Estimating Mutual Information on Data Streams

Pavel Efros – Fabian Keller, Emmanuel Müller, Klemens Böhm
Mutual Information

... an information-theoretic correlation measure.

**Intuitively:** *The reduction of uncertainty on one random variable* \( X \) *given knowledge of another variable* \( Y \).

**Advantage over traditional correlation measures:** non-linear, sensitive to any kind of dependence.

**Application domains:** Many (our example: stock data)
Mutual Information

... an information-theoretic correlation measure.

Intuitively: *The reduction of uncertainty on one random variable X given knowledge of another variable Y.*

**Advantage over traditional correlation measures:** non-linear, sensitive to any kind of dependence.

**Application domains:** Many (our example: stock data)

*Unsolved problem on data streams!*
Data streams ⇒ Dependencies change over time
Mutual Information On Data Streams

Data streams ⇒ Dependencies change over time

Analysts have many questions:
What is the mutual information in 2000?
Mutual Information On Data Streams

Data streams ⇒ Dependencies change over time

Analysts have many questions:

How does the MI of 2000 compare to the overall MI?
Mutual Information On Data Streams

Data streams ⇒ Dependencies change over time

Analysts have many questions:

... and to the MI in the 90's?
Mutual Information On Data Streams

Data streams ⇒ Dependencies change over time

Analysts have many questions:

... and in 2000 if go from February to February instead?
Mutual Information On Data Streams

Data streams $\Rightarrow$ Dependencies change over time

Not only retrospective –
Even more examples for online scenario (continuous / rolling queries)
Contributions

We provide a DSMS which can answer such ad-hoc queries efficiently.
Contributions

We provide a DSMS which can answer such ad-hoc queries efficiently.

But: **Approximation is required** to avoid storing the entire data stream.
Queries on Multiple Time Scales

Analysts may compare:

*Current minute vs previous minute?*

*Today vs yesterday?*

*Current year vs previous year?*

 Raises the question:

How to achieve same quality over all time scales?
Contributions

- We provide a DSMS which can answer such ad-hoc queries efficiently.
- Multiscale Sampling
Agenda

- Introduction
  - Challenges
  - Contributions

- MISE (Mutual Information Stream Estimator)
  - Query Anchor
  - Framework
  - Multiscale Sampling

- Experiments

- Conclusion
Mutual Information Estimation Basics

On static data: Kraskov principle has emerged as the leading estimator (good bias/variance properties)

Our goal: Make use of this established estimation principle

Requires: Handling the dynamics of nearest neighbor relationships
Dynamics of Nearest Neighbors

Dynamics of:

- $k$ nearest neighbor distance (let $k = 1$)
- Marginal counts $\equiv$ number of points in marginal slice

\[
\begin{array}{cccc}
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
\end{array}
\]

\[
\begin{array}{cccc}
t & MC_x & MC_y \\
0 & 0 & 0 & 0 \\
\end{array}
\]
Dynamics of Nearest Neighbors

Dynamics of:

- $k$ nearest neighbor distance (let $k = 1$)
- Marginal counts $\equiv$ number of points in marginal slice

\[
\begin{array}{ccc}
1 & 0 & 1 \\
0 & 0 & 01 \\
0 & 0 & 1 \\
0 & 1 & 1 \\
\end{array}
\]

\[
\begin{array}{c|cc}
0 & 0 & 0 \\
1 & 0 & 1 \\
\end{array}
\]

\[
\begin{array}{cccc}
t & MC_x & MC_y \\
0 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 \\
\end{array}
\]
Dynamics of Nearest Neighbors

Dynamics of:

- $k$ nearest neighbor distance (let $k = 1$)
- Marginal counts $\equiv$ number of points in marginal slice

\[
\begin{array}{ccc}
0 & 0 & 0 \\
1 & 1 & 0 \\
2 & 1 & 1 \\
\end{array}
\]

\[
\begin{array}{cccc}
t & MC_x & MC_y \\
0 & 0 & 0 \\
1 & 0 & 1 \\
2 & 1 & 1 \\
\end{array}
\]
Dynamics of Nearest Neighbors

Dynamics of:

- $k$ nearest neighbor distance (let $k = 1$)
- Marginal counts $\equiv$ number of points in marginal slice

```
 0 1 2 3
0 0 0 1
2 1 1 0
3 1 0 0
```

```
<table>
<thead>
<tr>
<th>t</th>
<th>MC_x</th>
<th>MC_y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
```
Dynamics of Nearest Neighbors

Dynamics of:

- $k$ nearest neighbor distance (let $k = 1$)
- Marginal counts $\equiv$ number of points in marginal slice

\begin{tabular}{cccc}
& $t$ & $MC_x$ & $MC_y$ \\
0 & 0 & 0 & 0 \\
1 & 0 & 1 & 1 \\
2 & 1 & 1 & 1 \\
3 & 1 & 1 & 1 \\
4 & 1 & 2 & 2 \\
\end{tabular}
Dynamics of Nearest Neighbors

Dynamics of:
- $k$ nearest neighbor distance (let $k = 1$)
- Marginal counts $\equiv$ number of points in marginal slice

\begin{table}[h]
\begin{tabular}{cccc}
\hline
$t$ & $MC_x$ & $MC_y$ \\
\hline
0 & 0 & 0 \\
1 & 0 & 1 \\
2 & 1 & 1 \\
3 & 1 & 1 \\
4 & 1 & 2 \\
5 & 1 & 3 \\
\hline
\end{tabular}
\end{table}

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{chart.png}
\caption{Chart showing dynamics of nearest neighbors.}
\end{figure}
Dynamics of Nearest Neighbors

Dynamics of:

- $k$ nearest neighbor distance (let $k = 1$)
- Marginal counts $\equiv$ number of points in marginal slice

<table>
<thead>
<tr>
<th>$t$</th>
<th>$MC_x$</th>
<th>$MC_y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>
Dynamics of Nearest Neighbors

Dynamics of:
- \( k \) nearest neighbor distance (let \( k = 1 \))
- Marginal counts \( \equiv \) number of points in marginal slice

\[
\begin{array}{ccccccc}
1 & 0 & 1 & & & & \\
0 & 0 & 0 & 1 & & & \\
2 & 1 & 1 & & & & \\
3 & 1 & 1 & & & & \\
4 & 1 & 2 & & & & \\
5 & 1 & 3 & & & & \\
6 & 2 & 3 & & & & \\
7 & 2 & 0 & & & & \\
\end{array}
\]
Dynamics of Nearest Neighbors

Dynamics of:
- $k$ nearest neighbor distance (let $k = 1$)
- Marginal counts $\equiv$ number of points in marginal slice

![Graph showing dynamics of $MC_x$ over time](image_url)
Mutual Information Estimation

Why is this data important?

<table>
<thead>
<tr>
<th>t</th>
<th>MC_x</th>
<th>MC_y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

Adapted Kraskov principle:

\[ M_{I \text{est}} = \psi(k) + \psi(N) - \psi(MC_x + 1) - \psi(MC_y + 1) \]
Mutual Information Estimation

Why is this data important?

<table>
<thead>
<tr>
<th>t</th>
<th>$MC_x$</th>
<th>$MC_y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

Adapted Kraskov principle:

$$MI_{est} = \psi(k) + \psi(N) - \psi(MC_x + 1) - \psi(MC_y + 1)$$

Digamma function

For subsequence $Q_0^4$:

$$MI_{est} = \psi(1) + \psi(5) - \psi(2) - \psi(3)$$
Mutual Information Estimation

Why is this data important?

<table>
<thead>
<tr>
<th>$t$</th>
<th>$MC_x$</th>
<th>$MC_y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

Adapted Kraskov principle:

\[ Ml_{est} = \psi(k) + \psi(N) - \psi(MC_x + 1) - \psi(MC_y + 1) \]

For subsequence $Q_0^7$:

\[ Ml_{est} = \psi(1) + \psi(8) - \psi(3) - \psi(1) \]
... a data structure which keeps track of marginal counts.

- “lives” at a certain time point $t$
- stores the $X$ and $Y$ values of the corresponding data point.
- a method $\text{INSERTRIGHT}(Q_{t'})$
  which adds a data point $Q_{t'}$ in forward time direction, i.e., $t' > t$,
- a method $\text{INSERTLEFT}(Q_{t'})$
  which adds a data point $Q_{t'}$ in backward time direction, i.e., $t' < t$,
- a method $\text{QUERY}(t_1, t_2)$
  which returns the marginal counts $MC_x(Q_t, Q_{t_1}^{t_2})$ and $MC_y(Q_t, Q_{t_1}^{t_2})$,
Query Anchor

... a data structure which keeps track of marginal counts.

- "lives" at a certain time point $t$
- stores the $X$ and $Y$ values of the corresponding data point.
- a method $\text{INSERTRIGHT}(Q_{t'})$
  which adds a data point $Q_{t'}$ in forward time direction, i.e., $t' > t$,
- a method $\text{INSERTLEFT}(Q_{t'})$
  which adds a data point $Q_{t'}$ in backward time direction, i.e., $t' < t$,
- a method $\text{QUERY}(t_1, t_2)$
  which returns the marginal counts $MC_X(Q_t, Q_{t_2}^{t_1})$ and $MC_Y(Q_t, Q_{t_2}^{t_1})$, 
Query Anchor

... a data structure which keeps track of marginal counts.

