Incremental Variational Inference applied to Latent Dirichlet Allocation

Cédric Archambeau
Amazon
cedrica@amazon.de

Beyza Ermiş
Bogazici University
beyza.ermis@boun.edu.tr

Abstract

We introduce incremental variational inference, which generalizes incremental EM and provides an alternative to stochastic variational inference. It also naturally extends to the distributed setting. We apply incremental variational inference to LDA and show that there is a benefit to do multiple passes over the data in the large-scale setting. Incremental inference does not require to set a learning rate, converges faster to a local optimum of the variational bound and enjoys the attractive property of monotonically increasing it like its batch counterpart.

1 Introduction

Approximate Bayesian inference has become mainstream in machine learning [1, 2] and enjoyed regained interest in the statistics [3, 4]. It constitutes an appealing alternative to Markov Chain Monte Carlo when one is interested in probabilistic data modelling. Approximate inference techniques are pragmatic, postulating an approximate model family and trying to find the best model within this family by optimizing a surrogate objective [5]. They are also practical, as the code implementing these inference algorithms is relatively easy to debug. For example, variational inference monotonically increases the variational objective, providing a sanity check for correctness and convergence.

Stochastic variational inference [6] was a first attempt to scale up approximate inference to massive data sets. It relies on stochastic optimization [7] and processes the data sequentially. Two drawbacks of stochastic variational inference over batch variational inference are that it requires to adjust a learning rate and does not share the attractive property of monotonically increasing the bound at training time. Importantly, it cannot easily be mapped on distributed architectures, such as multiprocessor and grid-computing hardware. Recent attempts in this direction include [8, 9, 10].

To address these shortcomings, we introduce incremental variational inference, which generalizes incremental EM proposed by [11]. Like stochastic variational inference, incremental variational inference processes the data sequentially. However, it does not require to adjust the learning rate. By maintaining a set of local statistics, it also preserves the property of monotonically increasing the variational objective at each iteration. We propose an extension of incremental variational inference that can be executed in a distributed environment without impacting the predictive performance.

In this note, we focus on Latent Dirichlet Allocation (LDA) [12, 13], a popular generative model for documents. However, the approximate inference scheme that we introduce is general and it is applicable to any latent variable model with a set of local and global variables.

Let us denote word n in document d by $x_{nd} \in \{1, \dots, V\}$ and its topic assignment by $z_{nd} \in \{1, \dots, K\}$. The generative model of LDA is defined as follows:

$$z_{nd} \mid \boldsymbol{\theta}_d \sim \text{Categorical}(\boldsymbol{\theta}_d), \qquad x_{nd} \mid z_{nd}, \{\boldsymbol{\phi}_k\}_{k=1}^K \sim \text{Categorical}(\boldsymbol{\phi}_{z_{nd}}),$$

where $\boldsymbol{\theta}_d \sim \text{Dirichlet}(\alpha_0 \mathbf{1}_K)$ and $\boldsymbol{\phi}_k \sim \text{Dirichlet}(\beta_0 \mathbf{1}_V)$. The parameters α_0 and β_0 are nonnegative reals.

2 Variational Inference for LDA

Variational inference maximizes a lower bound to the log marginal likelihood of the data by approximating the true posterior by postulating a simpler distribution, which is parametrized by a set of free parameters. In the case of LDA, the variational bound is given by

$$\begin{aligned}\ln p(X) &\geqslant \langle \ln p(X, Z, \Theta, \Phi)\rangle + \mathrm{H}[q(Z, \Theta, \Phi)] \\ &= \ln p(X) - \mathrm{KL}[q(Z, \Theta, \Phi) \| p(Z, \Theta, \Phi | X)],\end{aligned}$$

where $X = \{x_{nd}\}_{n,d}$, $Z = \{z_{nd}\}_{n,d}$, $\Theta = \{\boldsymbol{\theta}_d\}_d$ and $\Phi = \{\boldsymbol{\phi}_k\}_k$. The notation $\langle\cdot\rangle$ denotes an expectation wrt $q(Z, \Theta, \Phi)$, $\mathrm{H}[p]$ is the differential entropy and $\mathrm{KL}[q\|p]$ is the Kullback-Leibler divergence wrt q. Maximizing this bound is equivalent to minimizing the Kullback-Leibler divergence between the true posterior $p(Z, \Theta, \Phi|X)$ and the approximate posterior $q(Z, \Theta, \Phi)$. In general, this minimization problem is still problematic, unless we further restrict the form of $q(Z, \Theta, \Phi)$.

