Exploiting Self-Similarity For Change-Detection

Giacomo Boracchi and Manuel Roveri
Dipartimento di Elettronica, Informazione, e Bioingegneria, Politecnico di Milano

Contact: giacomo.boracchi@polimi.it

IJCNN 2014, Beijing

11 July 2014
THE MOTIVATING IDEA

...and our contribution
Motivating Idea

- Detecting changes in the data-generating process is very important as these might indicate **out of control states**
  - Faults in the sensing apparatus
  - Anomalous operating conditions
  - Environmental Changes
Motivating Idea

- Identifying changes in the data-generating process is very important as these might indicate **out of control states**
  - Faults in the sensing apparatus
  - Anomalous operating conditions
  - Environmental Changes
Motivating Idea

- Identifying changes in the data-generating process is very important as these might indicate **out of control states**
  - Faults in the sensing apparatus
  - Anomalous operating conditions
  - Environmental Changes

![Graph showing normal and out of control conditions](image-url)
Motivating Idea (cnt)

1. Fit a model / build a predictor for the time series
   \[ f_{\hat{\theta}}(t) \]

2. For each incoming samples compute the residuals
   \[ e(t) = s(t) - f_{\hat{\theta}}(t) \]

3. Monitor the stationarity of the residuals
Motivating Idea (cnt)

1. Fit a model / build a predictor for the time series
   \[ f_{\theta}(t) \]

2. For each incoming samples compute the residuals
   \[ e(t) = s(t) - f_{\theta}(t) \]

3. Monitor the stationarity of the residuals
Motivating Idea (cnt)

1. Fit a model / build a predictor for the time series
   \[ f_\theta(t) \]

2. For each incoming samples compute the residuals
   \[ e(t) = s(t) - f_\theta(t) \]

3. Monitor the stationarity of the residuals
Unfortunately, it is sometimes difficult to:

- find good model family (i.e., \( f \))
- reliably fit this model (i.e., estimating \( \hat{\theta} \))
Motivating Idea (cnt.)

- Often, signals and time series are redundant and exhibit self-similarity
Often, signals and time series are redundant and exhibit self-similarity.
Motivating Idea (cnt.)

- Often, signals and time series are redundant and exhibit self-similarity

In case of water consumption, the periodicity is due to inhabitants’ customary habits
The process, in its **normal state**, exhibits some structure which is **redundant, self similar, repeated**.
Motivating Idea (cnt.)

- The process, in its **normal state**, exhibits some structure which is **redundant, self similar, repeated**

- **Out of control states** instead exhibit **patterns** that are instead **different** from the normal state.

- Degree of similarity with the normal state changes

```
Normal ~ self-similarity

Out of Control → Change in the similarity with the normal state
```
Our Contribution

- We present a Change Detection Test (CDT) to sequentially monitor time series that uses self-similarity to
  - Characterize normal state of the process
  - Detect any departure from normal condition
Outline

- Self similarity as a powerful prior
- Problem Formulation
- Proposed Solution
  - Change Indicator
  - Search Regions
  - The Algorithm
- Experiments
- Discussion and Conclusions
SELF SIMILARITY

A powerful prior in signal-image processing
Self similarity is a powerful prior

- Texture completion
- Denoising (Regression)
- Inpainting (Reconstruction)

Never used for discriminative purposes in a sequential detection task

Image courtesy of Alessandro Foi
http://www.cs.tut.fi/~foi/
Self similarity as a powerful prior

- Self-similarity is measured patch-wise
- We consider 1D datastreams \( \{s(\tau), \tau = 1, \ldots \} \), \( s(\tau) \in \mathbb{R} \)
- We define a patch centered at \( t \) having size \( \nu \) as
  \[
  s_t = \{s(t - \nu), \ldots, s(t), \ldots, s(t + \nu)\}
  \]
- The distance between two patches is the \( \ell_2 \) norm of their difference
  \[
  \|s_t - s_\tau\|_2 = \sqrt{\sum_{i=-\nu}^{\nu} (s(t+i) - s(\tau+i))^2}
  \]
PROBLEM FORMULATION
Problem Formulation

- Let us assume that a **process** $S$ generates a **datastream** $\{s(\tau), \tau = 1, \ldots\}$, $s(\tau) \in \mathbb{R}$
  - $S$ has to exhibit self similarity in the normal state
- We say that there is a **change** at $T^*$ if $S$ **permanently shifts** from the **normal** state into an **out of control** state.
- We consider out of control states that **modifies self-similarity** of $S$
  - the **patches** from $\{s(\tau), \tau = 1, \ldots, T^*\}$ are **not similar to patches** from $\{s(\tau), \tau = T^* + 1, \ldots\}$.
- **Goal**: Given a **normal training sequence** $TS$, detect changes analyzing, in a sequential and online manner $\{s(\tau), \tau = L + 1, \ldots\}$
PROPOSED SOLUTION

