Distributed Inference for LDA

D. Newman, A. Asuncion, P. Smyth, M. Welling

Async. Dist. Learning of Topic Models

A. Asuncion, P. Smyth, M. Welling

Presented by:

Golnar Sheikhshab

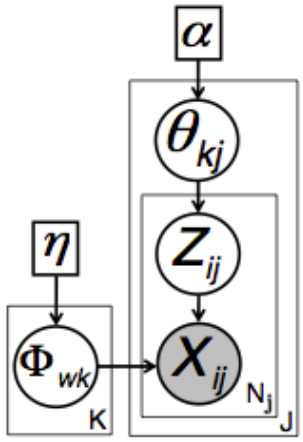
Outline

- Latent Dirichlet Allocation
- Distributed Inference for LDA
- Asynchronous Distributed Learning for Topic Models

Latent Dirichlet Allocation (LDA)

- State of the art topic modeling method
- Clusters the words into topics
  - Information Retrieval
  - Data visualization
  - …
- Also has applications in
  - Image Processing
  - Bioinformatics

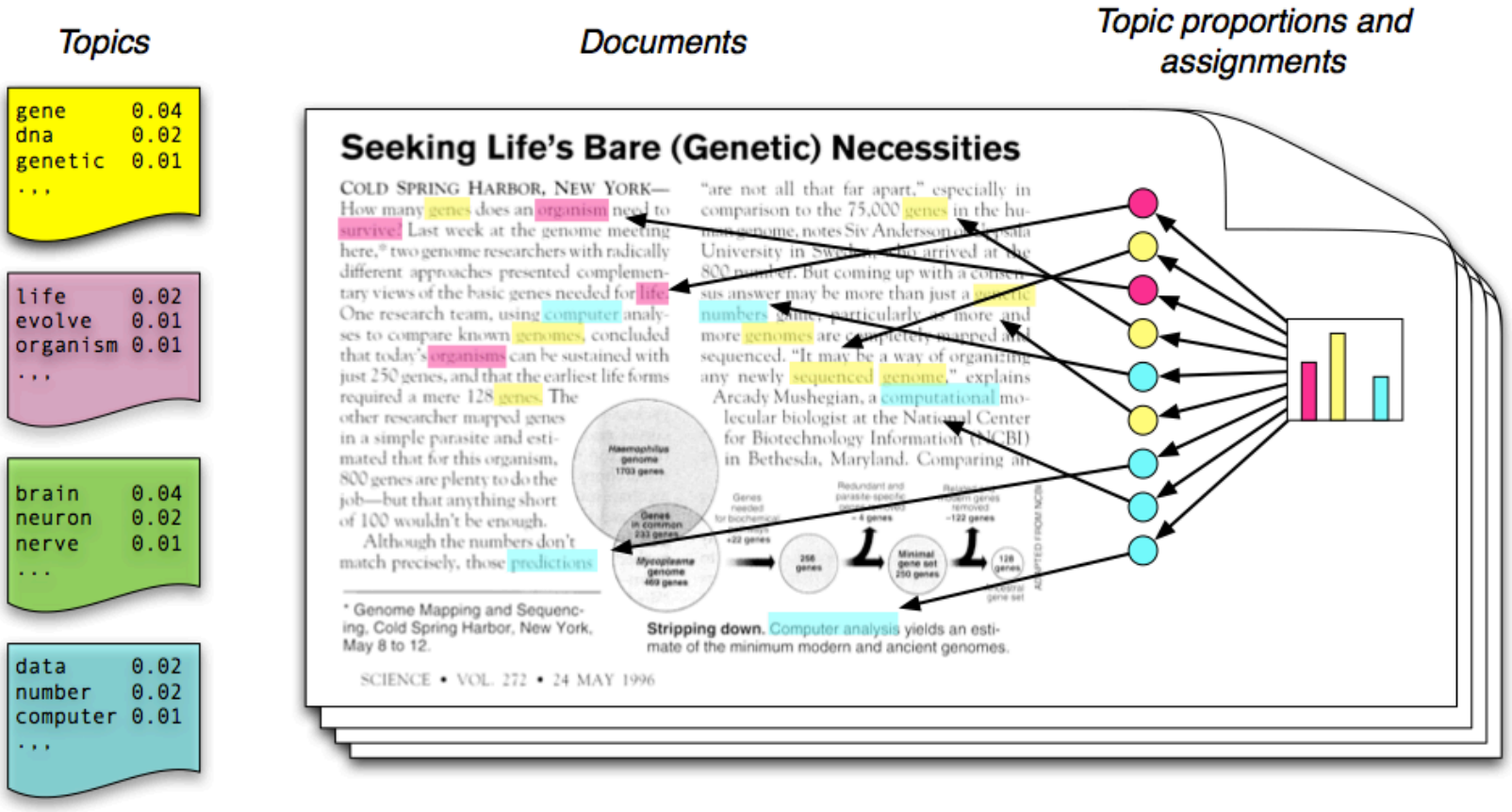
LDA



- For each Document:
  1. Randomly choose a distribution over topics.
  2. For each word in the document
     a) Randomly choose a topic from the distribution over topics in step #1.
     b) Randomly choose a word from the corresponding distribution over the vocabulary.

