Lazy Gaussian Process Committee for Real-Time Online Regression

Han Xiao

IT Security, Informatics (I20)
Technical University of Munich
xiaoh@in.tum.de

February 27, 2013
1 Preliminaries
   Regression
   Gaussian Process Regression

2 Motivation

3 Related Work

4 Lazy Gaussian Process Committee
   Intuition
   Prediction
   Data allocation
   Incremental Update

5 Experimental Results
   Numerical Evaluations
   Mouse-Trajectory Prediction

6 Conclusions
Deterministic Regression

Problem statement

Given a training set \( \mathcal{D} := \{(x_n, y_n)\}_{n=1}^{N} \) of \( N \) pairs of input vectors \( x_n \) and noisy scalar outputs \( y_n \), learn a function \( f \) transforming an input into the output given by

\[
y_n = f(x_n) + \epsilon_n,
\]

where \( \epsilon_n \sim \mathcal{N}(0, \sigma^2) \) and \( \sigma^2 \) is the variance of the noise.

Once \( f \) is learned, the output of a test point \( x_* \) can be predicted by \( y_* := f(x_*) \).
Deterministic Regression

Problem statement

Given a training set $\mathcal{D} := \{(x_n, y_n)\}_{n=1}^{N}$ of $N$ pairs of input vectors $x_n$ and noisy scalar outputs $y_n$, learn a function $f$ transforming an input into the output given by

$$y_n = f(x_n) + \epsilon_n,$$

where $\epsilon_n \sim \mathcal{N}(0, \sigma^2)$ and $\sigma^2$ is the variance of the noise.

Once $f$ is learned, the output of a test point $x_*$ can be predicted by $y_* := f(x_*)$. 
Deterministic Regression
Probabilistic Regression

Given a test point \((x_*, y_*)\) and a learned function \(f\), we need the probability \(p(y_* | x_*, f)\) for anomaly detection.

For instance,

- If \(p(y_* | x_*, f) < 0.05\), then \(x_*\) is an anomaly;
- (according to \(f\), the event “the output of \(x_*\) takes value of \(y_*\)” rarely happens.)
- otherwise \(x_*\) is legit.
Probabilistic Regression

Given a test point \((x_*, y_*)\) and a learned function \(f\), we need the probability \(p(y_* | x_*, f)\) for anomaly detection. For instance,

- If \(p(y_* | x_*, f) < 0.05\), then \(x_*\) is an anomaly; (according to \(f\), the event “the output of \(x_*\) takes value of \(y_*\)” rarely happens.)
- otherwise \(x_*\) is legit.
Probabilistic Regression for Anomaly Detection

95% confidence interval
Probabilistic Regression for Anomaly Detection

95% confidence interval
Probabilistic Regression for Anomaly Detection

95% confidence interval

anomaly
Gaussian Process Regression (GPR)

Overview

- Bayesian probabilistic framework (allowing one to assess the uncertainty of predictions);
- nonlinear regression (capability of complex patterns);
- non-parametric method (no pre-assumption of the underlying data distribution);
- state-of-the-art.
Gaussian Process Regression

Definition

GPR assumes the observed outputs behave according to

\[ p(y | x_1, \ldots, x_N) = \mathcal{N}(0, K), \]

where

- \( y := [y_1, \ldots, y_N]^\top \) is a vector of output values;
- \( K \) is an \( N \times N \) covariance matrix, whose entries are given by a covariance function, i.e. \( K_{ij} := k(x_i, x_j) \).

Intuition

If \( x_i \) and \( x_j \) are similar (i.e. \( k(x_i, x_j) \) close to 1), then their outputs \( y_i \) and \( y_j \) should be similar as well.
Gaussian Process Regression

Parameter-fitting

A frequently used covariance function is

\[ k(x_i, x_j) := \kappa^2 \exp \left( -\frac{1}{2}(x_i - x_j)^\top W (x_i - x_j) \right) + \sigma^2 \delta_{ij}, \]

where \([\kappa^2, \sigma^2, \{W\}]^\top\) are hyperparameters that need to be estimated from the data. They can be derived by maximizing the marginal likelihood function using a gradient based optimizer.

Bad scaling on a large data set

In each iteration the computation of the likelihood and the derivatives involves inversion of a matrix of size \(N \times N\), which requires time \(O(N^3)\).
Gaussian Process Regression

Parameter-fitting

A frequently used covariance function is

\[ k(x_i, x_j) := \kappa^2 \exp\left(-\frac{1}{2}(x_i - x_j)^\top W(x_i - x_j)\right) + \sigma^2 \delta_{ij}, \]

where \([\kappa^2, \sigma^2, \{W\}]^\top\) are hyperparameters that need to be estimated from the data. They can be derived by maximizing the marginal likelihood function using a gradient based optimizer.