Mean field variational inference (MVI) assumes the latent variables and the parameters are independent when conditioning on the data, that is, $q(Z, \Theta, \Phi) = \prod_{n,d} q(z_{nd}) \times \prod_d q(\boldsymbol{\theta}_d) \times \prod_k q(\boldsymbol{\phi}_k)$. In this case the lower bound is maximized when the factors are defined as follows [13]:

$$\begin{aligned} q(z_{nd}) &= \text{Categorical}(\boldsymbol{\pi}_{nd}), & \pi_{knd} &\propto e^{\langle \ln \theta_{kd}\rangle + \langle \ln \phi_{x_{nd}k}\rangle}, \\ q(\boldsymbol{\theta}_d) &= \text{Dirichlet}(\boldsymbol{\alpha}_d), & \alpha_{kd} &= \alpha_0 + \langle m_{kd}\rangle, \\ q(\boldsymbol{\phi}_k) &= \text{Dirichlet}(\boldsymbol{\beta}_k), & \beta_{vk} &= \beta_0 + \langle m_{vk}\rangle, \end{aligned} \tag{1}$$

where m_{kd} is the (unobserved) number of times topic k appeared in document d and m_{vk} the (unobserved) number of times word token v was assigned to topic k in the corpus. Hence, the special quantities $\langle m_{kd}\rangle$ and $\langle m_{vk}\rangle$ are expected counts under the variational approximation. They are respectively given by $\sum_n \pi_{knd}$ and $\sum_{n,d} \delta_v(x_{nd})\pi_{knd}$. The function $\delta_v(\cdot)$ is Dirac's delta at v.

MVI is a coordinate ascent method that converges to a local maximum of the variational bound [15]. Cycling through the updates in (1) ensures a monotonic increase of this bound. MVI is a batch inference approach: every update of the variational parameter β_{vk} requires updating all word-specific proportions $\boldsymbol{\pi}_{nd}$ beforehand, which is costly when the corpus is large. Stochastic variational inference (SVI) was recently proposed in the context and applied to LDA [16, 6]. The goal was to speed up inference and to scale up LDA to very large document collections.

SVI optimizes the lower bound by stochastic approximation [7]. It maintains a set of local and global parameters, which characterize the variational posteriors. Local variables are the indicator variables Z and the document-topic proportions Θ, which depend respectively on the local parameters $\{\boldsymbol{\pi}_{nd}\}_{n,d}$ and $\{\alpha_d\}_d$. The global variables are topic-word proportions Φ, which depend on the global parameters $\{\boldsymbol{\beta}_k\}_k$.

This leads to the following updates (document d is being picked at random) [6]:

$$\boldsymbol{\beta}_k^{(t)} = (1-\rho_t)\boldsymbol{\beta}_k^{(t-1)} + \rho_t \hat{\boldsymbol{\beta}}_k, \qquad \hat{\beta}_{vk} = \beta_0 + D {\textstyle\sum_{n=1}^{N_d}} \delta_v(x_{nd})\pi_{knd}, \tag{2}$$

where $\sum_t \rho_t = \infty$ and $\sum_t \rho_t^2 < \infty$. Throughout this work, we will use the learning rate $\rho_t = (t+\tau)^{-\kappa}$, where $\kappa \in (0.5, 1]$ and $\tau \geqslant 0$.

Intuitively, the second term on the right hand side of (2) is a noisy, but unbiased estimate of the expected number of counts appearing in (1), namely $\langle m_{vk}\rangle$. The variational parameters associated to the local variables (that is, $\boldsymbol{\pi}_{nd}$ and $\boldsymbol{\alpha}_d$) can be computed as in MVI. Typically, mini-batches are used to stabilize the gradients.