Exploiting self similarity for performing change-detection
The Proposed Solution

- We build a training set for *normal patches*

\[ P = \{s_t, t = \nu, ..., M - \nu\} \]
The Proposed Solution

- We build a training set for *normal patches*

\[ P = \{s_t, t = \nu, \ldots, M - \nu\} \]
The Proposed Solution

- We build a training set for *normal patches*
  \[ \mathbf{P} = \{\mathbf{s}_t, t = \nu, \ldots, M - \nu\} \]

- Intuition:
  \[
  \begin{cases}
  \exists \mathbf{s}_u \in \mathbf{P} \text{ similar to } \mathbf{s}_t, \forall t < T^* & \text{Normal} \\
  \not\exists \mathbf{s}_u \in \mathbf{P} \text{ similar to } \mathbf{s}_t, \forall t \geq T^* & \text{Out of Control}
  \end{cases}
  \]
The Proposed Solution

- We build a training set for normal patches
  \[ P = \{ s_t, t = \nu, ..., M - \nu \} \]
  
- Intuition:

\[
\begin{align*}
\exists s_u \in P \text{ similar to } s_t, \forall t < T^* \\
\not\exists s_u \in P \text{ similar to } s_t, \forall t \geq T^*
\end{align*}
\]
The Change Indicator

- We need to construct a **change indicator** \( x(t) \) to quantitatively assess our intuition

- We expect the change indicator \( x(t) \) to satisfy
  - \( \{x(t), t < T^*\} \) should be i.i.d. realizations of an **unknown random variable**
  - \( \{x(t), t \geq T^*\} \) should come from a **different distribution**, not necessarily being i.i.d.

- Out of **control states** can be detected as changes in the distribution of \( x \)
  - We can use any statistical process control technique
The Change Indicator (cnt.)

- The compute the change indicator $x(t)$ we first identify the most similar patch in $P$ to $s_t$.

- We define $\pi(\cdot)$ as the map that associate to $t$ the location $\pi(t)$ of the patch $P$ of that is most similar to $s_t$

$$\pi(t) = \arg\min_{\tau=\nu, \ldots, M-\nu} ||s_t - s_\tau||_2$$

the values of $\pi(\cdot)$ can be com

- $x(t)$ is the difference between the centers of $s_t$ and $s_{\pi(t)}$

$$x(t) = s(t) - s(\pi(t))$$
The Change Indicator (cnt.)

- The compute the change indicator $x(t)$ we first identify the most similar patch in $\mathbf{P}$ to $s_t$.

- We define $\pi(\cdot)$ as the map that associate to $t$ the location $\pi(t)$ of the patch $\mathbf{P}$ of that is most similar to $s_t$

$$
\pi(t) = \arg\min_{\tau=\nu,\ldots,\nu,M-\nu} ||s_t - s_{\tau}||_2
$$

- The values of $\pi(\cdot)$ can be computed.

- $x(t)$ is the difference between the centers of $s_t$ and $s_{\pi(t)}$

$$
x(t) = s(t) - s(\pi(t))
$$
In Ideal Conditions

- Assume perfect matches in normal conditions, i.e., \( s_t \) and \( s_{\pi(t)} \) differ only because of noise.

- Then, \( \forall t < T^* \)

\[
x(t) = s(t) - s(\pi(t)) = \eta
\]

i.i.d random variable and \( E[\eta] = 0 \)

- While \( \forall t > T^* \), we do not expect perfect matches: some bias appears in \( x(t) \), namely \( E[x(t)] \neq 0 \)

- In this case, it is possible to detect changes in \( x(t) \) by means of any sequential CDT.
In the Real Life

- In the real life, perfect matches are rare
  - Patches do not differ only because of noise
  - Noise affects also the association function $\pi(\cdot)$
- However, there is an experimental evidence that patch similarity well correlates with the similarity between their central pixels
  - This is the idea behind *Non Local Means filter* [Buades et al 2005], which introduced a well established paradigm in signal/image processing

---

Often, self similarity is due to **periodic or cyclic** nature of the phenomenon under monitoring.
Often, self similarity is due to **periodic or cyclic** nature of the phenomenon under monitoring.