$$\theta_{k,j} \sim D[\alpha] \qquad \phi_{w,k} \sim D[\eta] \qquad z_{ij} \sim \theta_{k,j} \qquad x_{ij} \sim \phi_{w,z_{ij}}$$

LDA


Topics
Documents
Topic proportions and assignments
gene 0.04
dna 0.02
genetic 0.01
...
life 0.02
evolve 0.01
organism 0.01
...
brain 0.04
neuron 0.02
nerve 0.01
...
data 0.02
number 0.02
computer 0.01
...
Seeking Life's Bare (Genetic) Necessities
COLD SPRING HARBOR, NEW YORK—
How many genes does an organism need to survive? Last week at the genome meeting here,* two genome researchers with radically different approaches presented complementary views of the basic genes needed for life. One research team, using computer analyses to compare known genomes, concluded that today's organisms can be sustained with just 250 genes, and that the earliest life forms required a mere 128 genes. The other researcher mapped genes in a simple parasite and estimated that for this organism, 800 genes are plenty to do the job—but that anything short of 100 wouldn't be enough.
Although the numbers don't match precisely, those predictions
"are not all that far apart," especially in comparison to the 75,000 genes in the human genome, notes Siv Andersson of Uppsala University in Sweden, who arrived at the 800 number. But coming up with a consensus answer may be more than just a genetic numbers game, particularly as more and more genomes are completely mapped and sequenced. "It may be a way of organizing any newly sequenced genome," explains Arcady Mushegian, a computational molecular biologist at the National Center for Biotechnology Information (NCBI) in Bethesda, Maryland. Comparing an
* Genome Mapping and Sequencing, Cold Spring Harbor, New York, May 8 to 12.
Stripping down. Computer analysis yields an estimate of the minimum modern and ancient genomes.
SCIENCE • VOL. 272 • 24 MAY 1996

LDA Parameter Estimation

- Gibbs Sampling
- Observed variables: wij
- Latent variables:
  - Zij
  - θkj and φwk (marginalized out)

$$P(z_{ij} = k | z^{\neg ij}, w) \propto \frac{N_{wk}^{\neg ij} + \eta}{\sum_w N_{wk}^{\neg ij} + W\eta} \left(N_{jk}^{\neg ij} + \alpha \right)$$

Distributed Inference Algorithms

- Approximate Distributed Inference (AD-LDA)
- Hierarchical Distributed Inference (HD-LDA)
  - Not to be confused with Hierarchical LDA

AD-LDA

- Run LDA on subsets of data on different processors + a synchronization step

$$p(z_{ijp} = k|\mathbf{z}_p^{\neg ijp}, \mathbf{x}, \alpha, \beta) \propto (\alpha + n_{k|jp}^{\neg ijp})(\beta + n_{x_{ij}|kp}^{\neg ijp})(W\beta + n_{kp}^{\neg ijp})^{-1}$$

$$n_{w|k} \leftarrow n_{w|k} + \sum_p (n_{w|kp} - n_{w|k}), \qquad n_{w|kp} \leftarrow n_{w|k}$$

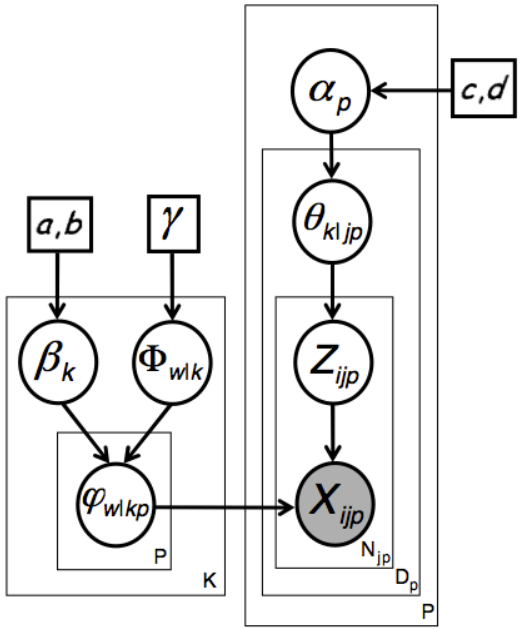
- Note: $n_{w|kp}$ is not the result of separate LDA models running on separate data.

