\hat{p}_h^{(n)} \leftarrow \langle s \rangle_h^{(n)} - \frac{[K^{-1}\langle \mathbf{s} \rangle h]_{nn}}{[K^{-1}]_{nn}} \tag{5}$$

$\rightarrow$ Efficiently implementable for all latent vars $h = 1, ..., H$ and data points $n = 1, ..., N$ using matrix operations – only 1 kernel matrix inversion for all N

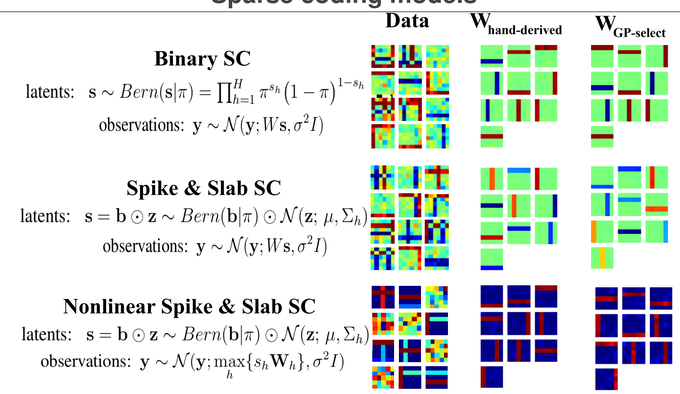
- Substitute Eq. (5) for $\mathbf{f}$ in the affinity based selection function Eq. (4)

Algorithm

for EM iterations $t = 1, \ldots, T$ **do**
  for data point $n = 1, \ldots, N$ **do**
    compute affinity of all latent variables $\hat{\mathbf{p}}_t^{(n)}$: (5)
    compute subset of relevant states $\mathcal{S}$: (4)
    compute truncated posterior $q_{n,t}(\mathbf{s})$, E-step: (2)
    update model parameters in M-step
    store $\langle \mathbf{s} \rangle^{(n)}_{q_t(\mathbf{s})}$ for $\mathbf{p}^{(n)}$ in EM iteration $t+1$
  end for
  optimize kernel hyperparams every T^* EM iterations
end for

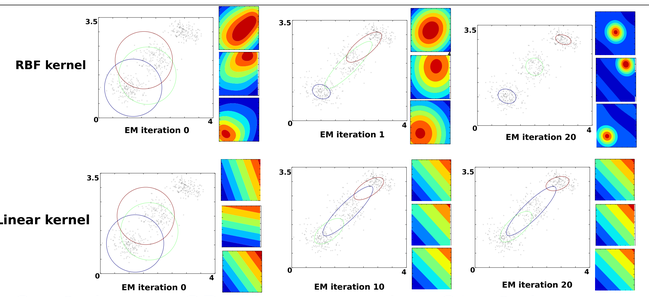
Experiments

Sparse coding models



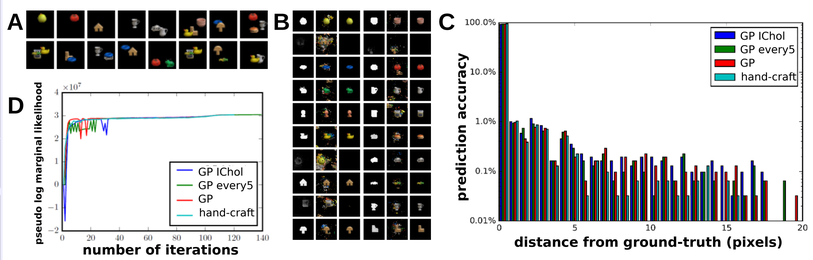
- Data: $N = 2,000$ with $D = 5 \times 5$ obs dims & $H = 10$ latent dims/bars gen. by each model, with GP-select to preselect $H' = 5$ dims
- Shown: final EM it; GP-select converges to GT params, $W_{GP-select}$

Gaussian mixture model



- Data: $C = 3$ clusters, GP-select to preselect $C' = 2$ clusters
- Shown: using the wrong selection function can do harm (i.e. miss patterns); sel. funcs need to be flexible and possibly nonlinear

Translation invariant occlusive models [1]



- Problem: locate objects in scene (**A**), with massive latent space complexity – # of obj. locations exponentiated by # of objects.
- Speed: partial incomplete Cholesky approx to for faster GP regression computation, update GP hyperparams every 5 EM its
- Shown: all 3 variants of GP-selection learn all objects (**B**) with accuracy equivalent to hand-crafted selection (**C** & **D**)

References

[1] Dai, Z. and Lücke, J. (2014). Autonomous document cleaning – a generative approach to reconstruct strongly corrupted scanned texts. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 36(10):1950–1962.
[2] J. Lücke and J. Eggert. (2010). Expectation Truncation And the Benefits of Preselection in Training Generative Models. Journal of Machine Learning Research (JMLR).
[3] Bornschein, J., Henniges, M., and Lücke, J. (2013). Are V1 simple cells opti- mized for visual occlusions? A comparative study. PLoS Computational Biology, 9(6):e1003062.
[4] Sheikh, A.-S., Shelton, J., and Lücke, J. (2014). A truncated EM approach for spike- and-slab sparse coding. Journal of Machine Learning Research (JMLR), 15:2653–2687.