Dense Message Passing for Sparse Principal Component Analysis

Kevin Sharp Magnus Rattray

University of Manchester

AIStats, Chia Laguna Resort, Sardinia, 14th May 2010

Outline

1. Introduction
   - Motivating Application: Gene Regulation
2. Dense Message Passing
   - Model Description
   - Algorithm Description
   - Statistical Mechanics Theory
3. Results
   - Simulated Data
   - Gene Expression Data
   - Marginal Likelihood Estimation
4. Summary

Motivation

- Gene regulation - inference of explanatory factors.
- Microarray data - ‘Large p small N’ regime.
- Explanatory factors have truly sparse loadings.
- Zero-norm priors allocate probability mass to truly sparse solutions.
- Easy to encode prior knowledge of sparse structure.
- But, zero-norm priors are problematic for inference.

Model - Probabilistic PCA

- For the n^{th} data point, $\boldsymbol{y}_n$, we assume:

$$\boldsymbol{y}_n = \boldsymbol{w} x_n + \epsilon_n \ ,$$

where $x_n \sim \mathcal{N}(0, 1)$.

- To simplify the description, $\epsilon_n \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$.
- Integrate out x:

$$P\left(\boldsymbol{y}_n \,|\, \boldsymbol{w}\right) = \mathcal{N}\left(\boldsymbol{y}_n \,\middle|\, \mathbf{0}, \mathbf{I} + \boldsymbol{w}\boldsymbol{w}^{\mathsf{T}}\right)$$

Model - Probabilistic PCA

- For the n^{th} data point, $\boldsymbol{y}_n$, we assume:

$$\boldsymbol{y}_n = \boldsymbol{w} x_n + \epsilon_n \ ,$$

where $x_n \sim \mathcal{N}(0, 1)$.

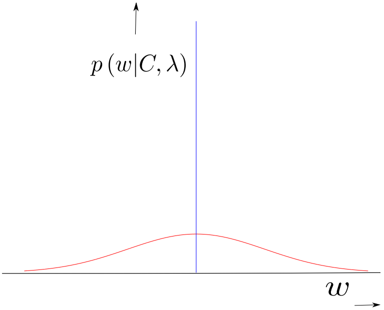
- To simplify the description, $\epsilon_n \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$.
- Integrate out x:

$$P\left(\boldsymbol{y}_n \,|\, \boldsymbol{w}\right) = \mathcal{N}\left(\boldsymbol{y}_n \,\middle|\, \mathbf{0}, \mathbf{I} + \boldsymbol{w}\boldsymbol{w}^{\mathsf{T}}\right)$$

Sparsity Mixture Prior

We use a **spike and slab** mixture prior:

$$P(\boldsymbol{w}|C,\lambda) = \prod_{j=1}^{p} \left[(1-C)\delta(w_j) + C\mathcal{N}(w_j|0,(\lambda)^{-1})\right]$$



C - Fraction of non-zero ws
λ - inverse width

Sparsity Mixture Prior

Express in factorised form using binary variables $z_j \in \{0, 1\}$:

$$P(\boldsymbol{v}, \boldsymbol{z}) = \prod_{j=1}^{p} \{(1 - C)\mathcal{N}(v_j \,|\, 0, 1)\}^{1-z_j} \left\{C\mathcal{N}\left(v_j \,|\, 0, \lambda^{-1}\right)\right\}^{z_j}$$

where $w_j = z_j v_j$ and $v_j \in \mathcal{R}$

Form of the Prior in High Dimensions

- $z_j \sim \text{Bernoulli}\,(C)$
- $\sum_j z_j \sim \text{Bin}(p, C)$
- For large dimension, p, the fraction of non-zero parameters is highly peaked at C.