Bad scaling on a large data set

In each iteration the computation of the likelihood and the derivatives involves inversion of a matrix of size \(N \times N\), which requires time \(O(N^3)\).
Gaussian Process Regression

Prediction

Once the hyperparameters are estimated, the predictive distribution of $y_*$ conditional on the training set $\mathcal{D}$ is also Gaussian

$$p(y_* | \mathcal{D}, x_*) = \mathcal{N} \left( k_*^\top K^{-1} y, k_* - k_*^\top K^{-1} k_* \right),$$

where $k_* := [k(x_*, x_1), \ldots, k(x_*, x_N)]^\top$ and $k_* := k(x_*, x_*)$.

Bad scaling on a large data set

For each new test point, it requires $\mathcal{O}(N)$ for the predictive mean and $\mathcal{O}(N^2)$ for the predictive variance.
Efficiency Problem of GPR

Training
Given a data set \( D := \{(x_n, y_n)\}_{n=1}^{N} \), estimate the hyperparameters of the covariance function. \( \mathcal{O}(N^3) \) for each iteration!

Testing
Given a test point \((x_*, y_*)\), compute the predictive distribution. \( \mathcal{O}(N) \) for the mean; \( \mathcal{O}(N^2) \) for the variance!

Bad scaling
A simple implementation of GPR can handle problems with at most a few thousands training examples, which prevents it from real-time applications dealing with large amounts of data.
Motivation

• For anomaly detection, we need a probabilistic regression model; → we focus on GPR.

• For real-time anomaly detection, we need to optimize the efficiency of GPR → we propose a Lazy Gaussian Process Committee (LGPC).
  • based on a set of “lazy learners”;
  • remove the need of parameter-fitting;
  • efficient online learning;
  • fast prediction.
Motivation

- For anomaly detection, we need a probabilistic regression model;
  → we focus on GPR.
- For real-time anomaly detection, we need to optimize the efficiency of GPR
  → we propose a Lazy Gaussian Process Committee (LGPC).
  - based on a set of “lazy learners”;
  - remove the need of parameter-fitting;
  - efficient online learning;
  - fast prediction.
Motivation

- For anomaly detection, we need a probabilistic regression model; → we focus on GPR.
- For real-time anomaly detection, we need to optimize the efficiency of GPR → we propose a Lazy Gaussian Process Committee (LGPC).
  - based on a set of “lazy learners”;
  - remove the need of parameter-fitting;
  - efficient online learning;
  - fast prediction.
GPR Approximations

- Based on a small set of “inducing inputs” (e.g., [Smola and Bartlett 2001], [Seeger, Williams, and Lawrence 2003], [Snelson and Ghahramani 2006], [Quinonero-Candela and Rasmussen 2005]);
- based on a set of “smaller” Gaussian processes (e.g., [Tresp 2000], [Nguyen-tuong and Peters 2008], [Chen and Ren 2009]).
Basic Idea

\[ D := \{(x_n, y_n)\}_{n=1}^N \]
Basic Idea

All data

\[ \mathcal{D} := \{(x_n, y_n)\}_{n=1}^{N} \]

Covariance matrix of a full GP

\[ K \]

Data partition

\[ \mathcal{D} := \{(x_n, y_n)\}_{n=1}^{N} \]

\[ \mathcal{D}_1 \]
\[ \mathcal{D}_2 \]
\[ \mathcal{D}_Q \]
Basic Idea

All data

\[ \mathcal{D} := \{ (x_n, y_n) \}_{n=1}^{N} \]

Covariance matrix of a full GP

\[ \mathbf{K} \]

Data partition

\[ \mathcal{D} := \{ (x_n, y_n) \}_{n=1}^{N} \]

Lazy Gaussian process committee

\[ \mathcal{D}_1 \rightarrow \mathcal{D}_2 \rightarrow \cdots \rightarrow \mathcal{D}_Q \]

GP member

\[ \mathbf{K}_1 \]

GP member

\[ \mathbf{K}_2 \]

GP member

\[ \mathbf{K}_Q \]
Basic Idea

All data
\[ \mathcal{D} := \{(x_n, y_n)\}_{n=1}^N \]

Covariance matrix of a full GP
\[ K \]

Matrix inversion
\[ O(N^3) \]

Data partition

Lazy Gaussian process committee

GP member
\[ K_1 \]

GP member
\[ K_2 \]

GP member
\[ K_Q \]

\[ O(QT^3) \]
Prediction on Query Point

How to predict a query point \((x_*, y_*)\)?

