# Stochastic expectation propagation

Yingzhen Li, Jose Miguel Hernandez-Lobato,
and Richard E. Turner

University of Cambridge

http://arxiv.org/abs/1506.04132

# Motivation

- Expectation propagation [Minka, 2001]
  - popular alternative to variational Bayes (VB) for approximate Bayesian inference
  - EP iteratively minimizes $KL(p||q)$ whereas VB minimizes $KL(q||p)$
  - $q$ is usually exponential family
  - EP: Iterative minimization as apposed to Assumed Density Filtering (ADF) where a data point is processed just once
  - works well for log-concave unimodal posterior distributions

# Motivation

- Expectation propagation [Minka, 2001]
  - popular alternative to variational Bayes (VB) for approximate Bayesian inference
  - EP iteratively minimizes $KL(p||q)$ whereas VB minimizes $KL(q||p)$
  - $q$ is usually exponential family
  - EP: Iterative minimization as apposed to Assumed Density Filtering (ADF) where a data point is processed just once
  - works well for log-concave unimodal posterior distributions
- EP is memory-intensive and does not scale well to big data
- Idea: Can we do stochastic EP (similar to SVI for VB)?

# Motivation

- Expectation propagation [Minka, 2001]
  - popular alternative to variational Bayes (VB) for approximate Bayesian inference
  - EP iteratively minimizes $KL(p||q)$ whereas VB minimizes $KL(q||p)$
  - $q$ is usually exponential family
  - EP: Iterative minimization as apposed to Assumed Density Filtering (ADF) where a data point is processed just once
  - works well for log-concave unimodal posterior distributions
- EP is memory-intensive and does not scale well to big data
- Idea: Can we do stochastic EP (similar to SVI for VB)?
  - Stochastic expectation propagation (SEP) [Li et al., 2015]
  - Expectation Propagation in the large-data limit [Dehaene and Barthelmé, 2015] introduces Averaged EP (AEP) which is easier to theoretically analyze than EP

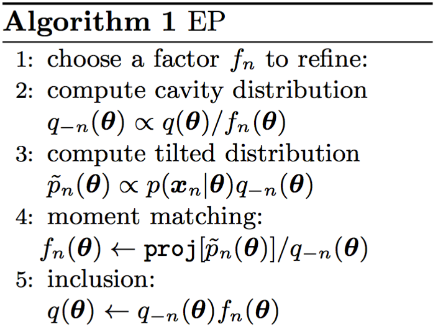
# Overview

- SEP: Stochastic EP that operates on mini-batches of data
- Global posterior approximation which is updated locally
- Similar predictive performance as EP
- Memory requirement reduced by a factor of $N$ compared to EP
- Much better uncertainty estimate than ADF
- Interesting connections to related work on distributed Bayesian inference (SMS: [Xu et al., 2014], EP as a way of life [Gelman et al., 2014])

# EP vs ADF

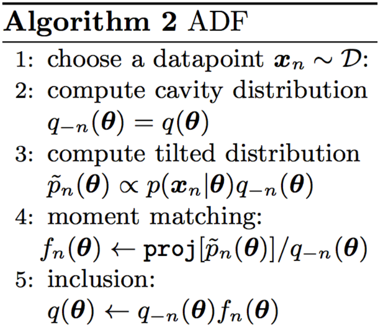
$$p(\boldsymbol{\theta}|\mathcal{D}) \propto p_0(\boldsymbol{\theta}) \prod_{n=1}^{N} p(x_n|\boldsymbol{\theta}) \tag{1}$$

$$p(\boldsymbol{\theta}|\mathcal{D}) \approx q(\boldsymbol{\theta}) \propto p_0(\boldsymbol{\theta}) \prod_{n=1}^{N} f_n(\boldsymbol{\theta}) \tag{2}$$