Search for similar patches should be constrained to the same time instants across the periods:
- This determines what «out of control states» are
- This improves computational complexity
The function $\pi(\cdot)$ is thus defined

$$\pi(t) = \arg\min_{\tau \in \mathcal{R}_{\phi,t,\delta}} \| s_t - s_\tau \|_2$$

Being, $\mathcal{R}_{\phi,t,\delta} = \bigcup_i \{ \tau, \text{ s.t. } |t_0 + i\phi - \tau| < \delta \}$
An example
An example
An example
An example
An example
An example
An example
The CDT on the change indicators

- A CDT can be used to detect online and sequentially, changes in the distribution of $x$.
- CDTs often require a training sequence containing values of $x$ that have been computed when $S$ is in the normal state.
- Change indicators are monitored by a CDT
  - We used the ICI-based CDT [Alippi et al 2010]

The Algorithm: The Training Phase

- CDTs often require a training sequence of values of $x$, computed when $S$ is in the normal state
- Change indicators are monitored by a CDT
  - We used the ICI-based CDT [Alippi et al 2010]
- The initial training set $TS$ is divided in two parts
The CDT Training Set

- data
- training set
- current patch
- current point
- centers in Search Region
- closest Patch

Training set for CDT
1- **input**: \( \{s(\tau), \tau = 1, \ldots, L\}, \nu, \delta, \phi, M \)
2- define \( \mathbf{P} \) from \( TS = \{s(\tau), \tau = 1, \ldots, M\} \) as in (3),
3- **for** \( (t = M + 1; t \leq L; t++) \) **do**
4- extract the patch \( s_t \) as in (1),
5- define the search region \( R_{t,\phi,\delta} \) as in (8),
6- compute the patch most similar to \( s_t \) in \( R_{t,\phi,\delta} \), (7)
7- compute the change indicator \( x(t) \) as in (6),
end
8- configure the CDT on \( \{x(t), t = M + 1, \ldots, L\} \),
9- wait for the next \( \nu \) samples,
10- **while** \( (s(t+\nu) \text{ arrives}) \) **do**
11- extract the patch \( s_t \) as in (1),
12- define the search region \( R_{t,\phi,\delta} \) as in (8),
13- compute the patch most similar to \( s_t \) in \( R_{t,\phi,\delta} \), (7)
14- compute the change indicator \( x(t) \) as in (6),
15- **if** \( \text{CDT}(\{x(\tau), \tau = M, \ldots, t\}) = l \) **then**
16- detect a structural change in \( S \) at \( \hat{T} = t \).
17- **return.**
18- \( t = t + 1; \)
end
The Algorithm

**Training Phase**

Compute the change indicators over normal data

1. **input:** \( \{s(\tau), \tau = 1, \ldots, L\}, \nu, \delta, \phi, M \)
2. define \(P\) from \(TS = \{s(\tau), \tau = 1, \ldots, M\}\) as in (3),
3. **for** \((t = M + 1; t \leq L; t++)\) **do**
   4. extract the patch \(s_t\) as in (1),
   5. define the search region \(R_{t,\phi,\delta}\) as in (8),
   6. compute the patch most similar to \(s_t\) in \(R_{t,\phi,\delta}\), (7)
   7. compute the change indicator \(x(t)\) as in (6),
4. **end**
8. configure the CDT on \(\{x(t), t = M + 1, \ldots, L\}\),
9. wait for the next \(\nu\) samples,
10. **while** \((s(t + \nu)\) arrives) **do**
    11. extract the patch \(s_t\) as in (1),
    12. define the search region \(R_{t,\phi,\delta}\) as in (8),
    13. compute the patch most similar to \(s_t\) in \(R_{t,\phi,\delta}\), (7)
    14. compute the change indicator \(x(t)\) as in (6),
    15. **if** \(\text{CDT}(\{x(\tau), \tau = M, \ldots, t\}) == 1\) **then**
        16. detect a structural change in \(S\) at \(\hat{T} = t\).
        17. **return.**
    18. \(t = t + 1;\)
10. **end**
Configure the ICI-based CDT on these change indicators