Let \(D_1, \ldots, D_Q\) be the training data maintained by each GP member, the predictive distribution can be computed as

\[
\hat{p}(y_* | D, x_*) \propto c \prod_{q=1}^{Q} p(y_* | D_q, x_*) \frac{\left[p(y_*)\right]^{Q-1}}{\left[p(y_*)\right]^{Q-1}},
\]

where \(c\) is a normalization constant.

The predictive mean and variance are given by

\[
\mathbb{E}_{\hat{p}}^{(N)}(y_*) = C_{\hat{p}}^{(N)}(y_*) \sum_{q \in Q} \left(C(y_* | D_q^{(N)}, x_*)^{-1} \mathbb{E}(y_* | D_q^{(N)}, x_*)\right)
\]

\[
C_{\hat{p}}^{(N)}(y_*) = \left( - \frac{(Q - 1)}{k_{**}} + \sum_{q \in Q} C(y_* | D_q^{(N)}, x_*)^{-1}\right)^{-1}.
\]
Prediction on Query Point

How to predict a query point \((x_*, y_*)\)?

Let \(D_1, \ldots, D_Q\) be the training data maintained by each GP member, the predictive distribution can be computed as

\[
\hat{p}(y_* | D, x_*) \propto c \prod_{q=1}^{Q} p(y_* | D_q, x_*) \frac{1}{[p(y_*)]^{Q-1}},
\]

where \(c\) is a normalization constant.

The predictive mean and variance are given by

\[
\mathbb{E}_{\hat{p}}^{(N)}(y_*) = C_{\hat{p}}^{(N)}(y_*) \sum_{q \in Q} \left(C(y_* | D_q^{(N)}, x_*)^{-1} \mathbb{E}(y_* | D_q^{(N)}, x_*)\right)
\]

\[
C_{\hat{p}}^{(N)}(y_*) = \left(-\frac{(Q-1)}{k_{**}} + \sum_{q \in Q} C(y_* | D_q^{(N)}, x_*)^{-1}\right)^{-1}.
\]
Prediction on Query Point

How to predict a query point \((x_*, y_*)\)?

Let \(D_1, \ldots, D_Q\) be the training data maintained by each GP member, the predictive distribution can be computed as

\[
\hat{p}(y_* | D, x_*) \propto c \prod_{q=1}^{Q} p(y_* | D_q, x_*) \left[ p(y_*) \right]^{Q-1},
\]

where \(c\) is a normalization constant.

The predictive mean and variance are given by

\[
\mathbb{E}_{\hat{p}}^{(N)}(y_*) = \mathbb{C}_{\hat{p}}^{(N)}(y_*) \sum_{q \in Q} \left( \mathbb{C}(y_* | D_q^{(N)}, x_*)^{-1} \mathbb{E}(y_* | D_q^{(N)}, x_*) \right)
\]

\[
\mathbb{C}_{\hat{p}}^{(N)}(y_*) = \left( - \frac{(Q - 1)}{k_{**}} + \sum_{q \in Q} \mathbb{C}(y_* | D_q^{(N)}, x_*)^{-1} \right)^{-1}.
\]
Data Allocation

How to allocate a new training example \((x_{N+1}, y_{N+1})\)?

- The data point comes sequentially;
- adapting the model after receiving each data point;
- but the model is consists of a set of GP members;
- which GPs should be selected for including this new data point?

Formally, denote \(D_q^{(N)}\) as the training examples allocated to the \(q^{th}\) GP at time \(N\), the update rule is formalized as

\[
D_q^{(N+1)|A} := \begin{cases} 
D_q^{(N)} \cup \{(x_{N+1}, y_{N+1})\} & \text{if } q \in A; \\
D_q^{(N)} & \text{otherwise.}
\end{cases}
\]
Data Allocation

How to allocate a new training example \((x_{N+1}, y_{N+1})\)?

- The data point comes sequentially;
- adapting the model after receiving each data point;
- but the model is consists of a set of GP members;
- which GPs should be selected for including this new data point?

Formally, denote \(D_q^{(N)}\) as the training examples allocated to the \(q^{th}\) GP at time \(N\), the update rule is formalized as

\[
D_q^{(N+1|\mathcal{A})} := \begin{cases} 
D_q^{(N)} \cup \{(x_{N+1}, y_{N+1})\} & \text{if } q \in \mathcal{A}; \\
D_q^{(N)} & \text{otherwise.}
\end{cases}
\]
How to Select the Set $\mathcal{A}$?

- Combinatorial problem;
- select $\mathcal{A}$ by optimizing some criterion.
How to Select the Set $\mathcal{A}$?

A combinatorial problem;
select $\mathcal{A}$ by optimizing some criterion.
How to Select the Set $A$?

- Combinatorial problem;
- select $A$ by optimizing some criterion.
An Active Selection Policy

**Goal**: Select at most $S$ GPs from the committee, such that their data inclusion can yield the maximal improvement for prediction.