**Algorithm 1** EP

```
1: choose a factor $f_n$ to refine:
2: compute cavity distribution
   $q_{-n}(\boldsymbol{\theta}) \propto q(\boldsymbol{\theta})/f_n(\boldsymbol{\theta})$
3: compute tilted distribution
   $\tilde{p}_n(\boldsymbol{\theta}) \propto p(\boldsymbol{x}_n|\boldsymbol{\theta}) q_{-n}(\boldsymbol{\theta})$
4: moment matching:
   $f_n(\boldsymbol{\theta}) \leftarrow \mathtt{proj}[\tilde{p}_n(\boldsymbol{\theta})]/q_{-n}(\boldsymbol{\theta})$
5: inclusion:
   $q(\boldsymbol{\theta}) \leftarrow q_{-n}(\boldsymbol{\theta}) f_n(\boldsymbol{\theta})$
```



**Algorithm 2** ADF

```
1: choose a datapoint $\boldsymbol{x}_n \sim \mathcal{D}$:
2: compute cavity distribution
   $q_{-n}(\boldsymbol{\theta}) = q(\boldsymbol{\theta})$
3: compute tilted distribution
   $\tilde{p}_n(\boldsymbol{\theta}) \propto p(\boldsymbol{x}_n|\boldsymbol{\theta}) q_{-n}(\boldsymbol{\theta})$
4: moment matching:
   $f_n(\boldsymbol{\theta}) \leftarrow \mathtt{proj}[\tilde{p}_n(\boldsymbol{\theta})]/q_{-n}(\boldsymbol{\theta})$
5: inclusion:
   $q(\boldsymbol{\theta}) \leftarrow q_{-n}(\boldsymbol{\theta}) f_n(\boldsymbol{\theta})$
```

# Problems with EP

- Computing cavity distribution requires removal of current approximation
- Approximating each factor individually leads to $O(N)$ storage
- ADF does not require storage of individual factors
- ADF cannot use multiple passes through the data
- Uncertainty estimates of ADF are not well calibrated

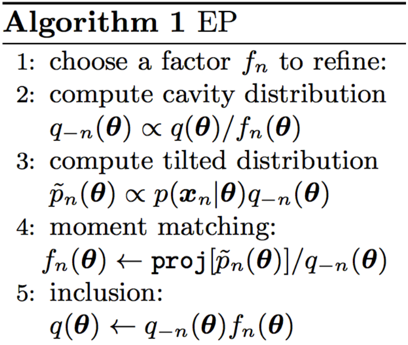
# Stochastic EP

- Idea: use a single factor which is the geometric mean of the individual approximations

$$f(\boldsymbol{\theta})^N = \prod_{n=1}^{N} f_n(\boldsymbol{\theta}) \approx \prod_{n=1}^{N} p(x_n|\boldsymbol{\theta}) \tag{3}$$

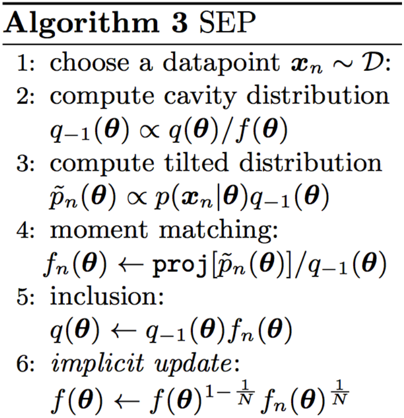
- Interpretation:
  - version of EP in which the approximating factors are tied
  - corrected version of ADF that prevents overfitting.
- Individual factors need not be stored (reduces memory requirement)

# EP vs SEP



**Algorithm 1** EP

```
1: choose a factor $f_n$ to refine:
2: compute cavity distribution
   $q_{-n}(\boldsymbol{\theta}) \propto q(\boldsymbol{\theta})/f_n(\boldsymbol{\theta})$
3: compute tilted distribution
   $\tilde{p}_n(\boldsymbol{\theta}) \propto p(\boldsymbol{x}_n|\boldsymbol{\theta})q_{-n}(\boldsymbol{\theta})$
4: moment matching:
   $f_n(\boldsymbol{\theta}) \leftarrow \texttt{proj}[\tilde{p}_n(\boldsymbol{\theta})]/q_{-n}(\boldsymbol{\theta})$
5: inclusion:
   $q(\boldsymbol{\theta}) \leftarrow q_{-n}(\boldsymbol{\theta})f_n(\boldsymbol{\theta})$
```