$$\sum_{w,k} n_{w|kp} = N$$

HD-LDA

$$\Phi_{w|k} \sim \mathcal{D}[\gamma]$$
$$\varphi_{w|kp} \sim \mathcal{D}[\beta_k \Phi_{w|k}]$$
$$z_{ijp} \sim \theta_{k|jp}$$

$$\alpha_p \sim \mathcal{G}[c,d]$$
$$\beta_k \sim \mathcal{G}[a,b]$$
$$\theta_{k|jp} \sim \mathcal{D}[\alpha_p]$$
$$x_{ijp} \sim \varphi_{w|z_{ijp}}$$



$$p(z_{ijp}=k|\mathbf{z}_p^{\neg ijp},\mathbf{x},\alpha,\beta,\Phi) \propto (\alpha_p + n_{k|jp}^{\neg ijp})(\beta_k \Phi_{x_{ij}|k} + n_{x_{ij}|kp}^{\neg ijp})(\beta_k + n_{kp}^{\neg ijp})^{-1}$$

$$\alpha_p \leftarrow \frac{c-1+\alpha_p \sum_{jk}\left[\Psi(\alpha_p + n_{k|jp}) - \Psi(\alpha_p)\right]}{d + K\sum_j \left[\Psi(K\alpha_p + n_{jp}) - \Psi(K\alpha_p)\right]}$$

$$\beta_k \leftarrow \frac{a-1+\beta_k \sum_{wp} \Phi_{w|k}\left[\Psi(\beta_k \Phi_{w|k} + n_{w|kp}) - \Psi(\beta_k \Phi_{w|k})\right]}{b + \sum_p \left[\Psi(\beta_k + n_{kp}) - \Psi(\beta_k)\right]}$$

$$\Phi_{w|k} \leftarrow \frac{\gamma-1+\sum_p \beta_k \Phi_{w|k}\left[\Psi(\beta_k \Phi_{w|k} + n_{w|kp}) - \Psi(\beta_k \Phi_{w|k})\right]}{\gamma W - W + \sum_{wp} \beta_k \Phi_{w|k}\left[\Psi(\beta_k \Phi_{w|k} + n_{w|kp}) - \Psi(\beta_k \Phi_{w|k})\right]}$$

HD-LDA

- Hyper-parameter selection
  - Guided by prior experience with AD-LDA
  - Note: $\sum_{w,k} n_{w|kp} \approx \frac{N}{P}$

    a and b were chosen such that $\beta_k = \frac{P-1}{P} N$

    c and d chosen such that $\alpha_p = 0.1$

    $\gamma = 1 + \frac{1}{T}$

  - Critical in getting good results

Experiments

- Why would AD-LDA work?
- Toy example 3 words, 2 topics

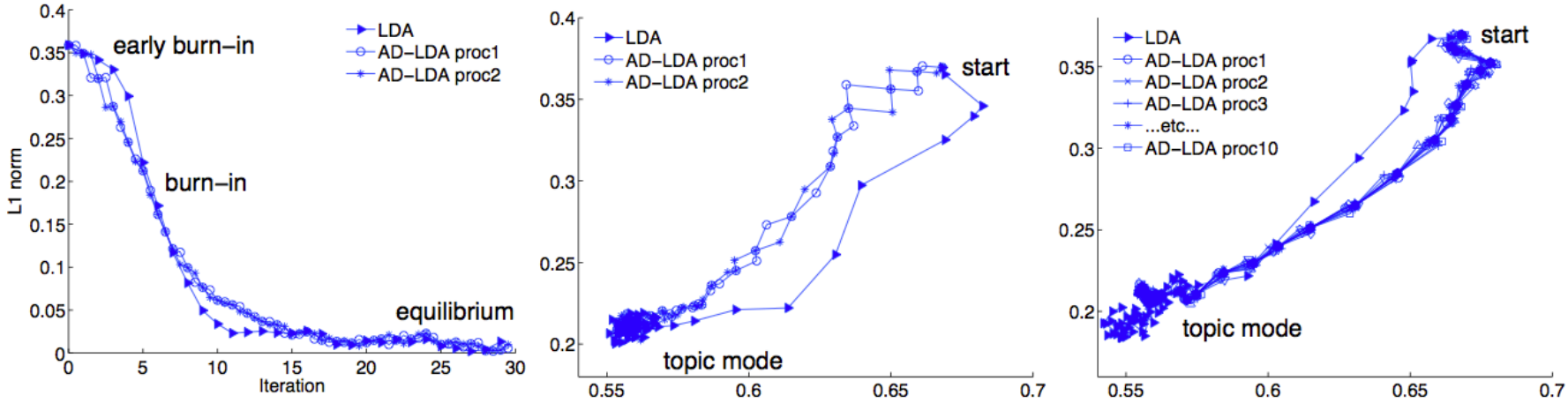


Figure 2: (Left) L_1 distance to the mode for LDA and for P=2 AD-LDA. (Center) Projection of topics onto simplex, showing convergence to mode. (Right) Same setup as center panel, but with $P = 10$ processors.

Experiments (Cont.)