**Define**:

- *Reference Set* $\mathcal{R}$: both inputs $X_\mathcal{R}$ and outputs $y_\mathcal{R}$ are observed;

- *pseudo-likelihood*:
  \[
  L^{(N)}_\mathcal{R} := -\frac{|\mathcal{R}|}{2} \log(2\pi) + \frac{1}{2} \log \left| \mathbb{C}^{(N)}_p (y_\mathcal{R})^{-1} \right| - \\
  \frac{1}{2} \left( y_\mathcal{R} - \mathbb{E}^{(N)}_p (y_\mathcal{R}) \right)^\top \mathbb{C}^{(N)}_p (y_\mathcal{R})^{-1} \left( y_\mathcal{R} - \mathbb{E}^{(N)}_p (y_\mathcal{R}) \right).
  \]

**Optimal selection** $\mathcal{A}^*$ at time $N + 1$

\[
\mathcal{A}^* := \arg \max_{\mathcal{A} \subseteq Q} L^{(N+1)|\mathcal{A}}_\mathcal{R} - L^{(N)}_\mathcal{R}, \text{ subject to } |\mathcal{A}| \leq S,
\]
An Active Selection Policy

**Goal**: Select at most $S$ GPs from the committee, such that their data inclusion can yield the maximal improvement for prediction.

**Define**:

- *Reference Set* $\mathcal{R}$: both inputs $X_\mathcal{R}$ and outputs $y_\mathcal{R}$ are observed;

- *pseudo-likelihood*:

  $$
  L_\mathcal{R}^{\langle N \rangle} := -\frac{|\mathcal{R}|}{2} \log(2\pi) + \frac{1}{2} \log \left| C_\hat{p}^{\langle N \rangle}(y_\mathcal{R})^{-1} \right| - $$

  $$
  \frac{1}{2} \left( y_\mathcal{R} - \mathbb{E}_\hat{p}^{\langle N \rangle}(y_\mathcal{R}) \right)^\top C_\hat{p}^{\langle N \rangle}(y_\mathcal{R})^{-1} \left( y_\mathcal{R} - \mathbb{E}_\hat{p}^{\langle N \rangle}(y_\mathcal{R}) \right).
  $$

**Optimal selection** $\mathcal{A}^*$ at time $N + 1$

$$
\mathcal{A}^* := \arg \max_{\mathcal{A} \subseteq Q} L_\mathcal{R}^{\langle N+1 \mid \mathcal{A} \rangle} - L_\mathcal{R}^{\langle N \rangle}, \text{ subject to } |\mathcal{A}| \leq S,
$$
An Active Selection Policy

**Goal:** Select at most $S$ GPs from the committee, such that their data inclusion can yield the maximal improvement for prediction.

**Define:**

- *Reference Set* $\mathcal{R}$: both inputs $X_\mathcal{R}$ and outputs $y_\mathcal{R}$ are observed;
- *pseudo-likelihood*:

  $$L_\mathcal{R}^{\langle N \rangle} := -\frac{|\mathcal{R}|}{2} \log(2\pi) + \frac{1}{2} \log \left| C_\hat{p}^{\langle N \rangle} (y_\mathcal{R})^{-1} \right| - \frac{1}{2} \left( y_\mathcal{R} - \mathbb{E}_\hat{p}^{\langle N \rangle} (y_\mathcal{R}) \right)^\top C_\hat{p}^{\langle N \rangle} (y_\mathcal{R})^{-1} \left( y_\mathcal{R} - \mathbb{E}_\hat{p}^{\langle N \rangle} (y_\mathcal{R}) \right).$$

**Optimal selection $\mathcal{A}^*$ at time $N+1$**

$$\mathcal{A}^* := \arg \max_{\mathcal{A} \subseteq \mathcal{Q}} L_\mathcal{R}^{\langle N+1 \mid \mathcal{A} \rangle} - L_\mathcal{R}^{\langle N \rangle}, \text{ subject to } |\mathcal{A}| \leq S,$$
Diminishing Return

Let \( F(\mathcal{A}) := L^{N+1}_{\mathcal{R}} | \mathcal{A} \) – \( L^{N}_{\mathcal{R}} \)
Diminishing Return

Let $F(A) := L_{\mathcal{R}}^{N+1 | A} - L_{\mathcal{R}}^{N}$
Diminishing Return

Let $F(\mathcal{A}) := L^{(N+1)}_{\mathcal{R}}(\mathcal{A}) - L^{(N)}_{\mathcal{R}}$
Diminishing Return

Let \( F(\mathcal{A}) := L^{N+1}_{\mathcal{R}}(\mathcal{A}) - L^{N}_{\mathcal{R}} \)
Diminishing Return

