Online Algorithms for Sum-Product Networks with Continuous Variables

Priyank Jaini
Ph.D. Seminar
Mixture Models

Consistent/Robust

- Tensor Decomposition and Spectral Learning
  - Offline
- Bayesian Learning
  - ADF, EP, SGD, oEM
  - Non convergence and local optima

Can be distributed; Practical problems

Bayesian Moment Matching algorithm

SPNs

- Bayesian Moment Matching for continuous SPNs
- Comparison with other deep networks
Streaming Data

Activity Recognition

- Walking
- On Bicycle
- In Vehicle
- Running

Recommendation

Challenge: update model after each observation
Algorithm’s characteristics

Online

\[ X_t \]

\[ \Theta^* \]

Distributed

\[ X_{a:b} \]

\[ X_{a:d} \]

\[ X_{c:d} \]

Consistent

\[ X_{1:t} \]
How can we learn mixture models robustly from streaming data?
Learning Algorithms

• Robust: Tensor Decomposition (Anandkumar et al., 2014), Spectral Learning (Hsu et al., 2012, Parikar and Xing, 2011); offline

• Online:
  • Assumed Density Filtering (Maybeck 1982; Lauritzen 1992; Opper & Winther 1999); not robust
  • Expectation Propagation (Minka 2001); does not converge
  • Stochastic Gradient Descent (Zhang 2004)
  • online Expectation Maximization (Cappe 2012)

SGD and oEM: local optimum and cannot be distributed
Learning Algorithms

• **Exact Bayesian Learning**: Dirichlet Mixtures (Ghosal et al, 1999), Gaussian Mixtures (Lijoi et al, 2005), Non-parametric Problems (Barron et al, 1999), (Freedman, 1999)

In theory; practical problems!
Bayesian Learning

• Uses Bayes’ Theorem

\[ P(\Theta|x) = \frac{P(\Theta)P(x|\Theta)}{P(x)} \]
Bayesian Learning – Mixture models

Data: $x_{1:n}$ where $x_i \sim \sum_{j=1}^{M} w_j \, N(x_i; \mu_j, \Sigma_j)$

$$P_n(\Theta) = \Pr(\Theta|x_{1:n})$$
$$\propto P_{n-1}(\Theta) \Pr(x_n|\Theta)$$
$$\propto P_{n-1}(\Theta) \sum_{j=1}^{M} w_j \, N(x_i; \mu_j, \Sigma_j)$$

Intractable!!!

Solution: Bayesian Moment Matching Algorithm
Method of Moments

- Probability distributions defined by set of parameters
- Parameters can be estimated by a set of moments

\[ X \sim N(X; \mu, \sigma^2) \]
\[ E[X] = \mu \]
\[ E[(X - \mu)^2] = \sigma^2 \]
Make Bayesian Learning Great Again

Bayesian Learning
And
Method of Moments

STRONGER TOGETHER
Gaussian Mixture Models

\[ x_i \sim \sum_{j=1}^{M} w_j \ N(x_i; \mu_j, \Sigma_j) \]

**Parameters:** weights, means and precisions (inverse covariance matrices)
Bayesian Moment Matching for Gaussian Mixture Models

Parameters: weights, means and precisions (inverse covariance matrices)

Prior: $P(w, \mu, \Lambda)$; product of Dirichlets and Normal-Wisharts

Likelihood:

$$P(x ; w, \mu, \Lambda) = \sum_{j=1}^{M} w_j N(x ; \mu_j, \Lambda_j^{-1})$$
Bayesian Moment Matching Algorithm

Exact Bayesian Update

\( x_t \)

Product of Dirichlets and Normal-Wisharts

Projection

Moment Matching

Mixture of Product of Dirichlets and Normal-Wisharts
Sufficient Moments

**Dirichlet**: $Dir (w_1, w_2 \ldots w_M; \alpha_1, \alpha_2 \ldots, \alpha_M)$

\[
E[w_i] = \frac{\alpha_i}{\sum_j \alpha_j}; \quad E[w_i^2] = \frac{\alpha_i(\alpha_i+1)}{(\sum_j \alpha_j)(1+\sum_j \alpha_j)}
\]