3 Incremental Variational Inference for LDA

Incremental variational inference (IVI) computes updates in a similar fashion as incremental EM [11]. Each iteration performs a partial variational E-step before performing a variational M-step. This amounts to maintaining a set of global statistics associated to the global variables, which are updated incrementally in the variational E-step by first subtracting the old statistics associated to a data point (or a mini-batch) and adding back the corresponding new one. The updated global statistics are then used in the variational M-step. Hence, IVI maintains an estimate of the global

Algorithm 1 Incremental Variational Inference (IVI)

```
Initialize $\beta_{vk}^{(0)}$ randomly; set $\alpha_{kd}$ = $\alpha_0$.
for $t = 1, 2, \cdots$ do
    Sample a document $d$ uniformly
    repeat
        $\pi_{knd}^{(t)} \propto e^{\langle \ln \theta_{kd} \rangle + \langle \ln \phi_{x_{nd}k} \rangle}$
        $\alpha_{kd} = \alpha_0 + \sum_{n=1}^{N_d} \pi_{knd}$
    until $\alpha_{kd}$ and $\pi_{knd}$ converge.
    $\beta_{vk} = \beta_0 + \langle m_{vk} \rangle + \sum_{n=1}^{N_d} \delta_v(x_{nd}) \big( \pi_{knd}^{(t)} - \pi_{knd}^{(t-1)} \big)$
end for
```

statistics based on the full data set, while SVI only considers a mini-batch-based estimate. IVI leads to the following incremental update for LDA:

$$\beta_{vk} = \beta_0 + \langle m_{vk} \rangle + \textstyle\sum_{n=1}^{N_d} \delta_v(x_{nd}) \big(\pi_{knd}^{(t)} - \pi_{knd}^{(t-1)} \big). \tag{3}$$

The updates for $\boldsymbol{\pi}_{nd}$ and $\boldsymbol{\alpha}_d$ are the same as in MVI as they are associated to the local variables. IVI does not require to have seen all the data points to make progress, but it ensures a monotonic increase of the bound. The price we have to pay is the storage of the previous aggregated proportions over the mini-batch. This can be costly when the number of topics K is large as the additional memory requirements scale as a constant factor times the number of words in the corpus. IVI for LDA is summarized in Algorithm 1.

Distributed Variational Inference for LDA The benefit of SVI and IVI in the context of large document collections is that they make faster progress by processing documents sequentially. To further speed up inference in IVI, we introduce synchronous distributed incremental variational inference (D-IVI), which infers topics comparable to those inferred by IVI, but with a significant reduction in computation time by handling multiple mini-batches in parallel. We consider one master and N workers, each of which holds $1/N$ of the documents in the corpus. The workers hold the local parameters $\{\boldsymbol{\pi}_{nd}\}_{n,d}$ and $\{\boldsymbol{\alpha}_d\}_d$. They independently carry-out partial variational E-step based on local copies of the global parameter $\{\boldsymbol{\beta}_k\}_k$ and send the statistics corrections associated to their mini-batch to the master, that is, $\sum_{n=1}^{N_d} \delta_v(x_{nd}) \big(\pi_{knd}^{(t)} - \pi_{knd}^{(t-1)} \big)$. The master collects the changes and updates the global parameters according to (3). He then sends back the updated estimate to the workers. In practice, there is a trade-off between the convergence speed and the amount of communication: smaller mini-batches speed up progress, but increase the communication overhead.

4 Experiments and Results

We conduct two types of experiments. First, we benchmark IVI against MVI and SVI. Second, we report speed-ups obtained with our distributed algorithm (D-IVI). We consider three benchmark corpora: Wikipedia articles, the scientific abstracts from Arxiv repository [24] and Amazon customer reviews (CR). The characteristics of the datasets are reported in the Suppl. Mat. We estimate the predictive probability over the vocabulary [13]. We wish to achieve high average per-word likelihood on held-out test documents. Under this metric, a higher score is better, as a better model will assign a higher probability to the held-out words.

IVI Prediction Results We compare the performance of IVI and MVI at the point where MVI converges to a solution. IVI yields the same result after processing half (Arxiv) to one tenth (Wikipedia) of the number of documents processed by MVI. Besides, we observe that IVI gives consistently better predictive performance than MVI and SVI when both of them converges to a solution. Finally, it can be observed that the bound monotonically increases unlike the SVI bound.

D-IVI Convergence and Speed-up Results Computations were done on N machines for $N = \{1, 5, 10, 20, 50\}$. The results are averaged over 5 runs with random initialization. In Figure 2 , we report wall clock time, speed-up and the log predictive probability. When N=5, we observe approximately a 4.8 times speed-up for the two corpora compared to single machine execution.

