FastEx: Hash Clustering with Exponential Families

Amr Ahmed, Sujith Ravi, Shravan M. Narayanamurthy, Alexander J. Smola

Presented by: Jianbo Yang

Duke University

January 25, 2013
Highlights

- **Large-scale clustering problem**
  The clustering problem are considered. # clusters = 1,000, # dimension = 5.6M, # instances = 710K

- **Probabilistic method**
  Let $\mathcal{X}$ be the domain of $m$ observations $\mathbf{X} = \{x_1, \ldots, x_m\}$. All observations are drawn from some distribution $p$, and the task is to estimate $p$.

- **Mixtures of exponential families model**
  The mixture of exponential models are adopted. That means the derivation is applicable to mixture of Multinomial, Gaussian, Poisson, Dirichlet distributions.

- **A new proposal distribution**
  Binary hashing is used to compute a fast proposal distribution. It has the following advantages:
  - 37.37X speedup over the baseline in the largest tested data
  - support of inference parallelization
Exponential Families

- In exponential families distributions over random variables are given by

\[ p(x; \theta) = \exp(\langle \phi(x), \theta \rangle - g(\theta)) \]  

(1)

where \( \phi(\cdot) : \mathcal{X} \to \mathcal{F} \) is a map from \( \mathcal{X} \) to the vector space \( \mathcal{F} \) of sufficient statistics (assume \( \mathcal{F} \) is a Hilbert space), \( \theta \in \mathcal{F} \) is the natural parameters, and \( g(\theta) \) ensures \( p(x; \theta) \) is properly normalized via

\[ g(\theta) := \log \int_{\mathcal{X}} \exp(\langle \phi(x), \theta \rangle)dx \]

- The mean parameter associated with (1) and the maximum likelihood estimate given \( \mathbf{X} \) are connected via \( \mu(\theta) = \mu(\mathbf{X}) \) where

\[ \mu(\theta) := \mathbf{E}_{x \sim p(x; \theta)}[\phi(x)] = \partial_\theta g(\theta) \quad \text{and} \quad \mu(\mathbf{X}) := \frac{1}{m} \sum_{i=1}^{m} \phi(x) \]
Conjugate priors $p(\theta)$ is also a member of the exponential family:

$$p(\theta|m_0, m_0\mu_0) = \exp(\langle m_0\mu_0, \theta \rangle - m_0 g(\theta) - h(m_0, m_0\mu_0))$$  \hspace{1cm} (2)

where the **sufficient statistics** is $\phi(\theta) = (-g(\theta), \theta)$, the **natural parameters** are $(m_0, m_0\mu_0)$ and $h(m_0, m_0\mu_0)$ is a log-partition function in the parameters of the conjugate prior.

Combining (1) and (2) and after normalization, yields the posterior

$$p(\theta|X) \propto p(X|\theta)p(\theta|m_0, m_0\mu_0)$$

$$= p(\theta|m_0 + m, m_0\mu_0 + m\mu[X])$$
Mixture Models

- Dirichlet-Multinomial model with $k$ components
  - Draw cluster membership $y_i \sim p(y|\theta)$, where $y \in \{1, \ldots, k\}$, $p(y|\theta)$ is a multinomial distribution.
  - Draw observations from the cluster distribution $x_i \sim p(x|\theta_{y_i})$

There are two exponential families components here. First, multinomial distribution $p(y|\theta)$ with prior $\theta \sim \text{Dir}(\alpha)$ for $k$ clusters. Second, cluster distribution $p(x|\theta_{y_i})$ with its conjugate prior.