- Prediction power

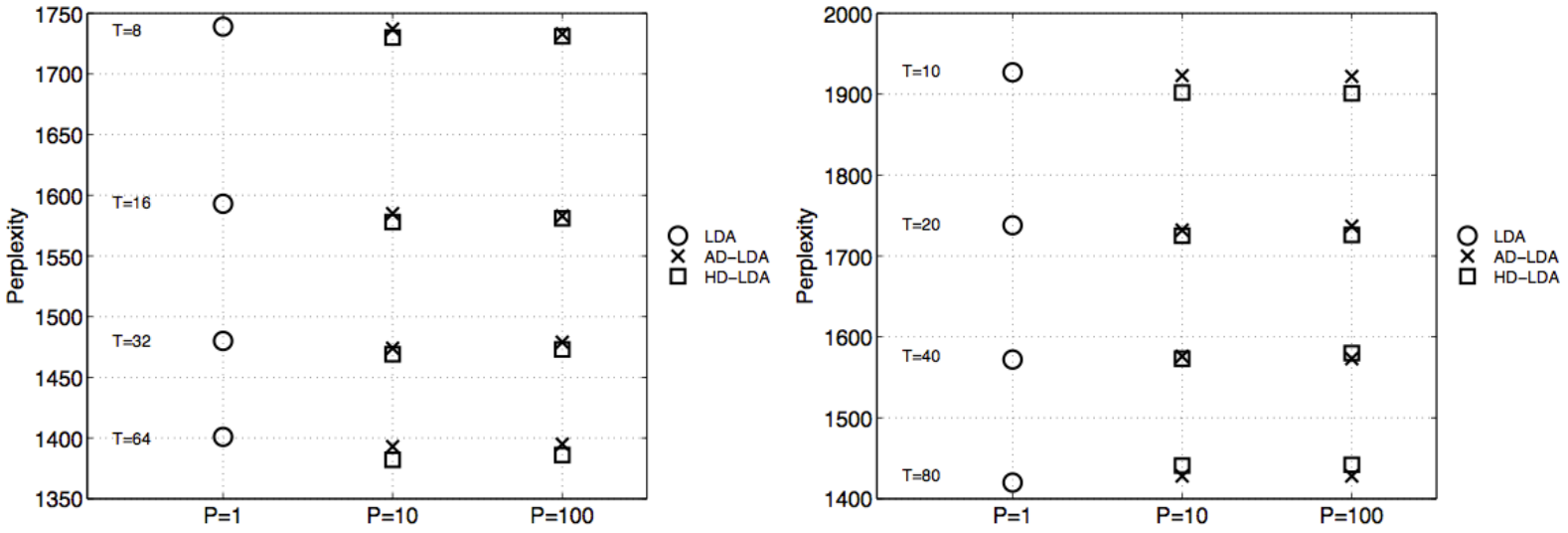
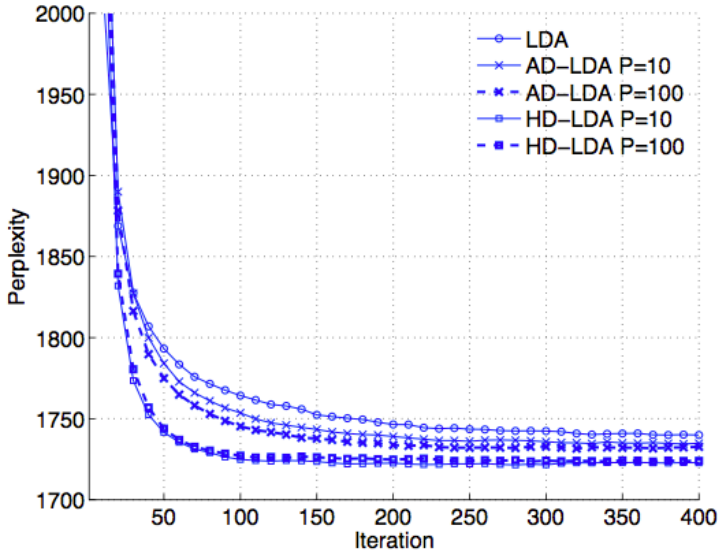


Figure 3: Test perplexity of models versus number of processors P for KOS (left) and NIPS (right). P=1 corresponds to LDA (circles), and AD-LDA (crosses), and HD-LDA (squares) are shown at P=10 and 100 .

Experiments (Cont.)