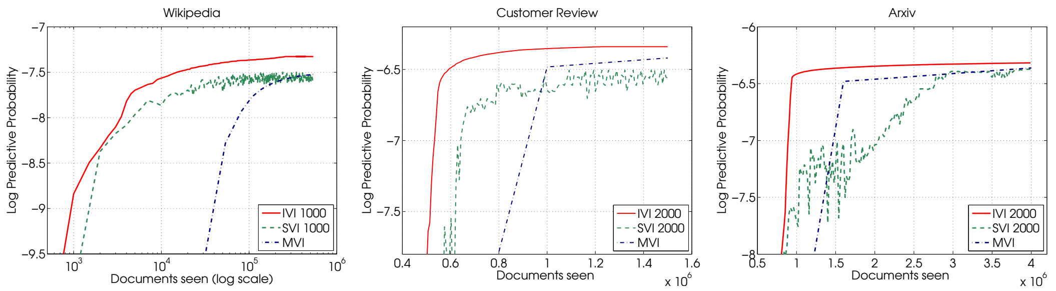

Figure 1: Per-word predictive probabilitiy for LDA as a function of the number of processed documents. We run experiments with the Wikipedia, Arxiv and Customer Review data sets. IVI converges faster and to a higher value on all datasets. (K=100, $\alpha_0 = 0.5$ and $\beta_0 = 0.05$)

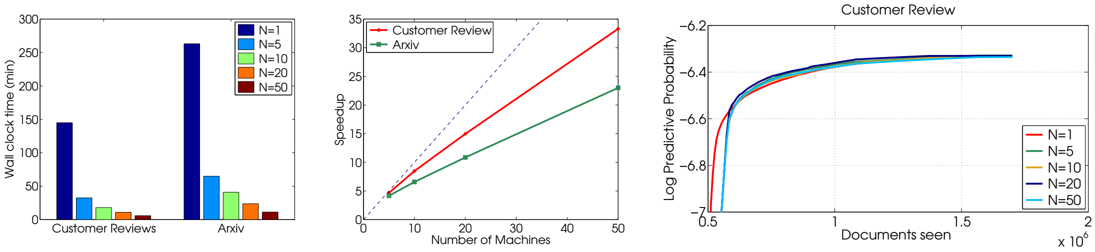

Figure 2: *Left:* Wall-clock time (in minutes) comparisons for D-IVI for different number of machines on Arxiv and Customer Review. *Middle:* Speed-up results of D-IVI for varying number of machines with respect to single machine. *Right:* Log predictive probability comparisons for IVI for different number of machines on Customer Review data set.

When N=50, we note a speed-up of approximately 33 and 22.5 respectively for CR and Arxiv. It can be observed that the rate of convergence slows down when the number of processors increases, since more iterations are needed to propagate the newest information to all the processors. However, it is important to note that one iteration in real time of D-IVI is up to number of machines times faster than one iteration of IVI, so D-IVI converges much more quickly than IVI. Figure 2–*Right* shows D-IVI performance against number of documents seen so far, demonstrating that the quality of the model learned is essentially the same for each N values. Finally, we investigated the effect of the mini-batch size (see Table 2 in Suppl. Mat. for full detail). When the mini-batch size is small, each update is more noisy and more iterations are needed to converge. In practice, the communication overhead can be mitigated by adjusting the mini-batch size.

5 Discussion

The price we have to pay when using D-IVI is that we have to store the proportions $\{\boldsymbol{\pi}_{nd}\}$ aggregated over the mini-batches. The additional memory requirements scale as K times the number of words in the corpus. In practice, the optimal number of topics is relatively small (in the hundreds), which means that the storage overhead is acceptable when we can afford to store the raw data.[1] Moreover, one should keep in mind that for other models than LDA, like for example mixture models, the additional storage requirements might be significantly lower than the raw data, which can be high-dimensional and dense.

[1] We refer the interested reader to Table 3 in the Suppl. Mat. which reports the average predictive log likelihood and time (in minutes) for different number of topics (up to 1000) for CR and Arxiv data.

References

[1] Christopher M Bishop et al., *Pattern recognition and machine learning*, springer New York, 2006.

[2] Kevin P Murphy, *Machine learning: a probabilistic perspective*, MIT press, 2012.

