Approximate Decentralized Bayesian Inference

Trevor Campbell, Jonathan P. How • Laboratory for Information and Decision Systems, MIT

Introduction

- **Problem:** Unsupervised learning in conditionally independent Bayesian models over a decentralized, ad-hoc network
- **Challenge:** Approximate inference breaks model structure required to combine posteriors

The Naïve Approach

- **Bayes' Approximate Posterior Combination**
  - Latent parameters: $\Theta = \{\theta_k\}_{k=1}^K$, data subsets: $Y = \{Y_i\}_{i=1}^N$

$$p(\Theta|Y) \propto p(\Theta)\,p(Y|\Theta) = p(\Theta)\prod_{i=1}^N p(Y_i|\Theta) \propto p(\Theta)\prod_{i=1}^N \frac{p(\Theta|Y_i)}{p(\Theta)} \quad (1)$$

  - Compute local variational posteriors $q_i(\Theta) \simeq p(\Theta|Y_i)$ and combine via (1):

$$q_i(\Theta) = \prod_k q_{\lambda_{ik}}(\theta_k)\ \forall i = 1,\dots,N, \quad q_0(\Theta) = \prod_k q_{\lambda_{0k}}(\theta_k)$$
$$\therefore q(\Theta) = \prod_k q_{\lambda_k}(\theta_k) \text{ where } \lambda_k = (1-N)\lambda_{0k} + \sum_i \lambda_{ik} \quad (2)$$

- **Example 1: Decentralized Gaussian Inference**
  - Works well for this simple model ($N = 10$ agents, 100 observations)

(a) Batch **(b)** Naïve Decentralized

Figure: (1a): Batch posterior of μ in black, with histogram of observed data. (1b): Decentralized posterior of μ in black, individual posteriors in color and correspondingly colored histogram of observed data.

- **Example 2: Decentralized 3-Component GMM Inference**
  - Fails for more complex models ($N = 10$ agents, 30 observations)

(a) Batch **(b)** Naïve Decentralized

Figure: (2a): Samples from the true posterior over μ, π. Particle position on the simplex (with $\pi_3 = 1 - \pi_1 - \pi_2$) represents the weights, while RGB color coordinates represent the three means. (2b): Samples from the naïvely constructed decentralized approximate posterior.

- **Why Naïve Fails: Approximate Inference Breaks Symmetry**
  - $q_i(\Theta)$ missing symmetry in unsupervised models required to combine via (1)

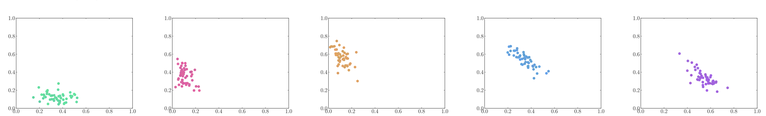


Figure: Batch variational Bayes approximate posterior samples from 5 random restarts. Comparison to Figure 2a shows that the approximate inference algorithm tends to converge to a random component of the permutation symmetry in the true posterior.

Approximate Merging of Posteriors with Symmetry (AMPS)

- **Step 1: Artificial Reintroduction of Symmetry**
  - *Symmetrize* approximate posteriors by summing over permutations P:

$$\tilde{q}_i(\Theta) \propto \sum_P \prod_k q_{P\lambda_{ik}}(\theta_k) \quad (3)$$

  - This reintroduces original model structure, i.e. for any permutation P,

$$\tilde{q}_i(P\theta_1,\dots,P\theta_K) = \tilde{q}_i(\theta_1,\dots,\theta_K) \quad (4)$$

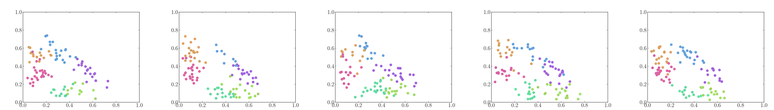


Figure: Samples from the symmetrized batch variational Bayes approximate posterior from 5 random restarts. Comparison to Figure 2a shows that symmetrization reintroduces the structure of the true posterior to the approximate posteriors.