- “lives” at a certain time point $t$
- stores the $X$ and $Y$ values of the corresponding data point.
- a method $\text{INSERTRIGHT}(Q_{t'})$
  which adds a data point $Q_{t'}$ in forward time direction, i.e., $t' > t$,
- a method $\text{INSERTLEFT}(Q_{t'})$
  which adds a data point $Q_{t'}$ in backward time direction, i.e., $t' < t$,
- a method $\text{QUERY}(t_1, t_2)$
  which returns the marginal counts $MC_X(Q_t, Q_{t_2}^{t_1})$ and $MC_Y(Q_t, Q_{t_2}^{t_1})$,
... a data structure which keeps track of marginal counts.

- “lives” at a certain time point $t$
- stores the $X$ and $Y$ values of the corresponding data point.
- a method \texttt{INSERT\_RIGHT}(Q_{t'})
  which adds a data point $Q_{t'}$ in forward time direction, i.e., $t' > t$,
- a method \texttt{INSERT\_LEFT}(Q_{t'})
  which adds a data point $Q_{t'}$ in backward time direction, i.e., $t' < t$,
- a method \texttt{QUERY}(t_1, t_2)
  which returns the marginal counts $MC_x(Q_t, Q_{t_1}^{t_2})$ and $MC_y(Q_t, Q_{t_1}^{t_2})$, 

... a data structure which keeps track of marginal counts.

- “lives” at a certain time point $t$
- stores the $X$ and $Y$ values of the corresponding data point.
- a method `INSERTRIGHT($Q_{t'}$)`
  which adds a data point $Q_{t'}$ in forward time direction, i.e., $t' > t$,
- a method `INSERTLEFT($Q_{t'}$)`
  which adds a data point $Q_{t'}$ in backward time direction, i.e., $t' < t$,
- a method `QUERY($t_1$, $t_2$)`
  which returns the marginal counts $MC_x(Q_t, Q_{t_2}^{t_1})$ and $MC_y(Q_t, Q_{t_2}^{t_1})$, 

Query Anchor

... a data structure which keeps track of marginal counts.

- “lives” at a certain time point $t$
- stores the $X$ and $Y$ values of the corresponding data point.
- a method \texttt{INSERT\_RIGHT}(Q_{t'})
  which adds a data point $Q_{t'}$ in forward time direction, i.e., $t' > t$,
- a method \texttt{INSERT\_LEFT}(Q_{t'})
  which adds a data point $Q_{t'}$ in backward time direction, i.e., $t' < t$,
- a method \texttt{QUERY}(t_1, t_2)
  which returns the marginal counts $MC_x(Q_t, Q_{t_2}^{t_1})$ and $MC_y(Q_t, Q_{t_2}^{t_1})$,.
... a data structure which keeps track of marginal counts.

- “lives” at a certain time point $t$
- stores the $X$ and $Y$ values of the corresponding data point.
- a method $\text{INSERTRIGHT}(Q_{t'})$
  which adds a data point $Q_{t'}$ in forward time direction, i.e., $t' > t$,
- a method $\text{INSERTLEFT}(Q_{t'})$
  which adds a data point $Q_{t'}$ in backward time direction, i.e., $t' < t$,
- a method $\text{QUERY}(t_1, t_2)$
  which returns the marginal counts $MC_X(Q_t, Q_{t_2}^{t_1})$ and $MC_Y(Q_t, Q_{t_2}^{t_1})$, 
Query Anchor

... a data structure which keeps track of marginal counts.

- “lives” at a certain time point $t$
- stores the $X$ and $Y$ values of the corresponding data point.
- a method $\text{INSERTRIGHT}(Q_{t'})$
  which adds a data point $Q_{t'}$ in forward time direction, i.e., $t' > t$,
- a method $\text{INSERTLEFT}(Q_{t'})$
  which adds a data point $Q_{t'}$ in backward time direction, i.e., $t' < t$,
- a method $\text{QUERY}(t_1, t_2)$
  which returns the marginal counts $MC_X(Q_t, Q_{t_1}^{t_2})$ and $MC_Y(Q_t, Q_{t_2}^{t_1})$, 
Information Stored in a Query Anchor
MISE Framework – Idea

Ensemble of Query Anchors

Insert:
- Add new anchor
- Forward initialization
- Reverse initialization
- Sampling (decides which anchors to delete)
**MISE Framework – Idea**

Ensemble of Query Anchors

**Insert:**
- Add new anchor
- Forward initialization
- Reverse initialization
- Sampling (decides which anchors to delete)
MISE Framework – Idea