[3] B. Wang and D. M. Titterington, "Convergence properties of a general algorithm for calculating variational bayesian estimates for a normal mixture model," *Bayesian Analysis*, vol. 1, pp. 625–650, 2006.

[4] A. Armagan and D.B. Dunson, "Sparse variational analysis of large longitudinal data sets," *Statistics and Probability Letters*, vol. 81, pp. 1056–1062, 2011.

[5] M. Wainwright and M. Jordan, "Graphical models, exponential families, and variational inference," *Foundations and Trends in Machine Learning*, vol. 1, no. 1–2, pp. 1–305, 2008.

[6] Matthew Hoffman, David Blei, Chong Wang, and John Paisley, "Stochastic variational inference," *Journal of Machine Learning Research*, vol. 14, no. 1, pp. 1303–1347, 2013.

[7] Herbert Robbins and Sutton Monro, "A stochastic approximation method," *Aannals of mathematical statistics*, pp. 400–407, 1951.

[8] Alexander Smola and Shravan Narayanamurthy, "An architecture for parallel topic models," *Proceedings of the VLDB Endowment*, vol. 3, no. 1-2, pp. 703–710, 2010.

[9] David Newman, Arthur Asuncion, Padhraic Smyth, and Max Welling, "Distributed algorithms for topic models," *Journal of Machine Learning Research*, vol. 10, pp. 1801–1828, 2009.

[10] Arthur Asuncion, Padhraic Smyth, and Max Welling, "Asynchronous distributed learning of topic models," in *Advances in Neural Information Processing Systems*, 2009, pp. 81–88.

[11] Radford Neal and Geoffrey Hinton, "A view of the EM algorithm that justifies incremental, sparse, and other variants," in *Learning in graphical models*, pp. 355–368. Springer, 1998.

[12] Thomas L Griffiths and Mark Steyvers, "Finding scientific topics," *Proceedings of the National Academy of Sciences*, vol. 101, no. suppl 1, pp. 5228–5235, 2004.

[13] David Blei, Andrew Ng, and Michael Jordan, "Latent Dirichlet allocation," *Journal of machine Learning research*, vol. 3, pp. 993–1022, 2003.

[14] Thomas Minka and John Lafferty, "Expectation-propagation for the generative aspect model," in *Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 2002, pp. 352–359.

[15] Matthew Beal, *Variational algorithms for approximate Bayesian inference*, Ph.D. thesis, University of London, 2003.

[16] Matthew Hoffman, Francis Bach, and David Blei, "Online learning for latent Dirichlet allocation," in *Advances in Neural Information Processing Systems*, 2010, pp. 856–864.

[17] Chong Wang, Xi Chen, Alex Smola, and Eric Xing, "Variance reduction for stochastic gradient optimization," in *Advances in Neural Information Processing Systems*, 2013, pp. 181–189.

[18] John Paisley, David Blei, and Michael Jordan, "Variational bayesian inference with stochastic search," *arXiv preprint arXiv:1206.6430*, 2012.

[19] Rajesh Ranganath, Chong Wang, Blei David, and Eric Xing, "An adaptive learning rate for stochastic variational inference," in *Proceedings of the 30th International Conference on Machine Learning*, 2013, pp. 298–306.

[20] Nicolas Le Roux, Mark Schmidt, and Francis Bach, "A stochastic gradient method with an exponential convergence _rate for finite training sets," in *Advances in Neural Information Processing Systems*, 2012, pp. 2663–2671.

[21] Yee Whye Teh, David Newman, and Max Welling, "A collapsed variational Bayesian inference algorithm for latent Dirichlet allocation," in *Advances in neural information processing systems*, 2006, pp. 1353–1360.

[22] James Foulds, Levi Boyles, Christopher DuBois, Padhraic Smyth, and Max Welling, "Stochastic collapsed variational Bayesian inference for latent Dirichlet allocation," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2013, pp. 446–454.

[23] Prem Gopalan, Jake Hofman, and David Blei, "Scalable recommendation with poisson factorization," *arXiv preprint arXiv:1311.1704*, 2013.

[24] Stephan Mandt and David Blei, "Smoothed gradients for stochastic variational inference," in *Advances in Neural Information Processing Systems*, 2014, pp. 2438–2446.

