Online Gaussian Estimation with Long-term Memory

Sai Xiao

University of California Santa Cruz
Introduction

**Problem**: on-line density estimation by mixture of Gaussian when the best experts change over time.

**Goal**: long-term memory learning algorithm.

**Existing work**: Mixing Past Posteriors in finite experts setting. [BW2002]

**Our proposal**: Mixing Past Posteriors or other long-term memory learning algorithm in infinite experts setting.
Recall MPP

- Make prediction $\hat{y}_t$ by
  $$p(\hat{y}_t) = w_t \cdot y_t$$

- Loss update:
  $$w_{t,i}^m = \frac{w_{t,i}e^{L_{t,i}}}{\text{normalization}}$$

- Mixing update:
  $$w_{t+1} = \alpha w_{t,i}^m + \sum_{q=0}^{t-1} \beta_t(q) w_q^m, \quad \sum_{q=0}^{t-1} \beta_t(q) = 1 - \alpha$$
Proposed Algorithm

Let $p_0(\mu) \sim N(\mu_0, \sigma_0^2)$, a conjugate prior for mean of Gaussian.

FOR $t = 1$ TO $T$ DO

- Make prediction $\hat{y}_t$ by

$$p(\hat{y}_t|y_{1:(t-1)}) \propto \int p(\hat{y}_t|\mu)p_{t-1}(\mu|y_{1:(t-1)})d\mu$$

- Get Loss update from posterior distribution:

$$\tilde{p}_t(\mu|y_{1:t}) \propto p_{t-1}(\mu|y_{1:(t-1)})p(y_t|\mu)$$

- Mixing update: $p_t(\mu|y_{1:t})$ is a linear combination of $\tilde{p}_t(\mu|y_{1:t})$ and past posteriors $\tilde{p}_q(\mu|y_{1:q})$, $q = 0, \cdots, t-1$.

$$p_t(\mu|y_{1:t}) = \alpha\tilde{p}_t(\mu|1:t) + \sum_{q=0}^{t-1} \beta_t(q)\tilde{p}_q(\mu|y_{1:q}), \quad \sum_{q=0}^{t-1} \beta_t(q) = 1 - \alpha$$
Claim: when $p_{t-1}(\mu|y_{1:(t-1)})$ is a mixture of Gaussian distribution, $p(y_t|\mu)$ is a Gaussian distribution. $\tilde{p}_t(\mu|y_{1:t})$ is also a mixture of Gaussian distribution with different weights.

Proof.

Assume without loss of generality that $p_{t-1}(\mu|y_{1:(t-1)})$ is a mixture of two univariate Gaussians.

$$p_{t-1}(\mu|y_{1:(t-1)}) = w N(\mu|a, \tau_1^2) + (1-w) N(\mu|b, \tau_2^2)$$

$$\tilde{p}_t(\mu|y_{1:t}) = \left[ w N(\mu|a, \tau_1^2) + (1-w) N(\mu|b, \tau_2^2) \right] N(y_t|\mu, \sigma^2)$$

$$= \frac{w q_1 h_1 + (1-w) q_2 h_2}{w q_1 + (1-w) q_2} = \frac{w q_1}{w q_1 + (1-w) q_2} h_1 + \frac{(1-w) q_2}{w q_1 + (1-w) q_2} h_2$$

where $q_1 = \int N(\mu|a, \tau_1^2) N(y_t|\mu, \sigma^2) du = N(y_t|a, \tau_1^2 + \sigma^2)$,

$q_2 = \int N(\mu|b, \tau_2^2) N(y_t|\mu, \sigma^2) du = N(y_t|b, \tau_2^2 + \sigma^2)$.

$h_1 \propto N(\mu|a, \tau_1^2) N(y_t|\mu, \sigma^2) = N(\mu|\frac{a \sigma^2 + \tau_1^2 y_t}{\sigma^2 + \tau_1^2}, \frac{\sigma^2 \tau_1^2}{\sigma^2 + \tau_1^2})$, 

$h_2 = N(\mu|\frac{b \sigma^2 + \tau_2^2 y_t}{\sigma^2 + \tau_2^2}, \frac{\sigma^2 \tau_2^2}{\sigma^2 + \tau_2^2})$
Loss Update

\[ \tilde{p}_t(\mu | y_1:t) \propto p_{t-1}(\mu | y_1:(t-1)) p(y_t | \mu) \]

Claim: when \( p_{t-1}(\mu | y_1:(t-1)) \) is a mixture of Gaussian distribution, \( p(y_t | \mu) \) is a Gaussian distribution. \( \tilde{p}_t(\mu | y_1:t) \) is also a mixture of Gaussian distribution with different weights.

Remark: In loss update step, a new set of Gaussian mixtures are created. The more past posterior in mixing update for \( p_{t-1}(\mu | y_1:(t-1)) \), the more number of new Gaussian mixtures you create.
Mixing update

Because every \( \tilde{p}_i(\mu|1:i), \ i = 1, \cdots, t \) is a mixture of Gaussians, \( p_t(\mu|y_1:t) \) is certainly a mixture of Gaussians.

\[
p_t(\mu|y_1:t) = \alpha \tilde{p}_t(\mu|1:t) + \sum_{q=0}^{t-1} \beta_t(q) \tilde{p}_q(\mu|y_1:q)
\]

**Figure:** Mixing strategy
Complexity

The number of distinct Gaussian mixtures

- **Static Experts** ($\alpha = 1$): 1
- Fixed Share to start vector: $t + 1$
- Fixed Share to Past (Uniform Past/Decaying Past): $2^t - 1$.
  - Long-term memory: contains all combinations from $y_1, \cdots, y_t$.
  - Based on remark in last slide, the mixture of Gaussians in all
    $\tilde{p}_i(\mu|1 : i)$, $i = 1, \cdots, t$ must be stored. The total number of Gaussian
    mixtures doubles every time after loss update.

<table>
<thead>
<tr>
<th>$t$</th>
<th>Mixture Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2 12</td>
</tr>
<tr>
<td>3</td>
<td>3 13 23 123</td>
</tr>
<tr>
<td>4</td>
<td>4 14 24 124 34 134 234 1234</td>
</tr>
</tbody>
</table>
Loss = - \log p(y_t | y_{1:(t-1)}).

Figure: Simulated 20 data.
Trick of reducing the number of mixture components

\[
  \begin{align*}
  t = 1 : & \quad 1 \\
  t = 2 : & \quad 2 \underline{\text{X}} \\
  t = 3 : & \quad 3 \phantom{\underline{\text{X}}} 13 \phantom{\underline{\text{X}}} 23 \phantom{\underline{\text{X}}} 123 \\
  t = 4 : & \quad 4 \phantom{\underline{\text{X}}} 14 \phantom{\underline{\text{X}}} 24 \phantom{\underline{\text{X}}} \underline{\text{124}} \phantom{\underline{\text{X}}} \underline{\text{34}} \phantom{\underline{\text{X}}} \underline{\text{134}} \phantom{\underline{\text{X}}} \underline{\text{234}} \phantom{\underline{\text{X}}} \underline{\text{1234}}
  \end{align*}
\]

- compare \( p(y_4|\text{all combination of } y_1, y_2, y_3) \) and keep good ones.
- compare ratio of predictive density: e.g. keep 124 or not? If 
  \( p(y_4|y_1, y_2) > p(y_2|y_1) \), keep it.
Figure: Simulated 100 data.
Figure: Simulated 1000 data.
After the presentation, I changed the method to reduce the number of mixture components. So, I did not include slide 10-13 in my report.

The new method and results are in section 4 of the report.