- Prediction power vs. iteration number and number of topics



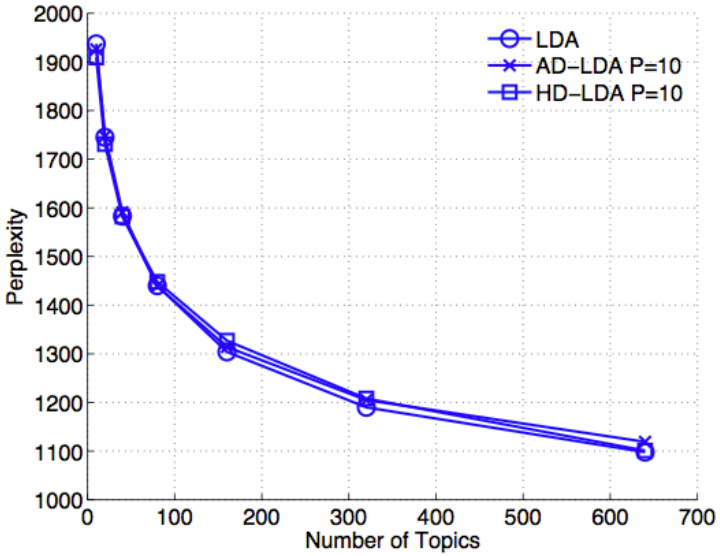


Figure 4: (Left) Test perplexity versus iteration. (Right) Test perplexity versus number of topics.

Experiments (Cont.)

- IR application TREC'S AP and FR collections

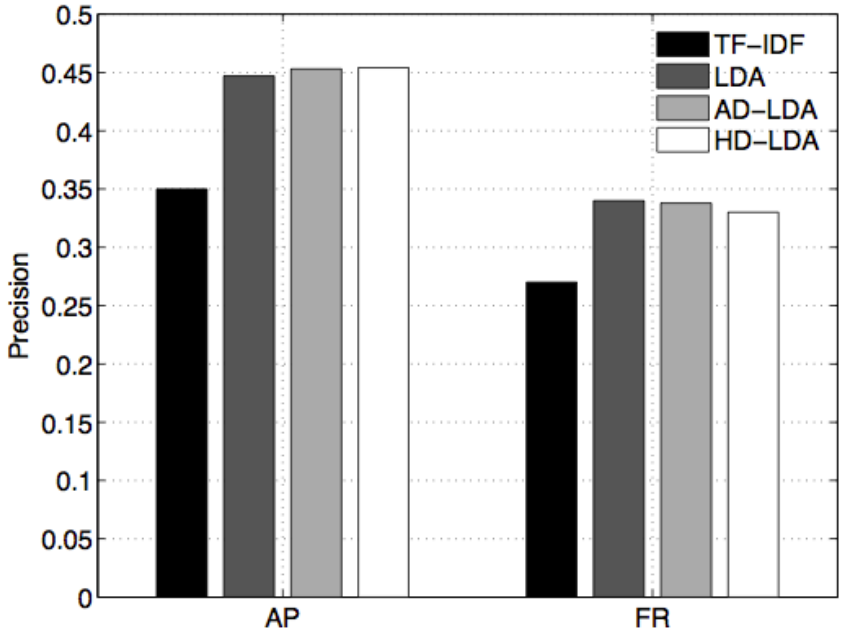
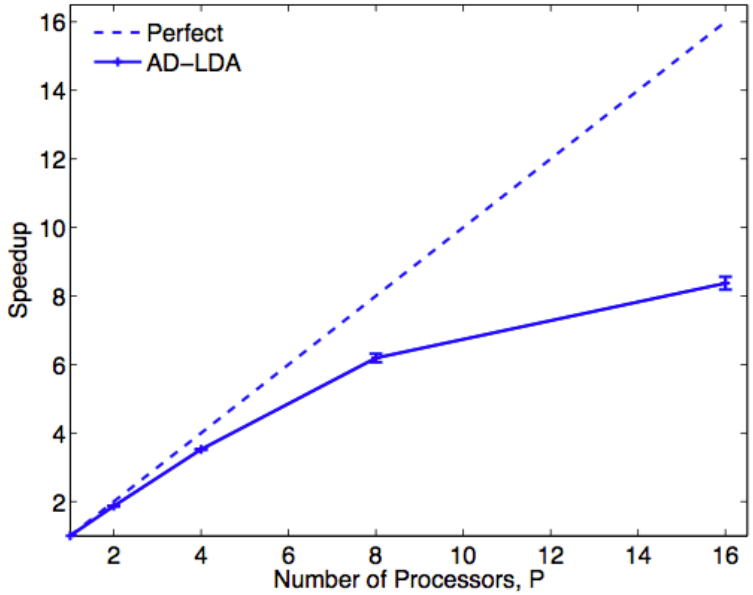


Figure 5: (Left) Precision/recall results.

Experiments (Cont.)

- Speedup experiment:
  - On one 16-processor machine: not reflecting the communication time



. (Right). Parallel speedup results.

Asynchronous Methods

- Advantages
  - No global synchronization step is required
  - Extremely fault-tolerant
  - Heterogeneous machines can be used
  - New processors and data can arrive any time

- Empirically
  - Robust
  - Converging fast

Asynchronous methods (Cont.)