Form of the Prior in High Dimensions

- $P(w_j | z_j = 1) = \mathcal{N}(w_j | 0, \lambda^{-1})$
- For large dimension, p:

$$\|\boldsymbol{w}\|^2 \sim \mathcal{N}(0, Cp/\lambda)$$

- For large dimension, p, this distribution is **approximately spherical** with radius $\sqrt{Cp/\lambda}$.

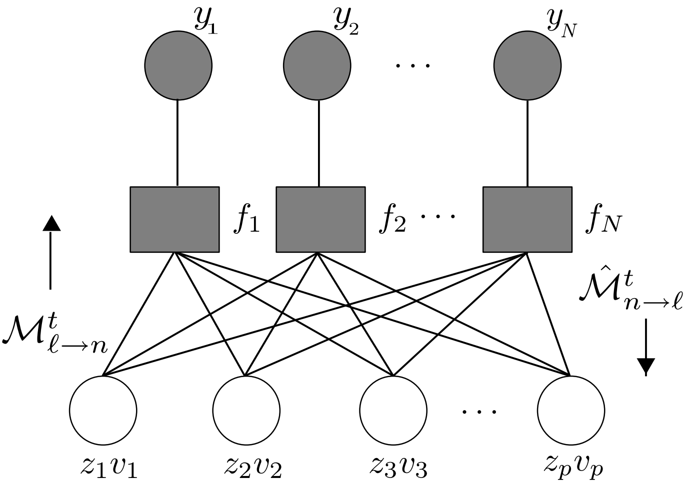
- Conclusion - A **constraint-based** prior:

$$p(\boldsymbol{w}, \boldsymbol{z}|C, \lambda) \propto \delta\left(\sum_{j=1}^{p} z_j - pC\right)\delta\left(\sum_{j=1}^{p} w_j^2 - \frac{pC}{\lambda}\right)$$

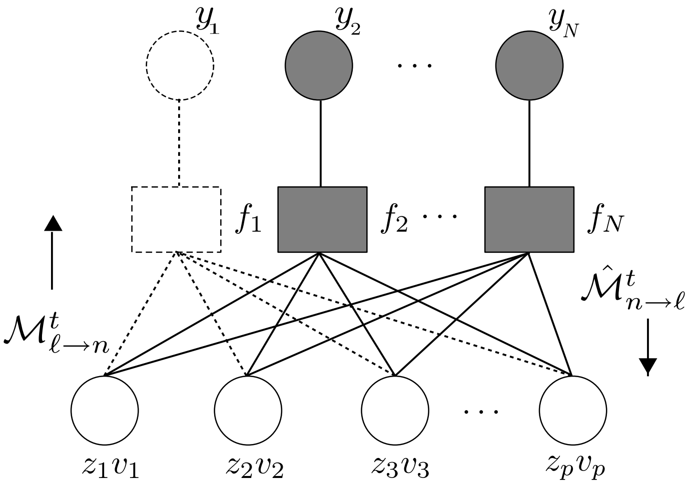
  is almost equivalent to the mixture prior in high dimensions.

- This proves useful for developing the message passing algorithm.

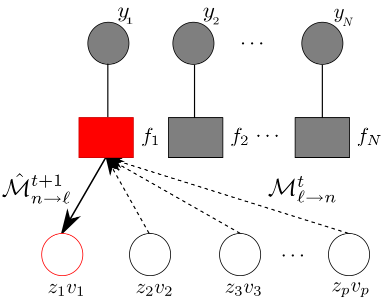
Factor Graph Representation



Belief Propagation

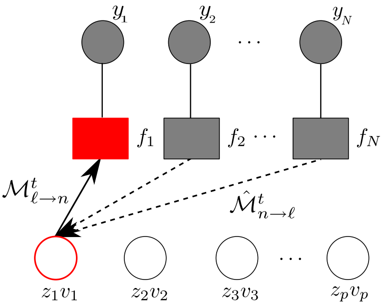


Factor to Variable Messages



$$\hat{\mathcal{M}}^{t+1}_{n\to\ell}(v_\ell, z_\ell) \propto \int \prod_{j\neq\ell} dv_j \sum_{\mathbf{z}\setminus z_\ell} f_n(\mathbf{y}_n, \mathbf{z}, \mathbf{v}) \prod_{j\neq\ell} \mathcal{M}^{t}_{j\to n}(v_j, z_j)$$