Let $F(A) := L^{N+1 | A} - L^{N}$
Diminishing Return

Let $F(A) := L_{\mathcal{R}}^{N+1 | A} - L_{\mathcal{R}}^N$
Submodularity

Let $F(A) := L^{(N+1)}_\mathcal{R}(A) - L^{(N)}_\mathcal{R}$, the submodular characteristic of $F$ indicates that

Submodularity of $F$

for all $A \subseteq B \subseteq Q$ and $q \in Q \setminus B$ it holds that

$$F(A \cup \{q\}) - F(A) \geq F(B \cup \{q\}) - F(B).$$

- Greedy algorithm was often used (Krause and Guestrin 2007; Krause, Singh, and Guestrin 2008; Krause et al. 2007); iteratively adds the element $q^* := \arg\max_{q \in Q \setminus A} F(A \cup \{q\})$;
- no polynomial time algorithm can provide a better approximation guarantee unless $P = NP$ (Feige 1998).
Submodularity

Let $F(\mathcal{A}) := L_{\mathcal{R}}^{N+1|\mathcal{A}} - L_{\mathcal{R}}^{N}$, the submodular characteristic of $F$ indicates that

<table>
<thead>
<tr>
<th>Submodularity of $F$</th>
</tr>
</thead>
<tbody>
<tr>
<td>for all $\mathcal{A} \subseteq \mathcal{B} \subseteq \mathcal{Q}$ and $q \in \mathcal{Q} \setminus \mathcal{B}$ it holds that $F(\mathcal{A} \cup {q}) - F(\mathcal{A}) \geq F(\mathcal{B} \cup {q}) - F(\mathcal{B})$.</td>
</tr>
</tbody>
</table>

- Greedy algorithm was often used (Krause and Guestrin 2007; Krause, Singh, and Guestrin 2008; Krause et al. 2007); iteratively adds the element $q^* := \arg \max_{q \in \mathcal{Q} \setminus \mathcal{A}} F(\mathcal{A} \cup \{q\})$;

- no polynomial time algorithm can provide a better approximation guarantee unless $P = NP$ (Feige 1998).
Greedy subset selection

**Input**: desired size of selection $S$ ($\geq 2$)

**Output**: greedy selection $\mathcal{A}$

1. Initialization $\mathcal{A} \leftarrow \emptyset$, $\mathcal{J} \leftarrow \{1, \ldots, Q\}$;
2. $\forall j \in \mathcal{J} : \Delta_j \leftarrow F(\mathcal{A} \cup \{j\}) - F(\mathcal{A})$;
3. $j^* \leftarrow \arg \max_{j \in \mathcal{J}} \Delta_j$;
4. $\mathcal{A} \leftarrow \mathcal{A} \cup \{j^*\}$, $\mathcal{J} \leftarrow \mathcal{J} \setminus \{j^*\}$;
5. for $s \leftarrow 2$ to $S$ do
6.   repeat
7.     $j^* \leftarrow \arg \max_{j \in \mathcal{J}} \Delta_j$;
8.     $\Delta_{j^*} \leftarrow F(\mathcal{A} \cup \{j^*\}) - F(\mathcal{A})$;
9.     if $\forall j \in \mathcal{J} \setminus \{j^*\} : \Delta_{j^*} > \Delta_j$ then
10.    $\mathcal{A} \leftarrow \mathcal{A} \cup \{j^*\}$, $\mathcal{J} \leftarrow \mathcal{J} \setminus \{j^*\}$;
11.   end
12. until $|\mathcal{A}| = s$;
13. end
How to Update the Covariance Matrix of each GP?

**Incremental Update**

\[
K^{(N+1)} := \begin{bmatrix} K^{(N)} & u^T \\ u & v \end{bmatrix}, \quad J^{(N+1)} := \begin{bmatrix} J^{(N)} & \frac{1}{\mu} gg^T \\ g^T & \mu \end{bmatrix},
\]

with \( u := [k(x_{N+1}, x_1), \ldots, k(x_{N+1}, x_N)]^T \), \( v := k(x_{N+1}, x_{N+1}) \),
and

\[
g := -\mu J^{(N)} u, \quad \mu := \left( v - u^T J^{(N)} u \right)^{-1}.
\]

**Non-increasing Update**

\[
J^{(N+1)} := \left[ PJ^{(N+1)} P \right]_{\uparrow} - \frac{1}{r} s^T s,
\]

where \( s := [[k(x_m, x_1), \ldots, k(x_m, x_{N+1})]P]_{\uparrow} \) and \( r := k(x_m, x_m) \).
How to Update the Covariance Matrix of each GP?