**Algorithm 3** SEP

```
1: choose a datapoint $\boldsymbol{x}_n \sim \mathcal{D}$:
2: compute cavity distribution
   $q_{-1}(\boldsymbol{\theta}) \propto q(\boldsymbol{\theta})/f(\boldsymbol{\theta})$
3: compute tilted distribution
   $\tilde{p}_n(\boldsymbol{\theta}) \propto p(\boldsymbol{x}_n|\boldsymbol{\theta})q_{-1}(\boldsymbol{\theta})$
4: moment matching:
   $f_n(\boldsymbol{\theta}) \leftarrow \texttt{proj}[\tilde{p}_n(\boldsymbol{\theta})]/q_{-1}(\boldsymbol{\theta})$
5: inclusion:
   $q(\boldsymbol{\theta}) \leftarrow q_{-1}(\boldsymbol{\theta})f_n(\boldsymbol{\theta})$
6: *implicit update*:
   $f(\boldsymbol{\theta}) \leftarrow f(\boldsymbol{\theta})^{1-\frac{1}{N}} f_n(\boldsymbol{\theta})^{\frac{1}{N}}$
```

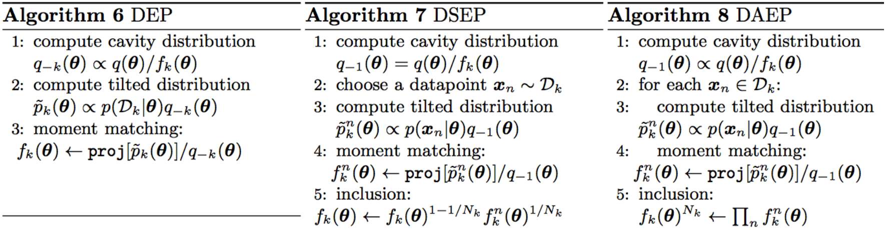
# Parallel SEP

- Minibatch version of SEP that operates on $M$ data points in parallel
- Parallel EP: $q(\boldsymbol{\theta}) = p_0(\boldsymbol{\theta}) \prod_{n \neq m} f_n(\boldsymbol{\theta}) \prod_m f_m(\boldsymbol{\theta})$
- Parallel SEP: $q(\boldsymbol{\theta}) = p_0(\boldsymbol{\theta}) f_{old}(\boldsymbol{\theta})^{N-M} \prod_m f_m(\boldsymbol{\theta})$
- Parallel SEP (implicit update): $f_{new}(\boldsymbol{\theta}) = f_{old}(\boldsymbol{\theta})^{1-M/N} \prod_{m=1}^{M} f_m(\boldsymbol{\theta})^{1/N}$
- $M = 1$ recovers vanilla SEP
- $M = N$ leads to AEP [Dehaene and Barthelmé, 2015]

# Distributed SEP

- Strong assumption: Dataset can be partitioned into $K$ subsets such that likelihood contribution within each subset is similar
- Idea: use different approximation within each of the subsets
- $q(\boldsymbol{\theta}) \propto p_0(\boldsymbol{\theta}) \prod_{k=1}^{K} f_k(\boldsymbol{\theta})^{N_k}$
- Approximating multiple likelihood terms requires MCMC in general. DSEP uses SEP within each subset.
- $K = 1$ recovers SEP and $K = N$ recovers EP

# DEP vs DSEP vs DAEP



**Algorithm 6** DEP

```
1: compute cavity distribution
   $q_{-k}(\boldsymbol{\theta}) \propto q(\boldsymbol{\theta})/f_k(\boldsymbol{\theta})$
2: compute tilted distribution
   $\tilde{p}_k(\boldsymbol{\theta}) \propto p(\mathcal{D}_k|\boldsymbol{\theta})q_{-k}(\boldsymbol{\theta})$
3: moment matching:
   $f_k(\boldsymbol{\theta}) \leftarrow \texttt{proj}[\tilde{p}_k(\boldsymbol{\theta})]/q_{-k}(\boldsymbol{\theta})$
```