Variable to Factor Messages



$$\mathcal{M}^t_{\ell\to n}(v_\ell, z_\ell) \propto P(v_\ell, z_\ell) \prod_{m\neq n} \hat{\mathcal{M}}^t_{m\to\ell}(v_\ell, z_\ell)$$

Marginal Beliefs

After t iterations, the approximate posterior marginal belief is:

$$p^t\left(z_\ell, v_\ell \mid \boldsymbol{Y}\right) = \frac{p\left(z_\ell, v_\ell\right) \prod_{m=1}^{N} \hat{\mathcal{M}}^t_{m\to\ell}\left(v_\ell, z_\ell\right)}{\int \mathrm{d}v_\ell \sum_{z_\ell} p\left(z_\ell, v_\ell\right) \prod_{m=1}^{N} \hat{\mathcal{M}}^t_{m\to\ell}\left(v_\ell, z_\ell\right)}$$

where $p\left(z_\ell, v_\ell\right)$ is the prior.

Two Problems

Unfortunately,

1. $$\hat{\mathcal{M}}^{t+1}_{n\to\ell}\left(v_\ell, z_\ell\right) \propto \int \prod_{j\neq\ell} dv_j \sum_{\boldsymbol{z}\setminus z_\ell} f_n\left(\boldsymbol{y}_n, \boldsymbol{z}, \boldsymbol{v}\right) \prod_{j\neq\ell} \mathcal{M}^{t}_{j\to n}\left(v_j, z_j\right)$$

   is hard to compute.

2. Belief propagation is not expected to converge for dense graphical models.

Solutions

1. Exploit the high-dimensionality:

   Use a **Gaussian approximation**.

2. Impose consistency requirements:

   Use the constraint-based prior to **enforce sparsity and length constraints** self-consistently at each iteration.

Uda and Kabashima - Statistical Mechanical Development of a Sparse Bayesian Classifier, *J. Phys. Soc. Japan*, 2005

Gaussian Approximation (1)

Notice that likelihood factors may be written as:

$$f_n(\boldsymbol{y}_n, \boldsymbol{z}, \boldsymbol{v}) = \frac{1}{\sqrt{(2\pi)^p \left(1 + \|\boldsymbol{w}\|^2\right)}} \exp\left(-\frac{\boldsymbol{y}_n^{\mathrm{T}}\boldsymbol{y}_n}{2} + \Delta_n^2/2\right) ,$$

with Δ_n defined by: $\Delta_n = \frac{\sum_{j=1}^{p} y_j^n z_j v_j}{\sqrt{1+\|\boldsymbol{w}\|^2}}$

For large dimension, p, Central Limit Theorem permits a Gaussian approximation.

Gaussian Approximation (2)

For constant $\|\boldsymbol{w}\|^2$, we replace Δ_n by:

$$\frac{y_\ell^n z_\ell v_\ell}{\sqrt{1 + Cp/\lambda}} + \underbrace{\frac{1}{\sqrt{1 + Cp/\lambda}} \sum_{j \neq \ell} y_j^n m_{j \to n}^t}_{\langle \Delta_{n \setminus \ell} \rangle_{n \setminus \ell}^t} + \sqrt{V_{n \setminus \ell}^t}\, u$$

where $u \sim \mathcal{N}(0, 1)$.

$m_{j \to n}^t$ is the mean of $z_j v_j$ under the cavity distribution with the n^{th} data point removed.