**Incremental Update**

\[
K^{\langle N+1 \rangle} := \begin{bmatrix} K^{\langle N \rangle} & u^\top \\ u & v \end{bmatrix}, \\
J^{\langle N+1 \rangle} := \begin{bmatrix} J^{\langle N \rangle} + \frac{1}{\mu} gg^\top & g \\ g^\top & \mu \end{bmatrix},
\]

with \( u := [k(x_{N+1}, x_1), \ldots, k(x_{N+1}, x_N)]^\top \), \( v := k(x_{N+1}, x_{N+1}) \), and

\[
g := -\mu J^{\langle N \rangle} u, \quad \mu := \left( v - u^\top J^{\langle N \rangle} u \right)^{-1}.
\]

**Non-increasing Update**

\[
J^{\langle N+1 \rangle} := \left[ P J^{\langle N+1 \rangle} P \right]^{\uparrow} - \frac{1}{r} s^\top s,
\]

where \( s := \left[ [k(x_m, x_1), \ldots, k(x_m, x_{N+1})] P \right]^{\uparrow} \) and \( r := k(x_m, x_m) \).
Other Technical Details

- How to construct the reference set?
- Capacity of each GP? Which point should be removed?
- How to perform fast prediction using nearest GPs?

Details can be found in the paper.
Setup

Six large scale data sets.
delta: $7,129 \times 6$, bank: $8,192 \times 8$, cpuact: $8,192 \times 12$,
elevator: $8,752 \times 17$, houses: $20,640 \times 8$, sarcos: $44,484 \times 21$.

- Accuracy (root mean square error)
- Efficiency (training and prediction time)

Baseline

- GPR: standard GP regression; offline
- SGPP: sparse GP using pseudo-inputs (Snelson and Ghahramani 2006); offline
- LoGP: local GP regression (Nguyen-tuong and Peters 2008);
- BCM: Bayesian committee machine (Tresp 2000);
- SOGP: sparse online GP regression (Csato and Opper 2002).
Setup

Six large scale data sets.
delta: 7,129 × 6, bank: 8,192 × 8, cpuact: 8,192 × 12, elevator: 8,752 × 17, houses: 20,640 × 8, sarcos: 44,484 × 21.

- Accuracy (root mean square error)
- Efficiency (training and prediction time)

Baseline

- GPR: standard GP regression; offline
- SGPP: sparse GP using pseudo-inputs (Snelson and Ghahramani 2006); offline
- LoGP: local GP regression (Nguyen-tuong and Peters 2008);
- BCM: Bayesian committee machine (Tresp 2000);
- SOGP: sparse online GP regression (Csato and Opper 2002).
Setup

Six large scale data sets.

delta: 7,129 × 6, bank: 8,192 × 8, cpuact: 8,192 × 12,
elevator: 8,752 × 17, houses: 20,640 × 8, sarcos: 44,484 × 21.

- Accuracy (root mean square error)
- Efficiency (training and prediction time)

Baseline

- GPR: standard GP regression; offline
- SGPP: sparse GP using pseudo-inputs (Snelson and Ghahramani 2006); offline
- LoGP: local GP regression (Nguyen-tuong and Peters 2008);
- BCM: Bayesian committee machine (Tresp 2000);
- SOGP: sparse online GP regression (Csato and Opper 2002).
Setup

Six large scale data sets.
delta: 7, 129 × 6, bank: 8, 192 × 8, cpuact: 8, 192 × 12, elevator: 8, 752 × 17, houses: 20, 640 × 8, sarcos: 44, 484 × 21.

- Accuracy (root mean square error)
- Efficiency (training and prediction time)

Baseline

- GPR: standard GP regression; offline
- SGPP: sparse GP using pseudo-inputs (Snelson and Ghahramani 2006); offline
- LoGP: local GP regression (Nguyen-tuong and Peters 2008);
- BCM: Bayesian committee machine (Tresp 2000);
- SOGP: sparse online GP regression (Csato and Opper 2002).
LGPC Setup

- 20 GP members (size of the committee);
- all GP were initialized randomly
- each GP maintained at most 100 training examples (capacity);
- each time 5 members were selected for data inclusion ($|A|$);