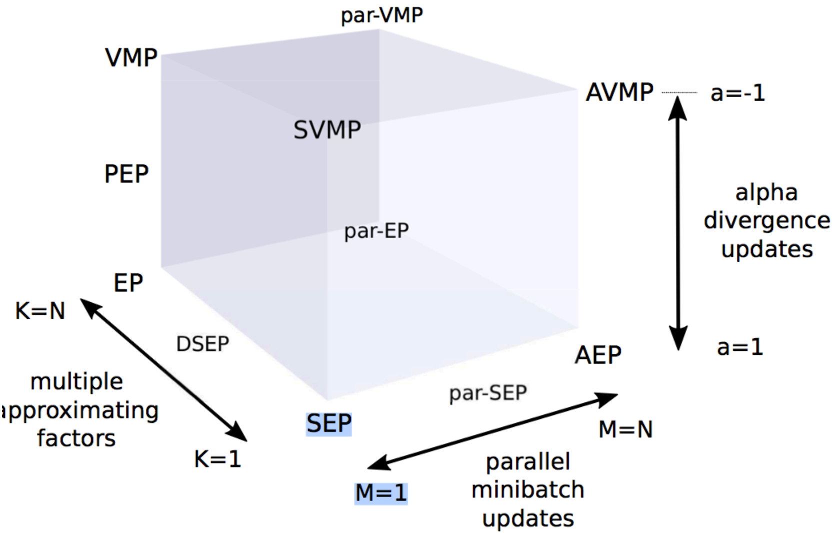
**Algorithm 7** DSEP

```
1: compute cavity distribution
   $q_{-1}(\boldsymbol{\theta}) = q(\boldsymbol{\theta})/f_k(\boldsymbol{\theta})$
2: choose a datapoint $\boldsymbol{x}_n \sim \mathcal{D}_k$
3: compute tilted distribution
   $\tilde{p}_k^n(\boldsymbol{\theta}) \propto p(\boldsymbol{x}_n|\boldsymbol{\theta})q_{-1}(\boldsymbol{\theta})$
4: moment matching:
   $f_k^n(\boldsymbol{\theta}) \leftarrow \texttt{proj}[\tilde{p}_k^n(\boldsymbol{\theta})]/q_{-1}(\boldsymbol{\theta})$
5: inclusion:
   $f_k(\boldsymbol{\theta}) \leftarrow f_k(\boldsymbol{\theta})^{1-1/N_k} f_k^n(\boldsymbol{\theta})^{1/N_k}$
```

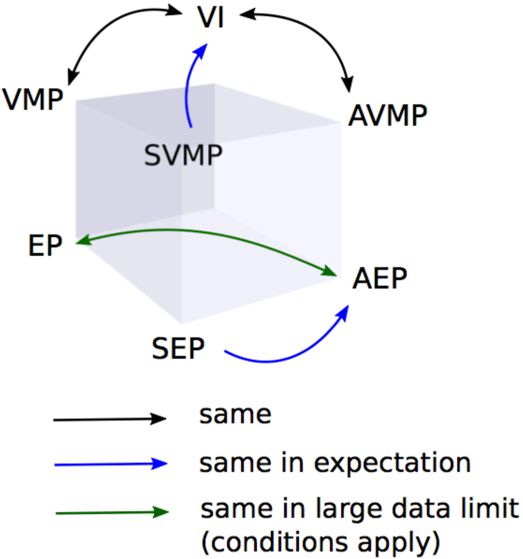
**Algorithm 8** DAEP

```
1: compute cavity distribution
   $q_{-1}(\boldsymbol{\theta}) \propto q(\boldsymbol{\theta})/f_k(\boldsymbol{\theta})$
2: for each $\boldsymbol{x}_n \in \mathcal{D}_k$:
3:     compute tilted distribution
       $\tilde{p}_k^n(\boldsymbol{\theta}) \propto p(\boldsymbol{x}_n|\boldsymbol{\theta})q_{-1}(\boldsymbol{\theta})$
4:     moment matching:
       $f_k^n(\boldsymbol{\theta}) \leftarrow \texttt{proj}[\tilde{p}_k^n(\boldsymbol{\theta})]/q_{-1}(\boldsymbol{\theta})$
5: inclusion:
   $f_k(\boldsymbol{\theta})^{N_k} \leftarrow \prod_n f_k^n(\boldsymbol{\theta})$
```