- Still 2 steps:
  - Gibbs sampling
  - Synchronization
    - Gossip-based

$$P(z_{pij} = k | z_p^{\neg ij}, w_p) \propto \frac{(N^{\neg p} + N^p)_{wk}^{\neg ij} + \eta}{\sum_w (N^{\neg p} + N^p)_{wk}^{\neg ij} + W\eta} \left(N_{pjk}^{\neg ij} + \alpha\right)$$

Algorithm 1 Async-LDA

for each processor p in parallel **do**
  repeat
    Sample z^p locally (Equation 2)
    Receive N_{wk}^g from random proc g
    Send N_{wk}^p to proc g
    if p has met g before **then**
      $N_{wk}^{\neg p} \leftarrow N_{wk}^{\neg p} - \tilde{N}_{wk}^g + N_{wk}^g$
    else
      $N_{wk}^{\neg p} \leftarrow N_{wk}^{\neg p} + N_{wk}^g$
    end if
  until convergence
end for

Asynchronous methods (Cont.)

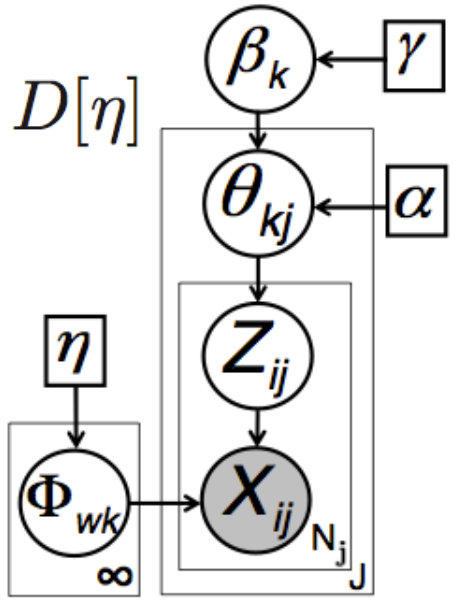
- One over Hierarchical Dirichlet Process (HDP)

- HDP

$$\beta_k \sim D[\gamma/L] \qquad \theta_{k,j} \sim D[\alpha\beta_k] \qquad \phi_{w,k} \sim D[\eta]$$

$$z_{ij} \sim \theta_{k,j} \qquad x_{ij} \sim \phi_{w,z_{ij}}$$

When L goes to infinity



Parallel-HDP & Async-HDP

Algorithm 2 Parallel-HDP

repeat
  for each processor p in parallel **do**
    Sample z^p locally
    Send N^p_{wk}, β^p_k to master node
  end for
  $N_{wk} \leftarrow \sum_p N^p_{wk}$
  $\beta_k \leftarrow (\sum_p \beta^p_k)$ / P
  Resample α, β_k, γ globally
  Distribute $N_{wk}, \alpha, \beta_k, \gamma$ to all processors
until convergence

Algorithm 3 Async-HDP

for each processor p in parallel **do**
  repeat
    Sample z^p and then $\alpha^p, \beta^p_k, \gamma^p$ locally
    Receive $N^g_{wk}, \alpha^g, \beta^g_k$ from random proc g
    Send $N^p_{wk}, \alpha^p, \beta^p_k$ to proc g
    if p has met g before **then**
      $N^{\neg p}_{wk} \leftarrow N^{\neg p}_{wk} - \tilde{N}^g_{wk} + N^g_{wk}$
    else
      $N^{\neg p}_{wk} \leftarrow N^{\neg p}_{wk} + N^g_{wk}$
    end if
    $\alpha^p \leftarrow (\alpha^p + \alpha^g)$ / 2 and $\beta^p_k \leftarrow (\beta^p_k + \beta^g_k)$ / 2
  until convergence
end for

$$P(z_{pij} = k | z_p^{\neg ij}, w_p) \propto \begin{cases} \frac{(N^{\neg p} + N^p)^{\neg ij}_{wk} + \eta}{\sum_w (N^{\neg p} + N^p)^{\neg ij}_{wk} + W\eta} \left(N^{\neg ij}_{pjk} + \alpha^p \beta^p_k \right), & \text{if } k \leq K_p \\ \frac{\alpha^p \beta^p_{\text{new}}}{W}, & \text{if } k \text{ is new} \end{cases}$$

Parallel-HDP & Async-HDP (Cont.)