**Normal-Wishart**: $NW (\mu, \Lambda ; \mu_0, \kappa, W, v)$

$\Lambda \sim Wi(W, v)$ and $\mu|\Lambda \sim N_d (\mu_0, (\kappa \Lambda)^{-1})$

\[
E[\mu] = \mu_0
\]
\[
E[(\mu - \mu_0)(\mu - \mu_0)^T] = \frac{\kappa+1}{\kappa(v-d-1)} W^{-1}
\]
\[
E[\Lambda] = vW
\]
\[
Var(\Lambda_{ij}) = v(W_{ij}^2 + W_{ii}W_{jj})
\]
Overall Algorithm

- Bayesian Step
  - Compute posterior $P_t(\Theta)$ based on observation $x_t$

- Sufficient Moments
  - Compute set of sufficient moments $S$ for $P_t(\Theta)$

- Moment Matching
  - System of linear equations
  - Linear complexity in the number of components
Bayesian Moment Matching Algorithm
- Uses Bayes’ Theorem + Method of Moments
- Analytic solutions to Moment matching (unlike EP, ADF)
- One pass over data

Bayesian Moment Matching

Online
Distributed
Consistent ?
## Experiments

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Instances</th>
<th>oEM</th>
<th>oBMM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abalone</td>
<td>4177</td>
<td>-2.65</td>
<td>-1.82</td>
</tr>
<tr>
<td>Banknote</td>
<td>1372</td>
<td>-9.74</td>
<td>-9.65</td>
</tr>
<tr>
<td>Airfoil</td>
<td>1503</td>
<td>-15.86</td>
<td>-16.53</td>
</tr>
<tr>
<td>Transfusion</td>
<td>748</td>
<td>-13.26</td>
<td>-13.09</td>
</tr>
<tr>
<td>CCPP</td>
<td>9568</td>
<td>-16.53</td>
<td>-16.51</td>
</tr>
<tr>
<td>Comp.Ac</td>
<td>8192</td>
<td>-132.04</td>
<td>-118.82</td>
</tr>
<tr>
<td>Kinematics</td>
<td>8192</td>
<td>-10.37</td>
<td>-10.32</td>
</tr>
<tr>
<td>Northridge</td>
<td>2929</td>
<td>-18.31</td>
<td>-17.97</td>
</tr>
<tr>
<td>Plastic</td>
<td>1650</td>
<td>-9.46</td>
<td>-9.01</td>
</tr>
</tbody>
</table>
## Experiments

<table>
<thead>
<tr>
<th>Data (Features)</th>
<th>Instances</th>
<th>oEM</th>
<th>oBMM</th>
<th>oDMM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Heterogeneity(16)</td>
<td>3,930,257</td>
<td>-176.2</td>
<td>-174.3</td>
<td>-180.7</td>
</tr>
<tr>
<td>Magic (10)</td>
<td>19,000</td>
<td>-33.4</td>
<td>-32.1</td>
<td>-35.4</td>
</tr>
<tr>
<td>Year MSD (91)</td>
<td>515,345</td>
<td>-513.7</td>
<td>-506.5</td>
<td>-513.8</td>
</tr>
<tr>
<td>Miniboone (50)</td>
<td>130,064</td>
<td>-58.1</td>
<td>-54.7</td>
<td>-60.3</td>
</tr>
</tbody>
</table>

### Avg. Log-Likelihood

<table>
<thead>
<tr>
<th>Data (Features)</th>
<th>Instances</th>
<th>oEM</th>
<th>oBMM</th>
<th>oDMM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Heterogeneity(16)</td>
<td>3,930,257</td>
<td>77.3</td>
<td>81.7</td>
<td>17.5</td>
</tr>
<tr>
<td>Magic (10)</td>
<td>19,000</td>
<td>7.3</td>
<td>6.8</td>
<td>1.4</td>
</tr>
<tr>
<td>Year MSD (91)</td>
<td>515,345</td>
<td>336.5</td>
<td>108.2</td>
<td>21.2</td>
</tr>
<tr>
<td>Miniboone (50)</td>
<td>130,064</td>
<td>48.6</td>
<td>12.1</td>
<td>2.3</td>
</tr>
</tbody>
</table>