# Relationship to other methods



## B) Relationships between fixed points



VI

VMP

AVMP

SVMP

EP

AEP

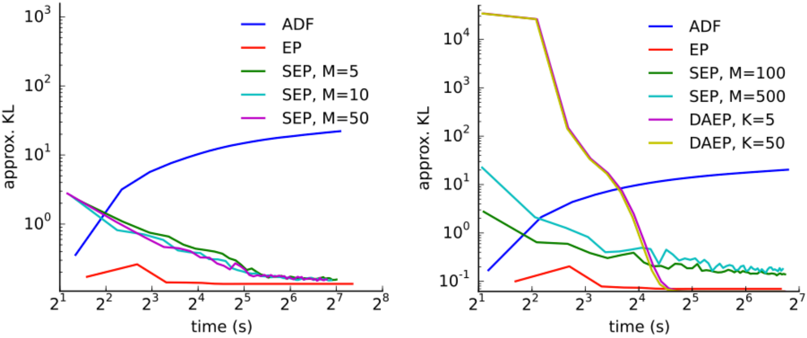
SEP

same

same in expectation

same in large data limit
(conditions apply)

# Bayesian probit regression on toy data

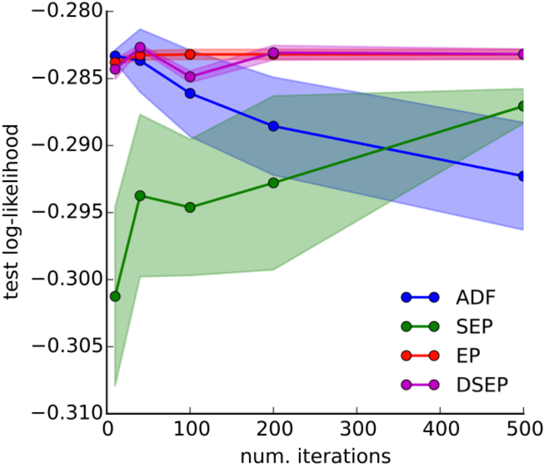


(a)

(b)

- Toy data: $N = 5000$, $D = 4$. Ground truth: NUTS
- $x$ sampled from Gaussian distribution (a) or from a Mixture of Gaussians with J = 5 components (b)

# Bayesian probit regression on MNIST



(c)

# Results on UCI: SEP (K=1, M=1)

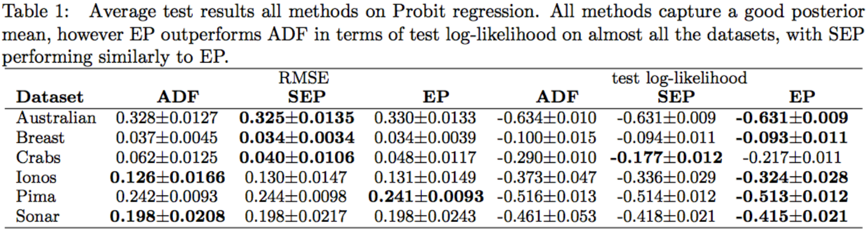


Table 1: Average test results all methods on Probit regression. All methods capture a good posterior mean, however EP outperforms ADF in terms of test log-likelihood on almost all the datasets, with SEP performing similarly to EP.

| | | RMSE | | | test log-likelihood | |
|---|---|---|---|---|---|---|
| **Dataset** | **ADF** | **SEP** | **EP** | **ADF** | **SEP** | **EP** |
| Australian | 0.328±0.0127 | **0.325±0.0135** | 0.330±0.0133 | -0.634±0.010 | -0.631±0.009 | **-0.631±0.009** |
| Breast | 0.037±0.0045 | **0.034±0.0034** | 0.034±0.0039 | -0.100±0.015 | -0.094±0.011 | **-0.093±0.011** |
| Crabs | 0.062±0.0125 | **0.040±0.0106** | 0.048±0.0117 | -0.290±0.010 | **-0.177±0.012** | -0.217±0.011 |
| Ionos | **0.126±0.0166** | 0.130±0.0147 | 0.131±0.0149 | -0.373±0.047 | -0.336±0.029 | **-0.324±0.028** |
| Pima | 0.242±0.0093 | 0.244±0.0098 | **0.241±0.0093** | -0.516±0.013 | -0.514±0.012 | **-0.513±0.012** |
| Sonar | **0.198±0.0208** | 0.198±0.0217 | 0.198±0.0243 | -0.461±0.053 | -0.418±0.021 | **-0.415±0.021** |