[25] Michael C Hughes and Erik Sudderth, "Memoized online variational inference for dirichlet process mixture models," in *Advances in Neural Information Processing Systems*, 2013, pp. 1133–1141.

[26] Ramesh Nallapati, William Cohen, and John Lafferty, "Parallelized variational EM for latent Dirichlet allocation: An experimental evaluation of speed and scalability," in *Data Mining Workshops, 2007. ICDM Workshops 2007. Seventh IEEE International Conference on*. IEEE, 2007, pp. 349–354.

[27] Jason Wolfe, Aria Haghighi, and Dan Klein, "Fully distributed EM for very large datasets," in *Proceedings of the 25th international conference on Machine learning*. ACM, 2008, pp. 1184–1191.

[28] Ke Zhai, Jordan Boyd-Graber, Nima Asadi, and Mohamad Alkhouja, "Mr. lda: A flexible large scale topic modeling package using variational inference in MapReduce," in *Proceedings of the 21st international conference on World Wide Web*. ACM, 2012, pp. 879–888.

[29] Bo Thiesson, Christopher Meek, and David Heckerman, "Accelerating EM for large databases," *Machine Learning*, vol. 45, no. 3, pp. 279–299, 2001.

Supplementary Material

Table 1: Characteristics of data sets used in experiments.

	Wikipedia	Arxiv	CustomerReview
Number of documents in training set	39565	782385	452944
Number of documents in test set	10000	100000	100000
Average number of words per document	260	116	151
Number of words in vocabulary	42419	141927	120043

Table 2: Log-prediction-probability (LPP) and runtime (in terms of minutes per pass) of the IVI for different number of mini-batch sizes and number of machines (number of topics = 50).

Datasets	Customer Review						Arxiv					
Mini-batch Size		Number of Machines						Number of Machines				
		1	5	10	20	50		1	5	10	20	50
1000	LPP	-6.33	-6.33	-6.33	-6.33	-6.33	LPP	-6.42	-6.42	-6.42	-6.42	-6.42
	Time	148	36	21	13	6.7	Time	285	72.3	46.2	24.7	11.8
2000	LPP	-6.33	-6.33	-6.33	-6.33	-6.33	LPP	-6.42	-6.42	-6.42	-6.42	-6.42
	Time	145	32.5	18	11	5.9	Time	263	65	41	23.7	11.3
5000	LPP	-6.31	-6.31	-6.31	-6.31	-6.31	LPP	-6.40	-6.40	-6.40	-6.40	-6.40
	Time	140	30	16.5	9.35	4.2	Time	250	60	38	23	11
MVI (full batch)	LPP	-6.41	-	-	-	-	LPP	-6.48	-	-	-	-
	Time	127					Time	240				

Table 3: Log-prediction-probability (LPP) and runtime (in terms of minutes per iteration) of the IVI for different number of topics and number of processors (mini-batch size = 2000).

Datasets	Customer Review						Arxiv					
Number of Topics		Number of Machines						Number of Machines				
		1	5	10	20	50		1	5	10	20	50
25	LPP	-6.46	-6.46	-6.46	-6.46	-6.46	LPP	-6.57	-6.57	-6.57	-6.57	-6.57
	Time	138	31.6	16.7	10.8	5.3	Time	224	61	37	21.6	10.1
50	LPP	-6.33	-6.33	-6.33	-6.33	-6.33	LPP	-6.42	-6.42	-6.42	-6.42	-6.42
	Time	145	32.5	18	11	5.9	Time	263	65	41	23.7	11.3
100	LPP	-6.29	-6.29	-6.29	-6.29	-6.29	LPP	-6.33	-6.33	-6.33	-6.33	-6.33
	Time	148	33.2	18.6	11.5	6.1	Time	268	68	43	24.5	11.7
200	LPP	-6.49	-6.49	-6.49	-6.49	-6.49	LPP	-6.46	-6.46	-6.46	-6.46	-6.46
	Time	159	35.4	19.5	11.9	6.3	Time	297	73.7	46.2	26.8	12.8
1000	LPP	-6.84	-6.84	-6.84	-6.84	-6.84	LPP	-6.97	-6.97	-6.97	-6.97	-6.97
	Time	167	37.3	21.2	12.4	6.7	Time	306	78	49	28.2	13.4