Ten repetitions on each data sets.
## LGPC versus Baseline Methods

Smaller value is better

<table>
<thead>
<tr>
<th>Model</th>
<th>delt</th>
<th>bank</th>
<th>cpua</th>
<th>elev</th>
<th>hous</th>
<th>sarc</th>
</tr>
</thead>
<tbody>
<tr>
<td>LGPC</td>
<td>0.041</td>
<td>0.084</td>
<td>0.072</td>
<td>0.053</td>
<td>0.157</td>
<td>0.032</td>
</tr>
<tr>
<td>LoGP</td>
<td>0.065</td>
<td>0.188</td>
<td>0.219</td>
<td>0.107</td>
<td>0.232</td>
<td>0.089</td>
</tr>
<tr>
<td>BCM_o</td>
<td>0.044</td>
<td>0.113</td>
<td>0.115</td>
<td>0.066</td>
<td>0.164</td>
<td>0.070</td>
</tr>
<tr>
<td>BCM_s</td>
<td>0.043</td>
<td>0.108</td>
<td>0.114</td>
<td>0.069</td>
<td>0.180</td>
<td>0.073</td>
</tr>
<tr>
<td>BCM_a</td>
<td>0.045</td>
<td>0.122</td>
<td>0.119</td>
<td>0.083</td>
<td>0.203</td>
<td>0.077</td>
</tr>
<tr>
<td>SOGP</td>
<td>0.040</td>
<td>0.047</td>
<td>0.074</td>
<td>0.038</td>
<td>0.143</td>
<td>0.023</td>
</tr>
<tr>
<td>GPR</td>
<td>0.039</td>
<td>0.041</td>
<td>0.030</td>
<td>0.031</td>
<td>0.115</td>
<td>0.016</td>
</tr>
<tr>
<td>SGPP</td>
<td>0.045</td>
<td>0.061</td>
<td>0.079</td>
<td>0.065</td>
<td>0.161</td>
<td>0.095</td>
</tr>
</tbody>
</table>
Comparison of Computation Speed

Time cost in second required for training and predicting

![Graph showing comparison of computation speed for different methods. The x-axis represents the number of training/test points (×10^3), and the y-axis represents the training/prediction time in seconds. The graph compares methods including GPR, SGPP, LoGP, LGPC, BCMo, BCMs, BCMa, and SOGP.]
Exploration of Model Parameters

Size of the committee

For large data sets, increasing the size of the committee leads to higher predictive accuracy.

<table>
<thead>
<tr>
<th>$Q$</th>
<th>delt</th>
<th>bank</th>
<th>cpua</th>
<th>elev</th>
<th>hous</th>
<th>sarc</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>0.043</td>
<td>0.093</td>
<td>0.087</td>
<td>0.065</td>
<td>0.171</td>
<td>0.034</td>
</tr>
<tr>
<td>10</td>
<td>0.042</td>
<td>0.102</td>
<td>0.079</td>
<td>0.056</td>
<td>0.167</td>
<td>0.033</td>
</tr>
<tr>
<td>15</td>
<td><strong>0.041</strong></td>
<td>0.085</td>
<td>0.075</td>
<td>0.057</td>
<td>0.161</td>
<td>0.034</td>
</tr>
<tr>
<td>20</td>
<td>0.041</td>
<td>0.084</td>
<td><strong>0.072</strong></td>
<td>0.053</td>
<td>0.157</td>
<td>0.033</td>
</tr>
<tr>
<td>25</td>
<td>0.041</td>
<td><strong>0.076</strong></td>
<td>0.080</td>
<td>0.053</td>
<td>0.156</td>
<td><strong>0.032</strong></td>
</tr>
<tr>
<td>30</td>
<td>0.041</td>
<td>0.085</td>
<td>0.095</td>
<td><strong>0.052</strong></td>
<td><strong>0.155</strong></td>
<td>0.032</td>
</tr>
</tbody>
</table>
Exploration of Model Parameters

Capacity of each GP member

Increasing the capacity of each GP member improves the predictive accuracy.

<table>
<thead>
<tr>
<th>$T$</th>
<th>delt</th>
<th>bank</th>
<th>cpua</th>
<th>elev</th>
<th>hous</th>
<th>sarc</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>0.042</td>
<td>0.114</td>
<td>0.098</td>
<td>0.072</td>
<td>0.172</td>
<td>0.040</td>
</tr>
<tr>
<td>100</td>
<td>0.041</td>
<td>0.084</td>
<td>0.072</td>
<td>0.053</td>
<td>0.157</td>
<td>0.032</td>
</tr>
<tr>
<td>150</td>
<td><strong>0.040</strong></td>
<td>0.068</td>
<td>0.057</td>
<td>0.040</td>
<td>0.152</td>
<td>0.028</td>
</tr>
<tr>
<td>200</td>
<td>0.040</td>
<td><strong>0.054</strong></td>
<td><strong>0.050</strong></td>
<td><strong>0.039</strong></td>
<td><strong>0.146</strong></td>
<td><strong>0.022</strong></td>
</tr>
</tbody>
</table>
Mouse-Trajectory in Online Banking

Applied LGPC for learning mouse-trajectory of different users in an Internet banking scenario.

Instruction
Please click start to start the task.
user_id=1358622719256
Mouse-Trajectory in Online Banking

Applied LGPC for learning mouse-trajectory of different users in an Internet banking scenario.
Mouse-Trajectory in Online Banking

Applied LGPC for learning mouse-trajectory of different users in an Internet banking scenario.