Gaussian Approximation (3)

- The variance, $V^t_{n\backslash\ell}$ is given by:

$$\frac{1}{1+Cp/\lambda}\sum_{j,k\neq\ell} y_j^n y_k^n \left\langle \left(z_j v_j - m^t_{j\to n}\right)\left(z_k v_k - m^t_{k\to n}\right)\right\rangle^t_{n\backslash\ell}$$

- For large dimension, p, fluctuations about the sample mean are $\mathcal{O}\left(\frac{1}{\sqrt{p}}\right)$: $V^t_{n\backslash\ell}$ is *self-averaging*.

- $V^t \approx \frac{1}{(1+Cp/\lambda)}\left(Cp/\lambda - \sum_{j=1}^p \left(m^t_j\right)^2\right)$

Consistency - Constraint-based Prior

1. The spike and slab prior can be written:

$$P(\boldsymbol{v}, \boldsymbol{z}) \propto \prod_{j=1}^{p} \exp\left(-\frac{1}{2}\left(1 - z_j + Gz_j\right) v_j^2 + \gamma z_j\right)$$

   where $\gamma = \ln\left(\frac{C\sqrt{\lambda}}{1-C}\right)$ and $G = \lambda$.

2. Adjust G and γ at each iteration to satisfy the constraint-based prior *on average*:
   $\sum_{j=1}^{p} \langle z_j \rangle^t = Cp$ and $\sum_{j=1}^{p} \left\langle z_j v_j^2 \right\rangle^t = Cp/\lambda$

3. Note, after convergence, $G \neq \lambda$ and $\gamma \neq \ln\left(\frac{C\sqrt{\lambda}}{1-C}\right)$

4. Consistent with replica analysis.

Consistency - Constraint-based Prior

1. The spike and slab prior can be written:

$$P(\boldsymbol{v}, \boldsymbol{z}) \propto \prod_{j=1}^{p} \exp\left(-\frac{1}{2}\left(1 - z_j + Gz_j\right) v_j^2 + \gamma z_j\right)$$

   where $\gamma = \ln\left(\frac{C\sqrt{\lambda}}{1-C}\right)$ and $G = \lambda$.

2. Adjust G and γ at each iteration to satisfy the constraint-based prior *on average*:
   $\sum_{j=1}^{p} \langle z_j \rangle^t = Cp$ and $\sum_{j=1}^{p} \left\langle z_j v_j^2 \right\rangle^t = Cp/\lambda$

3. Note, after convergence, $G \neq \lambda$ and $\gamma \neq \ln\left(\frac{C\sqrt{\lambda}}{1-C}\right)$

4. Consistent with replica analysis.

Consistency - Constraint-based Prior

1. The spike and slab prior can be written:

$$P(\boldsymbol{v}, \boldsymbol{z}) \propto \prod_{j=1}^{p} \exp\left(-\frac{1}{2}\left(1 - z_j + G z_j\right) v_j^2 + \gamma z_j\right)$$

where $\gamma = \ln\left(\frac{C\sqrt{\lambda}}{1-C}\right)$ and $G = \lambda$.

2. Adjust G and γ at each iteration to satisfy the constraint-based prior *on average*:
$\sum_{j=1}^{p} \left\langle z_j \right\rangle^t = Cp$ and $\sum_{j=1}^{p} \left\langle z_j v_j^2 \right\rangle^t = Cp/\lambda$

3. Note, after convergence, $G \neq \lambda$ and $\gamma \neq \ln\left(\frac{C\sqrt{\lambda}}{1-C}\right)$

4. Consistent with replica analysis.

Consistency - Constraint-based Prior

1. The spike and slab prior can be written:

$$P(\boldsymbol{v}, \boldsymbol{z}) \propto \prod_{j=1}^{p} \exp\left(-\frac{1}{2}\left(1 - z_j + G z_j\right) v_j^2 + \gamma z_j\right)$$

   where $\gamma = \ln\left(\frac{C\sqrt{\lambda}}{1-C}\right)$ and $G = \lambda$.