# Results using Probabilistic BackProp

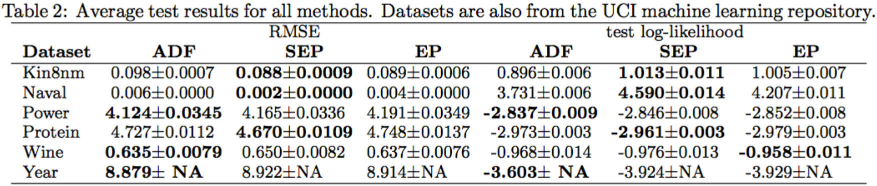


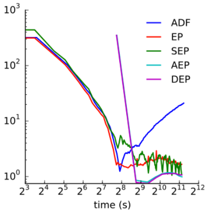
Table 2: Average test results for all methods. Datasets are also from the UCI machine learning repository.

| | RMSE | | | test log-likelihood | | |
|---|---|---|---|---|---|---|
| **Dataset** | **ADF** | **SEP** | **EP** | **ADF** | **SEP** | **EP** |
| Kin8nm | 0.098±0.0007 | **0.088±0.0009** | 0.089±0.0006 | 0.896±0.006 | **1.013±0.011** | 1.005±0.007 |
| Naval | 0.006±0.0000 | **0.002±0.0000** | 0.004±0.0000 | 3.731±0.006 | **4.590±0.014** | 4.207±0.011 |
| Power | **4.124±0.0345** | 4.165±0.0336 | 4.191±0.0349 | **-2.837±0.009** | -2.846±0.008 | -2.852±0.008 |
| Protein | 4.727±0.0112 | **4.670±0.0109** | 4.748±0.0137 | -2.973±0.003 | **-2.961±0.003** | -2.979±0.003 |
| Wine | **0.635±0.0079** | 0.650±0.0082 | 0.637±0.0076 | -0.968±0.014 | -0.976±0.013 | **-0.958±0.011** |
| Year | **8.879± NA** | 8.922±NA | 8.914±NA | **-3.603± NA** | -3.924±NA | -3.929±NA |

# Memory consumption

| **Dataset** | $N$ | $d$ | MB reduction |
|---|---:|---:|---:|
| Kin8nm | 8,192 | 8 | 58MB |
| Naval | 11,934 | 16 | 147MB |
| Power Plant | 9,568 | 4 | 37MB |
| Protein | 45,730 | 9 | **694MB** |
| Wine | 1,599 | 11 | 14MB |
| Year | 515,340 | 90 | **65107MB** |

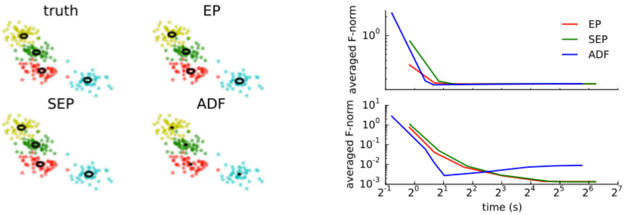
(a)



(b)

Figure 7: (a) Memory reduction figures on regression datasets. (b) Performance of EP methods on Bayesian logistic regression with sampling moment computations.