### Running Time
Bayesian Moment Matching

- **Discrete Data**: Omar (2015, PhD Thesis) for Dirichlets; Rashwan, Zhao & Poupart (AISTATS’16) for SPNs; Hsu & Poupart (NIPS’16) for Topic Modelling

- **Continuous Data**: Jaini & Poupart, 2016 (*arxiv*); Jaini, Rashwan et al, (PGM’16) for SPNs; Poupart, Chen, Jaini et al (NetworksML’16)

- **Sequence Data and Transfer Learning**: Jaini, Poupart et al, (submitted to ICLR’17)
Mixture Models

Consistent/Robust

Tensor Decomposition And Spectral Learning
Offline

Bayesian Learning

Can be distributed; Practical problems

ADF, EP, SGD, oEM
Non convergence and local optima

Online

SPNs

Bayesian Moment Matching for continuous SPNs
Comparison with other deep networks

Bayesian Moment Matching algorithm Consistent?
What is a Sum-Product Network?

Proposed by Poon and Domingos (UAI 2011) equivalent to Arithmetic Circuits (Darwiche 2003)

Deep architecture with clear semantics

Tractable Probabilistic Graphical Models
What is a Sum-Product Network?

\[ P(x_1, x_2) = P(x_1, x_2)x_1x_2 + P(x_1, \overline{x_2})x_1\overline{x_2} + P(\overline{x_1}, x_2)\overline{x_1}x_2 + P(\overline{x_1}, \overline{x_2})\overline{x_1}\overline{x_2} \]

SPNs - Directed Acyclic Graphs

A valid SPN is

- **Complete**: Each sum node has children with same scope
- **Decomposable**: Each product node has children with disjoint scope
SPN represents a joint distribution over a set of random variables

Example:
Query: $\Pr(X_1 = 1, X_2 = 0)$

$Pr(X_1 = 1, X_2 = 0) = \frac{34.8}{100}$

Linear Complexity – two bottom passes for any query
Learning SPNS

- **Parameter Learning**:
  - **Maximum Likelihood**: SGD (Poon & Domingos, 2011) slow convergence, inaccurate; EM (Perharz, 2015) inaccurate; Signomial Programming (Zhao & Poupart, 2016)
  - **Bayesian Learning**: BMM (Rashwan et al., 2016) accurate; Collapsed Variational Inference (Zhao et al., 2016) accurate

Discrete

- Online parameter learning for continuous SPNs (Jaini et al. 2016)
  - extend oBMM to Gaussian SPNS
Continuous SPNS

Gaussian

\[ N^1(X_1) \quad N^2(X_1) \quad N^3(X_2) \quad N^4(X_2) \]
Continuous SPNS

A Gaussian mixture of Gaussians (Unnormalised)

\[ \sum_{i=1}^{4} \mathcal{N}(x | \mu_i, \Sigma_i) \]
Continuous SPNS

Hierarchical mixture of Gaussians (Unnormalised)

\[
\begin{align*}
N^1(X_1) & \quad N^2(X_1) \quad N^3(X_2) \quad N^4(X_2)
\end{align*}
\]
Bayesian Learning of SPNs

Parameters: weights, means and precisions (inverse covariance matrices)

Prior: $P(w, \mu, \Lambda)$; product of Dirichlets and Normal-Wisharts

Likelihood:
$P(x; w, \mu, \Lambda) = SPN(x; w, \mu, \Lambda)$

Posterior:
$P(w, \mu, \Lambda; data) \propto P(w, \mu, \Lambda) \prod_n SPN(x_n; w, \mu, \Lambda)$
Bayesian Moment Matching Algorithm

Exact Bayesian Update

$\chi_t$

Product of Dirichlets and Normal-Wisharts

Projection

Moment Matching

Mixture of Product of Dirichlets and Normal-Wisharts
Overall Algorithm

• Recursive computation of all constants
  - Linear complexity in the size of SPN

• Moment Matching
  - System of linear equations
  - Linear complexity in the size of SPN

• Streaming Data
  - Posterior update: constant time w.r.t. amount of data
# Empirical Results