2. Adjust G and γ at each iteration to satisfy the constraint-based prior *on average*:
   $\sum_{j=1}^{p} \langle z_j \rangle^t = Cp$ and $\sum_{j=1}^{p} \left\langle z_j v_j^2 \right\rangle^t = Cp/\lambda$

3. Note, after convergence, $G \neq \lambda$ and $\gamma \neq \ln\left(\frac{C\sqrt{\lambda}}{1-C}\right)$
4. Consistent with replica analysis.

Replica Analysis (1)

- Compute average of the log marginal likelihood over all possible datasets for $p \to \infty$
- $\alpha = N/p$ is held constant (where N is the sample size).
- Works well for $\alpha \ll 1$ – 'large p small N'
- Not mathematically rigorous, but a useful tool.

Replica Analysis (2)

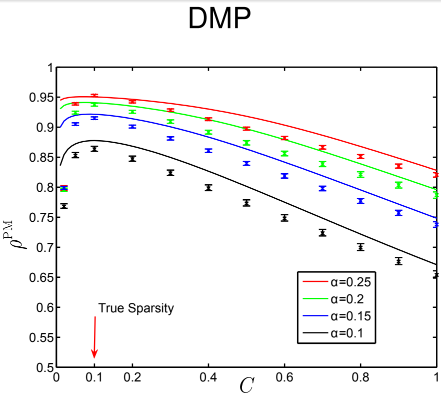
- Derive expressions involving the posterior mean (PM) parameter vector, $\boldsymbol{w}^{\mathrm{PM}}$:
  - squared length, $||\boldsymbol{w}^{\mathrm{PM}}||^2$
  - overlap with the true parameter vector, $\boldsymbol{w}^{\mathrm{PM}} \cdot \boldsymbol{w}^t$.

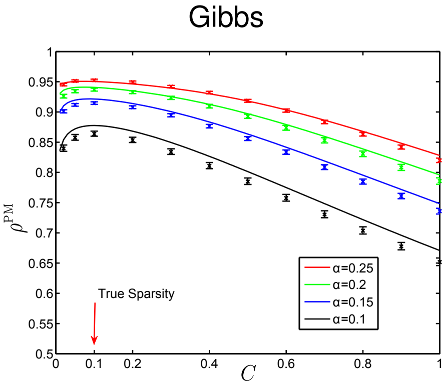
$$\boldsymbol{w}^t \sim \prod_{j=1}^{p} \left[(1 - C_t)\delta(w_j) + C_t \mathcal{N}(w_j | 0, (\lambda_t)^{-1}) \right]$$

- Can show that the algorithm is consistent with this analysis.
- Can compare algorithm performance to theory using

$$\rho^{\mathrm{PM}} = \frac{\boldsymbol{w}^{\mathrm{PM}} \cdot \boldsymbol{w}^t}{||\boldsymbol{w}^{\mathrm{PM}}|| \, ||\boldsymbol{w}^t||} \, .$$

Simulated Data - DMP vs Theory





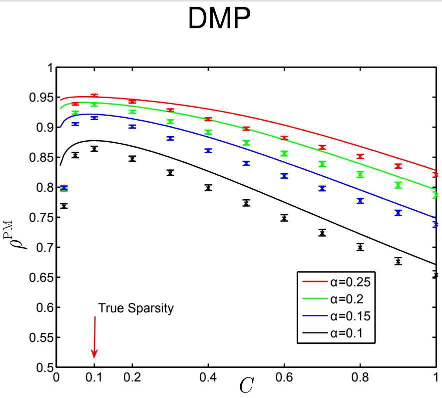
C - fraction of non-zero parameters;

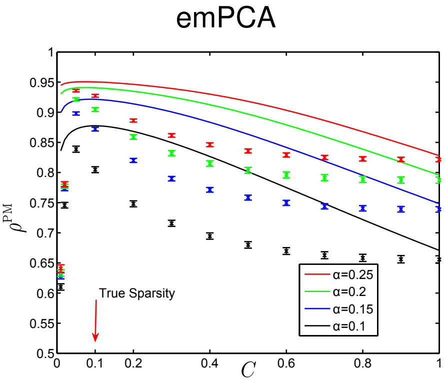
ρ^{PM} cosine angle between $\boldsymbol{w}^{\mathrm{PM}}$ and $\boldsymbol{w}^t$.