- New topics might be introduced while infering
- Bipartite matching across two topic sets might be required
- They don’t do it but it works!
  - They just combine the sets by topic ids

Experiments

- Async-LDA
- Prediction power

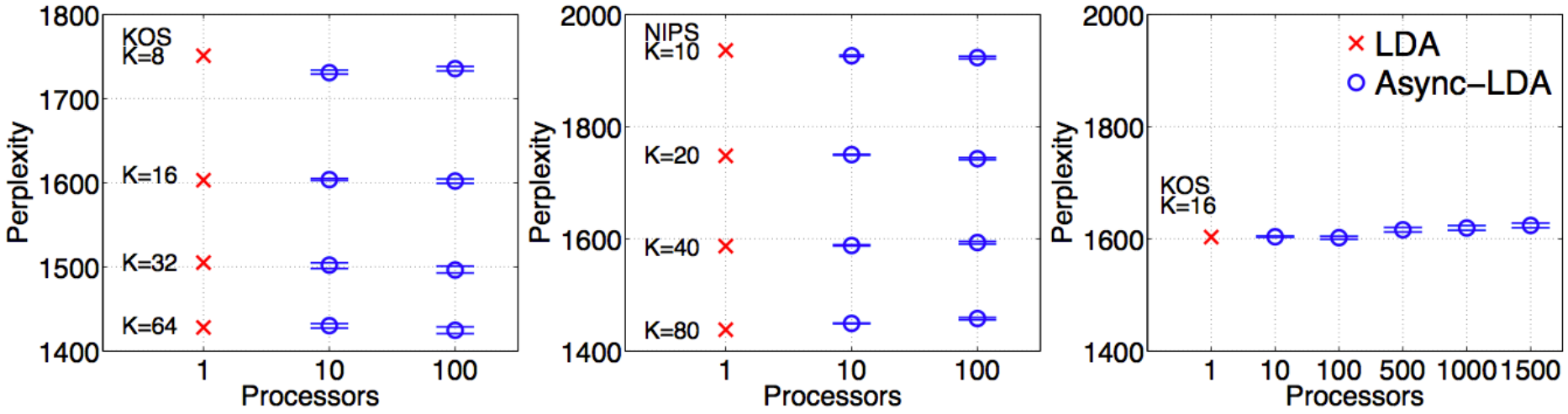


Figure 2: (a) Left: Async-LDA perplexities on KOS. (b) Middle: Async-LDA perplexities on NIPS. (c) Right: Async-LDA perplexities on KOS with many procs. Cache=5 when P$\geq$100. 3000 iterations run when P$\geq$500.

Experiments (Cont.)

- Async-LDA
- Speed analysis

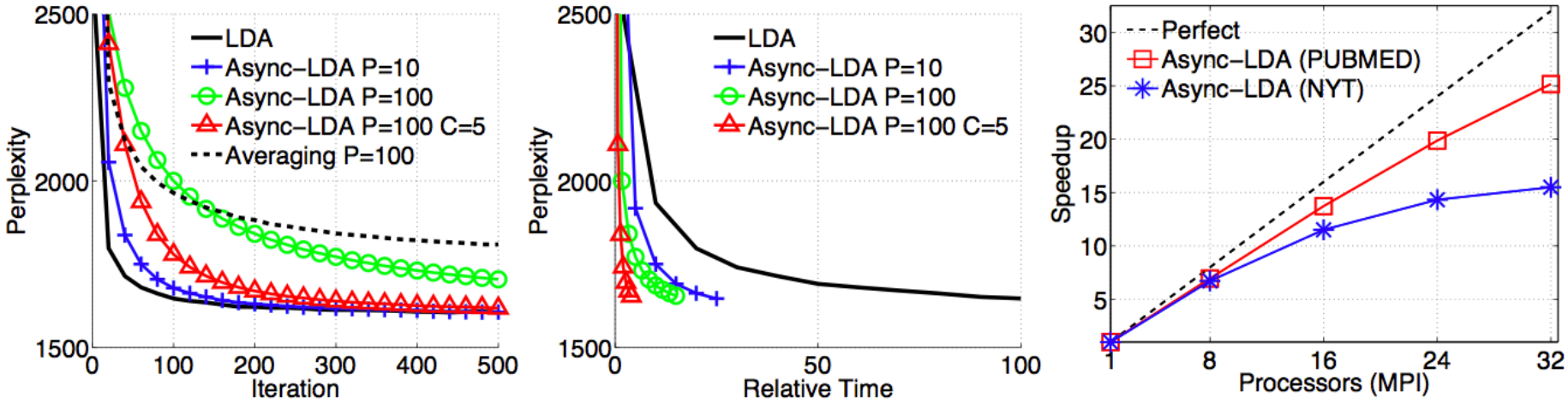


Figure 3: (a) Left: Convergence plot for Async-LDA on KOS, K=16. (b) Middle: Same plot with x-axis as relative time. (c) Right: Speedup results for NYT and PUBMED on a cluster, using Message Passing Interface.

Experiments (Cont.)