- For large scale dataset, collapsed Gibbs sampling is preferable, as the parameters $\theta$ in $p(y|\theta)$ are integrated out.
  - For a given $x_i$ draw $y_i \sim p(y_i|X, Y^{-i}) \propto p(y_i|Y^{-i}) p(x_i|y_i, X^{-i}, Y^{-i})$
  - Update the sufficient statistics for the two changed clusters
Taylor Approximation for Collapsed Inference

- Data likelihood

\[ p(\mathbf{X}|m_0, m_0 \mu_0) = \int p(\mathbf{X}|\theta)p(\theta|m_0, m_0 \mu_0)d\theta = \exp(h(m_0 + m, m_0 \mu_0 + m \mu[\mathbf{X}]) - h(m_0, m_0 \mu_0)) \] (3)

- By Bayes rule, the posterior is

\[ p(x|\mathbf{X}, m_0, \mu_0) \propto p(\mathbf{X} \cup \{x\}|m_0, m_0 \mu_0) \propto \exp(h(m_0 + m + 1, m_0 \mu_0 + m \mu[\mathbf{X}] + \phi(x))) \] (4)

Normalization of \( h \) is nontrivial to compute or even intractable.
Taylor approximation

\[ \partial_{m_0,m_0\mu_0} h(m_0, \mu_0 m_0) = E_{\theta \sim p(\theta|m_0,m_0\mu_0)}[-g(\theta), \theta] = (-\gamma^*, \theta^*) \]

hence

\[ h(m_0 + 1, m_0\mu_0 + \phi(x)) \approx h(m_0, m_0\mu_0) + \langle \theta^*, \phi(x) \rangle - \gamma^* \]

(5)

where \( \gamma^* \) is not needed for the inference.

Applying (5) to (4)

\[ p(x|X, m_0, m_0\mu_0) \approx \exp(\langle \phi(x), \theta^* \rangle - g(\theta^*)) \]

(6)

The advantage of (6) is that we don’t need to compute \( h \) directly but instead to estimate \( \theta^* \). According to (5), \( \theta^* = E_{\theta \sim p(\theta|X)}[\theta] \)

Accelerate \( \langle \phi(x), \theta^*_y \rangle \) in (6), as (6) is evaluated \( k \) times at each Gibbs sampler.
Accelerate $\langle \phi(x), \theta^*_y \rangle = \|\phi(x)\| \|\theta^*_y\| \cos \angle(\phi(x), \theta^*_y)$

Remember $\# \phi(x) > 10^5$, $\# \theta^*_y = 10^3$, and $\phi(x) \in \mathbb{R}^n$, $\theta^*_y \in \mathbb{R}^n$ and $n > 10^6$.

- **Theorem 1** For $u, v \in \mathbb{R}^n$ and vectors $w$ drawn from a spherically symmetric distribution on $\mathbb{R}^n$ the following relation between signs of inner products and the angle $\angle(u, v)$ between vectors hold:
  
  $\angle(u, v) = \pi \Pr\{\text{sgn}[\langle u, w \rangle] \neq \text{sgn}[\langle v, w \rangle]\}$.

  *Whenever $w$ falls into the angle between the unit vectors in the directions $u$ and $v$, we have the opposite sign.*

- **Definition 1** We denote by $f^l(v) \in \{0, 1\}^l$ a binary hash of $v$ and by $z^l(u, v)$ an estimate of the probability of matching signs, obtained as follows

  $[f^l(v)]_i := \text{sgn}[\langle v, w_i \rangle]$ where $w_i \sim U_m$ fixed and $z^l(u, v) := \frac{1}{l} \|f(u) - f(v)\|_1$

  where $z^l(u, v)$ is Hamming distance of $f(u)$ and $f(v)$ and $l$ is the number of bits.

  *Note modern CPU can compute Hamming distance super fast, and therefore cost of inner product is significantly reduced, roughly from $\ell \times 10^8$ to $l \times \max(10^5, \ell \times 10^3)$ where $\ell$ is the Gibbs sampling iterations.*
Unnormalized log-likelihood of an instance being assigned to a cluster via

\[ s^l(x, y) = \|\theta_y\| \|\phi(x)\| \cos \pi z^l(\phi(x), \theta_y) - g(\theta_y) - \log n_y \]  \hfill (7)

Error bound of \( s^l(x, y) \):

**Theorem 2** Given \( k \) mixture components and let \( l \) the number of bits used for hashing. Then the unnormalized cluster log-likelihood is bounded with probability at least \( 1 - \delta \) by

\[
\bar{s}^l(x, y) = \|\theta_y\| \|\phi(y)\| \cos\left[\pi \max(0, z^l(\phi(x), \theta_y) - \sqrt{\log k/\delta}/2l)\right] - g(\theta_y) - \log n_y
\]  \hfill (8)
An alternative is to employ the approximate upper bound as a proposal distribution in a Metropolis Hastings (MH) framework.