$N = 200$ samples, $\alpha = N/p$;

Results averaged over 50 sample datasets.

Simulated Data - DMP vs emPCA





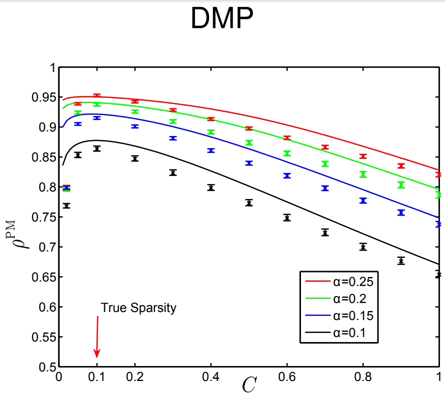
C - fraction of non-zero parameters;

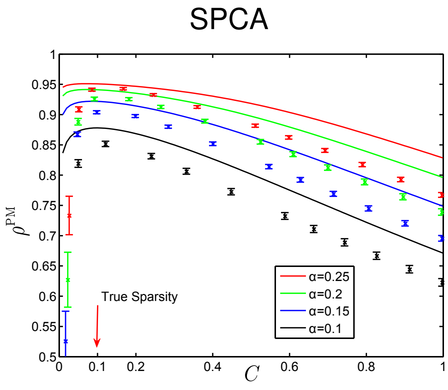
$N = 200$ samples, $\alpha = N/p$;

ρ^{PM} cosine angle between $\boldsymbol{w}^{PM}$ and $\boldsymbol{w}^{t}$.

Results averaged over 50 sample datasets.

Simulated data - DMP vs SPCA





C - fraction of non-zero parameters;

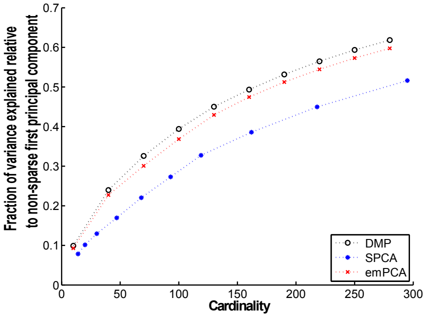
ρ^{PM} cosine angle between $\boldsymbol{w}^{\mathrm{PM}}$ and $\boldsymbol{w}^t$.

$N = 200$ samples, $\alpha = N/p$;

Results averaged over 50 sample datasets.

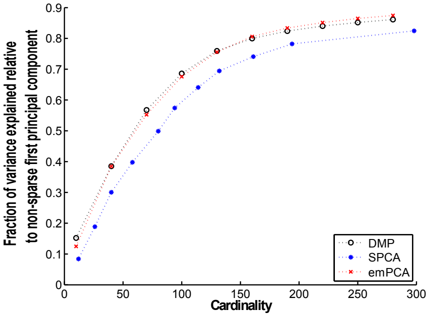
Gene Expression Data - DMP vs emPCA and SPCA