[Account overview]

Your financial overview
- Amount credit: 1258.66 EUR
- Amount debit: 0.00 EUR
- Overall balance: 1258.66 EUR
Mouse-Trajectory in Online Banking

Applied LGPC for learning mouse-trajectory of different users in an Internet banking scenario.

[Transaction details]

<table>
<thead>
<tr>
<th>Transaction details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Account: 4774033</td>
</tr>
<tr>
<td>Current account balance: 1258.66 EUR (Drawing credit: 0.00 EUR)</td>
</tr>
<tr>
<td>Beneficiary: Name, First name / Company (max. 27 characters)</td>
</tr>
<tr>
<td>Account No. of the beneficiary: (0118359)</td>
</tr>
<tr>
<td>Bank code (BLZ): (0110001)</td>
</tr>
<tr>
<td>Amount (EUR.Ct.): (1000)</td>
</tr>
<tr>
<td>Execution date: (06/01/2013 (mm/dd/yyyy))</td>
</tr>
<tr>
<td>Payment details (e.g., customer reference number, max. 108 characters) (optional)</td>
</tr>
<tr>
<td>Principal: Jane, Smith, Account No.: 4774033</td>
</tr>
<tr>
<td>Clear form</td>
</tr>
</tbody>
</table>
Mouse-Trajectory in Online Banking

Applied LGPC for learning mouse-trajectory of different users in an Internet banking scenario.

[TAN authentication]

Transactions
Domestic transfer order
Future date transfer
Overseas remittance
Transfer to a sub-account
Transfer order templates
Standing orders
Activate TAN list

Your transfer order

Beneficiary: Yurong, Tao
Account of the beneficiary: 0118369
BLZ: 0110001
Financial institution: DEUTSCHE BANK PGK SAAR
Amount: 1000 EUR
Payment details
Sender Name: Jane, Smith
Your account: 4774030 00
TAN-Input

Please enter the following TAN:
TAN No. 025

[change order] execute transfer order

Current step
1. Enter data
2. You are here: Check and release data
3. Confirmation
Mouse-Trajectory in Online Banking
Applied LGPC for learning mouse-trajectory of different users in an Internet banking scenario.

[Confirmation]
Setup

- 10 participants, each with three trials;
- input information was same for all trials;
- Javascript code was developed for tracking mouse coordinates on every `onmousemove` event;

The trajectories of the first two trials (ca. 2700 points/user) were used for training models. The goal was to predict the trajectory of the last trial (ca. 1000 points/user).

Baselines: GPR and SOGP
Setup

- 10 participants, each with three trials;
- input information was same for all trials;
- Javascript code was developed for tracking mouse coordinates on every onmousemove event;

The trajectories of the first two trials (ca. 2700 points/user) were used for training models.

The goal was to predict the trajectory of the last trial (ca. 1000 points/user).

Baselines: GPR and SOGP
Setup

- 10 participants, each with three trials;
- input information was same for all trials;
- Javascript code was developed for tracking mouse coordinates on every `onmousemove` event;

The trajectories of the first two trials (ca. 2700 points/user) were used for training models.
The goal was to predict the trajectory of the last trial (ca. 1000 points/user).
Baselines: GPR and SOGP
Trajectory Prediction

“Transaction details”

LGPC

SOGP

GPR
Trajectory Prediction

“TAN authentication”
Remarks

Why LGPC?
a new training point arriving about every 10ms
(less than one minute of running time will result in thousands of data points)

LGPC is a more preferable method due to its fast learning speed.

Potential applications
- Distinguishing between individuals;
- early warning of identity theft.
Remarks

Why LGPC?
a new training point arriving about every 10ms
(less than one minute of running time will result in thousands of
data points)

LGPC is a more preferable method due to its fast learning speed.

Potential applications
  • Distinguishing between individuals;
  • early warning of identity theft.
Conclusions

Focused on the low-efficiency problem of Gaussian process regression.

- proposed a novel approximation scheme for real-time online learning;
- no training is required → new data is allocated by optimizing a submodular function;
- promising performance (accuracy and efficiency) on real-world data sets.

Future work

- determine the optimal size of the committee;
- effective strategies for initializing the hyperparameters;
- infringe the independence between GP members.
Conclusions

Focused on the low-efficiency problem of Gaussian process regression.

- proposed a novel approximation scheme for real-time online learning;
- no training is required → new data is allocated by optimizing a submodular function;
- promising performance (accuracy and efficiency) on real-world data sets.

Future work

- determine the optimal size of the committee;
- effective strategies for initializing the hyperparameters;
- infringe the independence between GP members.
Unuseless in HIVE