- Denote by $q$ the proposal distribution constructed from the bound on the log-likelihood after normalization.
- For a given $x_i$, sample $y_i^{\text{new}} \sim q(\cdot)$ and then accept the proposal using (8) with probability $r$ where

$$q(y) \propto \exp(\bar{s}^l(x, y)) \quad \text{and} \quad r = \frac{q(y_i^{\text{old}})p(y_i^{\text{new}})p(x_i|X^{-i}, y_i^{\text{new}}, m_0, \mu_0)}{q(y_i^{\text{new}})p(y_i^{\text{old}})p(x_i|X^{-i}, y_i^{\text{old}}, m_0, \mu_0)}$$

- Note that standard collapsed Gibbs sampler $p(x|X, \mu_0, m_0)$ would be computed for all $k$ candidate clusters. However, in this framework, we only need to compute 2 clusters: the proposed and old clusters.
Experiments

Documents clustering data
- 100 clusters, 292K articles, 2.5M unique words vocabulary
- 1000 clusters, 710K articles, 5.6M unique words vocabulary

Methods
- **Baseline**: Dirichlet Multinomial mixture model. It uses an uncollapsed likelihood and alternates between sampling clusters and drawing from the Dirichlet distribution of the posterior.
- **FastEx**: Unless stated otherwise, we use $l = 32$ bit to represent a document and cluster.

Evaluation criterion
- Suppose we have two clusters $C_1$ (predicted) and $C_2$ (ground truth), and then variation of information (VI) is

\[ \text{VI}(C_1, C_2) = H(C_1) + H(C_2) - 2I(C_1, C_2) \]

The lower value for VI the better quality of clustering.
Convergence Results

Figure: Left: Convergence of both a baseline and FastEx (time axis is in log-scale). Right: the effect of the hash size on the performance.
Timing Results

<table>
<thead>
<tr>
<th>Clusters $k$</th>
<th>Bitsize $l$</th>
<th>8</th>
<th>16</th>
<th>32</th>
<th>64</th>
<th>128</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposal</td>
<td></td>
<td>2.34</td>
<td>2.34</td>
<td>2.34</td>
<td>2.56</td>
<td>2.90</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>69.52</td>
<td>69.52</td>
<td>78.77</td>
<td>81.16</td>
<td>82.19</td>
</tr>
<tr>
<td>Proposal</td>
<td>1000</td>
<td>18.80</td>
<td>18.80</td>
<td>18.80</td>
<td>21.42</td>
<td>29.12</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>103.91</td>
<td>103.91</td>
<td>103.91</td>
<td>108.98</td>
<td>114.61</td>
</tr>
</tbody>
</table>

Figure: Average time (microseconds) spent per document for FastEx. **Proposal**: compute the proposal distribution. **Total**: proposal time + time to compute acceptance ratio + time to update the clusters sufficient statistics + time of harsh representation.

The total computation time for sampling 10x more clusters only increase slightly, mostly due to the increase in proposal time.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>FastEx Quality (VI)</th>
<th>Baseline Quality (VI)</th>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>$W_{100}$</td>
<td>5.04</td>
<td>5.60</td>
<td>9.25</td>
</tr>
<tr>
<td>$W_{1000}$</td>
<td>14.10</td>
<td>14.00</td>
<td>37.37</td>
</tr>
</tbody>
</table>

Figure: Clustering quality (VI) and absolute speedup achieved by FastEx over the baseline.
Conclusions and extensions

- A new algorithm is developed to perform scalable clustering for exponential families.
- Parallel sampling of $p(y_i | X, Y^{-i})$ is allowed.
- Future work includes the extension of the fast retrieval scheme to hierarchical clustering.