Armstrong *et al.*



$p = 12582$, $N = 72$

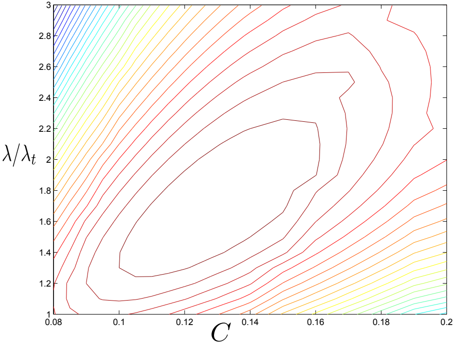
Ramaswamy *et al.*



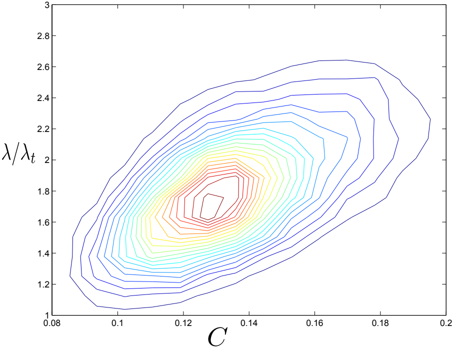
$p = 16063$, $N = 144$

Marginal Likelihood Estimation - Simulated Data

DMP



Gibbs



C - fraction of non-zero parameters;

$N = 200$ samples; dimension, $p = 2000$.

λ - assumed signal precision;

λ^t - true signal precision.

True sparsity - 0.1.

Summary

- Novel message passing algorithm for Sparse Bayesian PCA in high dimensions
- Message updates rendered tractable using a Gaussian approximation
- Convergence achieved by imposing consistency requirements derived from statistical mechanics analysis.
- Inference of posterior marginals exhibits near optimal performance compared to theory.
- Outperforms two other recently published algorithms.
- Approximation to Marginal Likelihood also available.

Summary

- Novel message passing algorithm for Sparse Bayesian PCA in high dimensions
- Message updates rendered tractable using a Gaussian approximation
- Convergence achieved by imposing consistency requirements derived from statistical mechanics analysis.
- Inference of posterior marginals exhibits near optimal performance compared to theory.
- Outperforms two other recently published algorithms.
- Approximation to Marginal Likelihood also available.

Summary

- Novel message passing algorithm for Sparse Bayesian PCA in high dimensions
- Message updates rendered tractable using a Gaussian approximation
- Convergence achieved by imposing consistency requirements derived from statistical mechanics analysis.
- Inference of posterior marginals exhibits near optimal performance compared to theory.
- Outperforms two other recently published algorithms.
- Approximation to Marginal Likelihood also available.

Summary

- Novel message passing algorithm for Sparse Bayesian PCA in high dimensions
- Message updates rendered tractable using a Gaussian approximation
- Convergence achieved by imposing consistency requirements derived from statistical mechanics analysis.
- Inference of posterior marginals exhibits near optimal performance compared to theory.
- Outperforms two other recently published algorithms.
- Approximation to Marginal Likelihood also available.

Summary

- Novel message passing algorithm for Sparse Bayesian PCA in high dimensions
- Message updates rendered tractable using a Gaussian approximation
- Convergence achieved by imposing consistency requirements derived from statistical mechanics analysis.
- Inference of posterior marginals exhibits near optimal performance compared to theory.
- Outperforms two other recently published algorithms.
- Approximation to Marginal Likelihood also available.

Summary

- Novel message passing algorithm for Sparse Bayesian PCA in high dimensions
- Message updates rendered tractable using a Gaussian approximation
- Convergence achieved by imposing consistency requirements derived from statistical mechanics analysis.
- Inference of posterior marginals exhibits near optimal performance compared to theory.
- Outperforms two other recently published algorithms.
- Approximation to Marginal Likelihood also available.

The Future

- Hyperparameter estimation using Marginal likelihood.
- Extension to multiple factors:
  - Relatively straightforward for orthogonal factors.

    (but will require efficient hyperparameter estimation).
  - For non-orthogonal factors the best approach is a subject of on-going research.

The Future

- Hyperparameter estimation using Marginal likelihood.
- Extension to multiple factors:
  - Relatively straightforward for orthogonal factors.

    (but will require efficient hyperparameter estimation).
  - For non-orthogonal factors the best approach is a subject of on-going research.

Explore further

Matlab code available from: http://www.cs.man.ac.uk/~sharpk