Stochastic Expectation Propagation for Large Scale Gaussian Process Classification

Daniel Hernández-Lobato, José Miguel Hernández-Lobato, Yingzhen Li, Thang Bui and Richard E. Turner

1Universidad Autónoma de Madrid. 2Harvard University. 3University of Cambridge.

1. Introduction to Gaussian Process Classification (GPC)

Description: We assume \( p(y|x,f) = \Phi(yxf(x)) \), where \( \Phi(\cdot) \) is a probit function, \( y_i \in \{-1,1\} \), and \( f(x) \) is generated from a Gaussian process, i.e., \( f(x) \sim \mathcal{GP}(0,k(x,\cdot)) \), for some covariance function \( k(x,\cdot) \). Training costs \( \mathcal{O}(n^3) \) since approximating \( p(f|X) \) involves the inverse of an \( n \times n \) matrix. Hyper-parameters are learnt via type-II maximum likelihood.

2. Expectation Propagation (EP) for Large Scale GPC

Description: Inducing point representation in which the targets of the \( m \) inducing points are not marginalized. Allows for very efficient training! \( X = (x_1, \ldots x_m) \), \( T = (f(x_1), \ldots f(x_m)) \). Let \( f \equiv (f(x_1), \ldots f(x_m))^T \). The posterior for \( f \) is:

\[
p(f|y) \propto \frac{\prod_{i=1}^m p(y_i|f_i)p(f_i)}{\prod_{i=1}^m p(y_i)} \
\]

where we have used \( p(f|\tilde{T}) \approx \prod_{i=1}^m p(f_i|\tilde{T}) \) and \( \phi(f_i) = \Phi(y_i \sqrt{m_i}/\sqrt{s_i} + \tilde{T}) \), with \( m_i = K_{i,i}, n^{1/2}, s_i = K_{i,i} - K_{i,i}K_{i,i}^{-1}K_{i,i}, \). The posterior is approximated using Expectation Propagation:

\[
p(f|y) \approx q(f) = \frac{\prod_{i=1}^m \phi(f_i)p(f_i)}{Z_n}, \quad \tilde{\phi}_i = \arg \min_{\phi_i} KL(\phi_i q|\phi_i q^2), \quad \text{where} \, \phi_i = \bar{y}_i \exp\{-0.5\bar{y}_i^T \bar{\mu} \bar{\mu}^T + \bar{\mu} \bar{\mu}^T \} \text{ and } \bar{\mu} = K_{i,i}^{-1} \bar{y}_i.
\]

3. Learning the Hyper-parameters in Expectation Propagation

The gradient of log of the marginal likelihood estimate \( Z_n \) is:

\[
\frac{\partial \log Z_n}{\partial \xi} = \theta^T \frac{\partial \log p(f|x)}{\partial \xi} - \theta^T \frac{\partial \log \pi}{\partial \xi} + \sum_{i=1}^n \frac{\partial \log Z_i}{\partial \xi},
\]

where \( Z_i \) is the normalization constant of \( \phi \| \phi_i \| \phi_i \eta \) and \( \eta \) are expected sufficient statistics, and \( \theta \) and \( \eta \) are natural parameters.

4. Stochastic Expectation Propagation (SEP)

Reduces the memory cost of EP by a factor of \( n \). SEP uses a single global factor \( \phi = \prod_{i=1}^m \phi_i \) to approximate the likelihood, where \( \phi_i = (\prod_{i=1}^m \phi_i)^2 \).

Algorithm: Parallel EP - Batch Mode

1. For each approximate factor \( \phi_i \) to update:
   1.1: Compute cavity: \( q_i(\theta_i) \propto q(\theta_i)/\phi_i \).
   1.2: Update \( \phi_i \), \( \tilde{\phi}_i = \arg \min KL(\phi_i q_i|\phi_i q_i^2) \).
   2.2: Update \( \phi_i \), \( \tilde{\phi}_i = \arg \min KL(\phi_i q_i|\phi_i q_i^2) \).
   2.3: Accumulate: \( \phi_{\text{new}}(\theta_i) = \phi_{\text{new}}(\theta_i)/\phi_i(\theta_i) \).
   3.2: Reconstruct \( q_i(\theta_i) \propto \phi_{\text{new}}(\theta_i)/\phi_i(\theta_i) \).

Algorithm: Parallel SEP - Batch Mode

1: Set \( q(f) \) equal to the prior \( p(f|X) \).
2: For each exact factor \( \phi_i \) to incorporate:
   2.1: Compute cavity: \( q_i(\theta_i) \propto q(\theta_i)/\phi_i \).
   2.2: Find \( \phi_i \), \( \phi_i = \arg \min KL(\phi_i q_i|\phi_i q_i^2) \).
   2.3: Update \( q(f) \), \( q(f) \propto \phi_{\text{new}}(\theta_i)q_i(\theta_i) \).

5. Experimental Results

Avg. Negative test log likelihood and training time in seconds.

<table>
<thead>
<tr>
<th>Problem</th>
<th>ADF</th>
<th>EP</th>
<th>ADF</th>
<th>EP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Australian</td>
<td>0.70±0.07</td>
<td>0.69±0.07</td>
<td>0.63±0.05</td>
<td>0.67±0.06</td>
</tr>
<tr>
<td>Breast</td>
<td>1.12±0.06</td>
<td>1.11±0.05</td>
<td>1.11±0.05</td>
<td>1.12±0.05</td>
</tr>
<tr>
<td>Crabs</td>
<td>0.08±0.06</td>
<td>0.06±0.06</td>
<td>0.06±0.07</td>
<td>0.08±0.06</td>
</tr>
<tr>
<td>Heart</td>
<td>0.45±0.18</td>
<td>0.40±0.13</td>
<td>0.39±0.11</td>
<td>0.46±0.17</td>
</tr>
<tr>
<td>Ionosphere</td>
<td>0.29±0.18</td>
<td>0.26±0.19</td>
<td>0.28±0.16</td>
<td>0.33±0.19</td>
</tr>
<tr>
<td>Pima</td>
<td>0.52±0.07</td>
<td>0.52±0.07</td>
<td>0.49±0.05</td>
<td>0.62±0.09</td>
</tr>
<tr>
<td>Sonar</td>
<td>0.40±0.15</td>
<td>0.33±0.10</td>
<td>0.35±0.11</td>
<td>0.46±0.24</td>
</tr>
</tbody>
</table>
| Avg. Time   | 18.2±0.13 | 19.3±0.5 | 18.8±0.11 | 145.4±0.13 | 36.0±0.3 | 149±1.0

Number of training instances: MNIST 60,000 and Airline 2,127,068.

6. Conclusions

- Stochastic expectation propagation (SEP) can be used as a practical alternative to expectation propagation (EP) for training Gaussian Process Classifiers on small and large datasets.
- SEP reduces the memory cost from \( O(\text{nm}) \) to \( O(nm^2) \), which is very good if \( n \gg m \).
- ADF also provides similar results to expectation propagation, but only when the model is simple (small \( m \)), or when the number of training instances is very large (large \( m \)).

http://dhnzl.org/  daniel.hernandez@uam.es