# Mixture of Gaussians for clustering

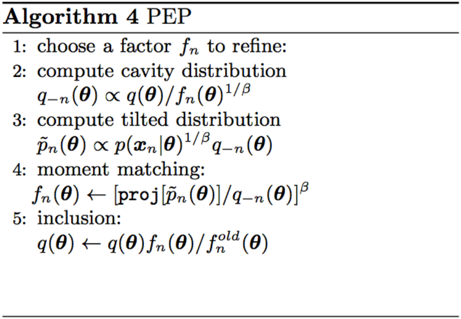


(a)

(b)

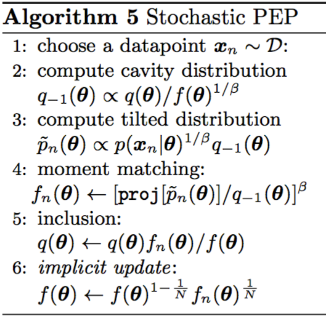
Figure 4: Posterior approximation for the mean of the Gaussian components. (a) shows posterior approximations over the cluster means (98% confidence level). The coloured dots indicate the true label (top-left) or the inferred cluster assignments (the rest). In (b) we show the error of the approximations as measured by the averaged Frobenius norm of the difference between the the closest means posterior samples and EP approximations, mean (top) and covariance (bottom).

# Stochastic power EP



**Algorithm 4** PEP

```
choose a factor $f_n$ to refine:
compute cavity distribution
    $q_{-n}(\boldsymbol{\theta}) \propto q(\boldsymbol{\theta})/f_n(\boldsymbol{\theta})^{1/\beta}$
compute tilted distribution
    $\tilde{p}_n(\boldsymbol{\theta}) \propto p(\boldsymbol{x}_n|\boldsymbol{\theta})^{1/\beta} q_{-n}(\boldsymbol{\theta})$
moment matching:
    $f_n(\boldsymbol{\theta}) \leftarrow [\texttt{proj}[\tilde{p}_n(\boldsymbol{\theta})]/q_{-n}(\boldsymbol{\theta})]^{\beta}$
inclusion:
    $q(\boldsymbol{\theta}) \leftarrow q(\boldsymbol{\theta}) f_n(\boldsymbol{\theta}) / f_n^{old}(\boldsymbol{\theta})$
```



**Algorithm 5** Stochastic PEP

```
choose a datapoint $\boldsymbol{x}_n \sim \mathcal{D}$:
compute cavity distribution
    $q_{-1}(\boldsymbol{\theta}) \propto q(\boldsymbol{\theta})/f(\boldsymbol{\theta})^{1/\beta}$
compute tilted distribution
    $\tilde{p}_n(\boldsymbol{\theta}) \propto p(\boldsymbol{x}_n|\boldsymbol{\theta})^{1/\beta} q_{-1}(\boldsymbol{\theta})$
moment matching:
    $f_n(\boldsymbol{\theta}) \leftarrow [\texttt{proj}[\tilde{p}_n(\boldsymbol{\theta})]/q_{-1}(\boldsymbol{\theta})]^{\beta}$
inclusion:
    $q(\boldsymbol{\theta}) \leftarrow q(\boldsymbol{\theta}) f_n(\boldsymbol{\theta}) / f(\boldsymbol{\theta})$
implicit update:
    $f(\boldsymbol{\theta}) \leftarrow f(\boldsymbol{\theta})^{1-\frac{1}{N}} f_n(\boldsymbol{\theta})^{\frac{1}{N}}$
```

Thank you!

# References I

Dehaene, G. and Barthelmé, S. (2015).
Expectation propagation in the large-data limit.
*arXiv preprint arXiv:1503.08060*.

Gelman, A., Vehtari, A., Jylänki, P., Robert, C., Chopin, N., and Cunningham, J. P. (2014).
Expectation propagation as a way of life.
*arXiv preprint arXiv:1412.4869*.

Li, Y., Hernandez-Lobato, J. M., and Turner, R. E. (2015).
Stochastic expectation propagation.
*arXiv preprint arXiv:1506.04132*.

Minka, T. P. (2001).
Expectation propagation for approximate bayesian inference.
In *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*, pages 362–369. Morgan Kaufmann Publishers Inc.

# References II

Xu, M., Lakshminarayanan, B., Teh, Y. W., Zhu, J., and Zhang, B. (2014).

Distributed Bayesian Posterior Sampling via Moment Sharing.

In *Advances in Neural Information Processing Systems*.