- **Step 2: Combine Using Bayes' Rule**
  - Apply (1) as before with $\tilde{q}_i$ instead of q_i:

$$q(\cdot) \propto q_0(\cdot)^{1-N}\prod_i \tilde{q}_i(\cdot) = \sum_{\{P_i\}_i}\prod_k \left[q_{\lambda_{0k}}(\theta_k)^{1-N}\prod_i q_{P_i\lambda_{ik}}(\theta_k)\right] \quad (5)$$

  - This distribution is intractable to use directly; contains $(K!)^N$ terms
- **Step 3: Optimize Out Unnecessary Structure**
  - Solve a combinatorial optimization over permutation matrices $\{P_i\}_{i=1}^N$:

$$\{P_i^\star\}_{i=1}^N \leftarrow \arg\max_{\{P_i\}_{i=1}^N} \sum_k A_k\left((1-N)\lambda_{0k} + \sum_i P_i\lambda_{ik}\right) \quad (6)$$
$$\text{s.t. } P_i \in S \quad \forall i$$

  where $A_k(\cdot)$ is the log-partition function for parameter θ_k
  - Solving (6) is equivalent to finding the maximum weight component of (5)
- **Step 4: Output the Decentralized Posterior**

$$q^\star(\Theta) = \prod_k q_{\lambda_k^\star}(\theta_k) \text{ where } \lambda_k^\star = (1-N)\lambda_{0k} + \sum_i P_i^\star\lambda_{ik} \quad (7)$$

Conclusions

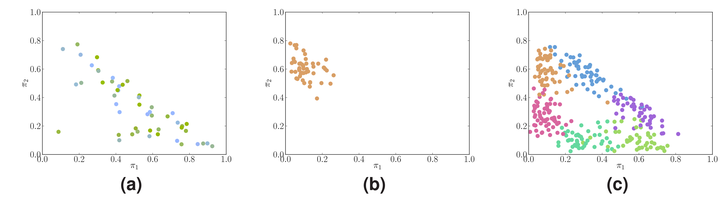
- AMPS provides ad-hoc, decentralized Bayesian learning
- Easy to derive and implement for exp family approximations
- Surprisingly, can improve model vs. batch/distributed methods

← **Paper!** **Code!** →

Paper: `http://arxiv.org/abs/1403.7471`
Code: `http://github.com/trevorcampbell/amps/`

Experiments

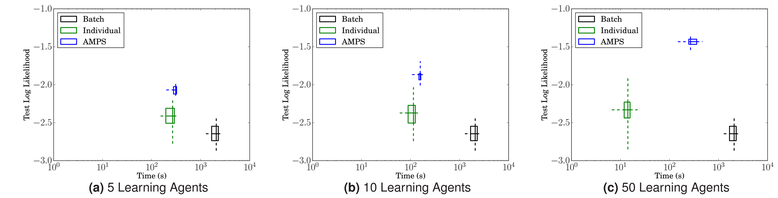
- **Synthetic: Mixture Model, Revisited**
  - Solve (6) by swapping rows of P_i until no swap improves objective
  - AMPS improves model significantly w.r.t. individual agent models
  - Works well despite high individual agent uncertainty



(a) **(b)** **(c)**

Figure: (5a): Samples from a typical individual posterior distribution $q_i(\Theta)$ from variational Bayesian inference. (5b): Samples from the decentralized posterior output by AMPS. Comparison to Figure 5a shows that the AMPS posterior merging procedure improves the posterior possessed by each agent significantly. (5c): Samples from the symmetrized decentralized posterior, for comparison to Figure 2a.

- **Real Data: Latent Dirichlet Allocation**
  - 20 Newsgroups dataset – model symmetry in ordering of topics
  - Solve (6) by iterating max-weight bipartite matching problems on rows of P_i
  - AMPS improves test log-likelihood over individual agents & batch inference
  - Solves smaller inference problems, less susceptible to local optima



(a) 5 Learning Agents **(b)** 10 Learning Agents **(c)** 50 Learning Agents

Figure: Plots of log likelihood on the test data for 5 (6a), 10 (6b), and 50 (6c) learning agents. The boxes bound the 25th and 75th percentiles of the data with medians shown in the interior, and whiskers show the maximum and minimum values.

- **Hierarchical/Streaming AMPS**
  - AMPS naturally extends to streaming data (20 Newsgroups)
  - Improves test log-likelihood over batch, SDA-Bayes (Broderick, NIPS '13)
  - AMPS10x10 is effectively 100 agents with hierarchical optimization
  - Hierarchical optimization reduces CPU time without sacrificing quality

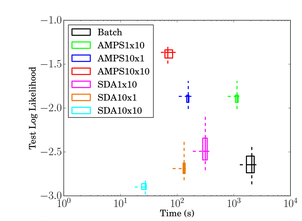


Figure: Comparison with SDA-Bayes. The $A \times B$ in the legend names refer to using A learning agents, where each splits their individual batches of data into B streaming subbatches.

ONR MURI N000141110688 E-Mail: `tdjc@mit.edu`