- Parallel-HDP & Async-HDP

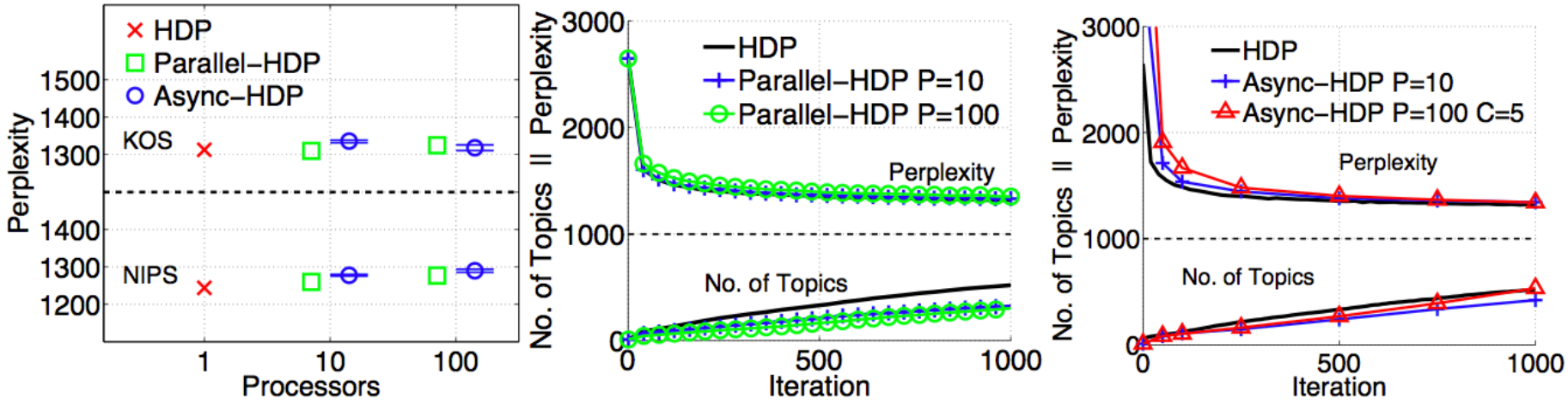


Figure 4: (a) Left: Perplexities for Parallel-HDP and Async-HDP. Cache=5 used for Async-HDP P=100. (b) Middle: Convergence plot for Parallel-HDP on KOS. (c) Right: Convergence plot for Async-HDP on KOS.

Experiments (Cont.)

- Parallel-HDP & Async-HDP
  - The number of topics does stabilize after thousands of iterations
  - Topics are generated at a slightly faster rate for Async-HDP than for Parallel-HDP
    - *Because Async-HDP takes a less aggressive approach on pruning small topics since processors need to be careful when pruning topics locally*.

Experiments (Cont.)

- 3 Different experiments here

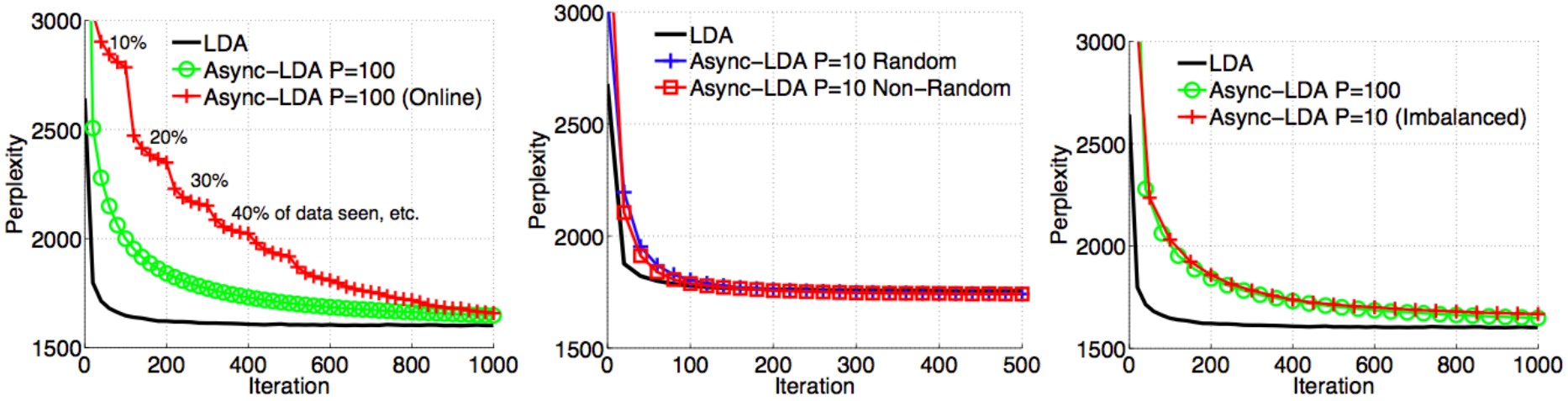


Figure 5: (a) Left: Online learning for Async-LDA (K=16) on KOS. (b) Middle: Comparing random vs. non-random distribution of documents for Async-LDA on NIPS, K=20. (c) Right: Async-LDA on KOS, K=16, where processors have varying amounts of data. In all 3 cases, Async-LDA converges to a good solution.

Conclusions

- Go Async-LDA!