Avg. log-likelihood and standard error based on 10-fold cross-validation

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Flow Size</th>
<th>Quake</th>
<th>Banknote</th>
<th>Abalone</th>
<th>Kinematics</th>
<th>CA</th>
<th>Sensorless Drive</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>8</td>
<td>8</td>
<td>22</td>
<td>48</td>
</tr>
<tr>
<td>oBMM (random)</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-1.82  (\pm) 0.19</td>
<td>-11.19 (\pm) 0.03</td>
<td>-2.47 (\pm) 0.56</td>
<td>1.58 (\pm) 1.28</td>
</tr>
<tr>
<td>oEM (random)</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-11.36 (\pm) 0.19</td>
<td>-11.35 (\pm) 0.03</td>
<td>-31.34 (\pm) 1.07</td>
<td>-3.40 (\pm) 6.06</td>
</tr>
<tr>
<td>oBMM (GMM)</td>
<td>4.80 (\pm) 0.67</td>
<td>-3.84 (\pm) 0.16</td>
<td>-4.81 (\pm) 0.13</td>
<td>-1.21 (\pm) 0.36</td>
<td>-11.24 (\pm) 0.04</td>
<td>-1.78 (\pm) 0.59</td>
<td>-</td>
</tr>
<tr>
<td>oEM (GMM)</td>
<td>-0.49 (\pm) 3.29</td>
<td>-5.50 (\pm) 0.41</td>
<td>-4.81 (\pm) 0.13</td>
<td>-3.53 (\pm) 1.68</td>
<td>-11.35 (\pm) 0.03</td>
<td>-21.39 (\pm) 1.58</td>
<td>-</td>
</tr>
</tbody>
</table>

oBMM performs better than oEM
## Empirical Results

Avg. log-likelihood and standard error based on 10-fold cross-validation

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Flow Size</th>
<th>Quake</th>
<th>Banknote</th>
<th>Abalone</th>
<th>Kinematics</th>
<th>CA</th>
<th>Sensorless Drive</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>8</td>
<td>8</td>
<td>22</td>
<td>48</td>
</tr>
<tr>
<td>oBMM (random)</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-1.82 ± 0.19</td>
<td>-11.19 ± 0.03</td>
<td>-2.47 ± 0.56</td>
<td>1.58 ± 1.28</td>
</tr>
<tr>
<td>oBMM (GMM)</td>
<td><strong>4.80 ± 0.67</strong></td>
<td><strong>-3.84 ± 0.16</strong></td>
<td><strong>-4.81 ± 0.13</strong></td>
<td><strong>-1.21 ± 0.36</strong></td>
<td><strong>-11.24 ± 0.04</strong></td>
<td><strong>-1.78 ± 0.59</strong></td>
<td>-</td>
</tr>
<tr>
<td>SRBM</td>
<td>-0.79 ± 0.01</td>
<td><strong>-2.38 ± 0.01</strong></td>
<td>-2.76 ± 0.01</td>
<td>-2.28 ± 0.01</td>
<td>-5.55 ± 0.02</td>
<td>-4.95 ± 0.01</td>
<td>-26.91 ± 0.03</td>
</tr>
<tr>
<td>GenMMN</td>
<td>-0.40 ± 0.01</td>
<td>-3.83 ± 0.01</td>
<td><strong>-1.70 ± 0.03</strong></td>
<td>-3.29 ± 0.10</td>
<td>-11.36 ± 0.02</td>
<td>-5.41 ± 0.14</td>
<td>-29.41 ± 1.19</td>
</tr>
</tbody>
</table>

oBMM is competitive with SBRM and GenMMN
Mixture Models

Consistent/Robust

Tensor Decomposition
And Spectral Learning
Offline

Bayesian Learning
Can be distributed;
Practical problems

Bayesian Moment Matching algorithm
Consistent?

Online

ADF, EP, SGD, oEM
Non convergence and local optima

SPNs

Bayesian Moment Matching for continuous SPNs
Comparison with other deep networks

oBMM performs better than oEM and comparable to other techniques
Conclusion and Future Work

Contributions:
- Online and distributed Bayesian Moment Matching algorithm
- Performs better than oEM w.r.t time and accuracy
- Extended it to continuous SPNs
- Comparative analysis with other deep learning methods

Future Work:
- Theoretical properties of BMM – consistent?
- Generalize oBMM to exponential family
- Extension to sequential data and transfer learning
- Online structure learning of SPNs
Thank you!