Ensemble of Query Anchors

Insert:
- Add new anchor
- Forward initialization
- Reverse initialization
- Sampling (decides which anchors to delete)
MISE Framework – Idea

Ensemble of Query Anchors

Insert:
- Add new anchor
- Forward initialization
- Reverse initialization
- Sampling (decides which anchors to delete)
MISE Framework – Idea

Ensemble of Query Anchors

Insert:
- Add new anchor
- Forward initialization
- Reverse initialization
- Sampling (decides which anchors to delete)
MISE Framework – Idea

Ensemble of Query Anchors

Query:
- Determine relevant query anchors
- Delegate query to anchors
- Aggregate estimation results
MISE Framework – Idea

Ensemble of Query Anchors

Query:
- Determine relevant query anchors
- Delegate query to anchors
- Aggregate estimation results
Multiscale Sampling

Define unit-free quantity of window size vs offset:

$$\Delta \equiv \frac{O}{W}$$
Multiscale Sampling

Multiscale Query Equivalence:
Define equivalence relation $\triangleq$ between queries $A$ and $B$ by:

$$A \triangleq B \iff \Delta_A = \Delta_B$$
Multiscale Sampling:

... is a sampling scheme which provides an equal expected number of sampling elements for all queries which belong to the same equivalence class.
Required Sampling Distribution

Multiscale sampling property fulfilled for (derivation in paper):

\[ P_n = \begin{cases} 
1 & \text{if } n \leq \alpha \\
\frac{\alpha}{n} & \text{otherwise}
\end{cases} \]

(1)
Required Sampling Distribution

Multiscale sampling property fulfilled for (derivation in paper):

\[ P_n = \begin{cases} 
1 & \text{if } n \leq \alpha \\
\frac{\alpha}{n} & \text{otherwise}
\end{cases} \]  

(1)
Required Sampling Distribution

Multiscale sampling property fulfilled for (derivation in paper):

\[
P_n = \begin{cases} 
1 & \text{if } n \leq \alpha \\
\frac{\alpha}{n} & \text{otherwise}
\end{cases}
\]  

(1)
Required Sampling Distribution

Multiscale sampling property fulfilled for (derivation in paper):

\[ P_n = \begin{cases} 
1 & \text{if } n \leq \alpha \\
\frac{\alpha}{n} & \text{otherwise} 
\end{cases} \]  

(1)
Sample Size

Relationship between $\alpha$ and sample size $S$ over time $T$:

$$E[S] = \lfloor \alpha \rfloor + O(\log T)$$

Two MISE versions:

- fixed $\alpha$, log-growing sample size $S$
- fixed sample size $S$, logarithmic adjustments to $\alpha$
EXPERIMENTS
Simple Example

Runtime Reference: 112.1 min
Runtime MISE: 3.8 min
Computed with only 1 query for every 10 inserts.
Experiment Question 1

... since we precompute queries

Question 1:
How many queries are required to make a stream approach worthwhile?

Trivially:
#queries = 0: Nothing to speed-up
#queries ≫ #inserts: Arbitrarily high speed-up
Experiment Question 1

Speed-up with ratio $\frac{\text{queries}}{\text{#inserts}}$: 1.0
Experiment Question 1

Speed-up with ratio $\frac{\text{#queries}}{\text{#inserts}}$: 0.1
Speed-up with ratio $\#\text{queries}/\#\text{inserts}: 0.01$

⇒ Stream processing worthwhile even when queries are rare.
Experiment Question 2

... since part of the query computation occurs on ’insert’

Question 2:
What is the raw processing speed (insert speed) of the stream?
Can we process high-frequent streams?
⇒ Despite precomputation:

Stream frequencies possible up to 100Hz (for high estimation quality) or 20kHz (for low estimation quality).
Question 3:
How does sampling influence estimation quality? What are the benefits of multiscale sampling?

huge experiment in the paper – complex results

- **Data**: 26 real world data streams
- **Ground truth**: unsampled estimator
- **Quality measures**: answerability, bias, variance
- **Competitors**: multiscale vs reservoir vs sliding window sampling
Question 3:
How does sampling influence estimation quality? What are the benefits of multiscale sampling?

huge experiment in the paper – complex results

⇒ Traditional sampling does not work with queries comprising multiple time scales
⇒ Very low estimation error
   Example: $\alpha = 1000, \Delta = 1.0 \Rightarrow$ MI value precise up to ±0.01
Conclusions

Summary

- First approach for estimating mutual information on data streams
- Proposal of multiscale sampling
- Experiments clearly show the benefit of a stream-specific solution
Conclusions

Summary

- First approach for estimating mutual information on data streams
- Proposal of multiscale sampling
- Experiments clearly show the benefit of a stream-specific solution

Thank you for your attention! Questions?