1- **input**: \( \{s(\tau), \tau = 1, \ldots, L\}, \nu, \delta, \phi, M \)
2- define \( P \) from \( TS = \{s(\tau), \tau = 1, \ldots, M\} \) as in (3),
3- **for** \( (t = M + 1; t \leq L; t++) \) **do**
4- extract the patch \( s_t \) as in (1),
5- define the search region \( R_{t,\phi,\delta} \) as in (8),
6- compute the patch most similar to \( s_t \) in \( R_{t,\phi,\delta} \), (7),
7- compute the change indicator \( x(t) \) as in (6),
8- **end**
9- configure the CDT on \( \{x(t), t = M + 1, \ldots, L\} \),
10- wait for the next \( \nu \) samples,
11- **while** (\( s(t + \nu) \) arrives) **do**
12- extract the patch \( s_t \) as in (1),
13- define the search region \( R_{t,\phi,\delta} \) as in (8),
14- compute the patch most similar to \( s_t \) in \( R_{t,\phi,\delta} \), (7),
15- compute the change indicator \( x(t) \) as in (6),
16- **if** \( \text{CDT}(\{x(\tau), \tau = M, \ldots, t\}) = = 1 \) **then**
17- detect a structural change in \( S \) at \( \hat{T} = t \).
18- return.
19- **end**
20- \( t = t + 1; \)
21- **end**
1- **input**: \( \{ s(\tau), \tau = 1, \ldots, L \} \), \( \nu, \delta, \phi, M \)
2- define \( P \) from \( TS = \{ s(\tau), \tau = 1, \ldots, M \} \) as in (3),
3- **for** \( t = M + 1; t \leq L; t++ \) **do**
4- extract the patch \( s_t \) as in (1),
5- define the search region \( R_{t,\phi,\delta} \) as in (8),
6- compute the patch most similar to \( s_t \) in \( R_{t,\phi,\delta} \), (7)
7- compute the change indicator \( x(t) \) as in (6),
**end**
8- configure the CDT on \( \{ x(t), t = M + 1, \ldots, L \} \),
9- wait for the next \( \nu \) samples,
10- **while** \( s(t + \nu) \) **arrives** **do**
11- extract the patch \( s_t \) as in (1),
12- define the search region \( R_{t,\phi,\delta} \) as in (8),
13- compute the patch most similar to \( s_t \) in \( R_{t,\phi,\delta} \), (7)
14- compute the change indicator \( x(t) \) as in (6),
15- **if** \( CDT(\{ x(\tau), \tau = M, \ldots, t \}) == 1 \) **then**
16- detect a structural change in \( S \) at \( \hat{T} = t \).
17- **return**.
18- \( t = t + 1; \)
**end**

## Operational Life

Crop a patch around \( s(t) \)
The Algorithm

1- **input**: \( \{s(\tau), \tau = 1, \ldots, L\}, \nu, \delta, \phi, M \)
2- define \( P \) from \( TS = \{s(\tau), \tau = 1, \ldots, M\} \) as in (3),
3- for \( t = M + 1; t \leq L; t++ \) do
   4- extract the patch \( s_t \) as in (1),
   5- define the search region \( R_{t, \phi, \delta} \) as in (8),
   6- compute the patch most similar to \( s_t \) in \( R_{t, \phi, \delta} \), (7),
   7- compute the change indicator \( x(t) \) as in (6),
end
8- configure the CDT on \( \{x(t), t = M + 1, \ldots, L\} \),
9- wait for the next \( \nu \) samples,
10- while \( s(t + \nu) \) arrives do
    11- extract the patch \( s_t \) as in (1),
    12- define the search region \( R_{t, \phi, \delta} \) as in (8),
    13- compute the patch most similar to \( s_t \) in \( R_{t, \phi, \delta} \), (7),
    14- compute the change indicator \( x(t) \) as in (6),
    15- if \( \text{CDT}(\{x(\tau), \tau = M, \ldots, t\}) = 1 \) then
        16- detect a structural change in \( S \) at \( \hat{T} = t \).
        return.
    end
18- \( t = t + 1; \)
end

Compute the change indicator \( x(t) \)
The Algorithm

1- \textbf{input:} \{\(s(\tau), \tau = 1, \ldots, L\)}, \(\nu, \delta, \phi, M\)
2- define \(P\) from \(TS = \{s(\tau), \tau = 1, \ldots, M\}\) as in (3),
3- \textbf{for} \(t = M + 1; t \leq L; t++\) \textbf{do}
4- extract the patch \(s_t\) as in (1),
5- define the search region \(R_{t,\phi,\delta}\) as in (8),
6- compute the patch most similar to \(s_t\) in \(R_{t,\phi,\delta}\), (7),
7- compute the change indicator \(x(t)\) as in (6),
8- configure the CDT on \(\{x(t), t = M + 1, \ldots, L\}\),
9- wait for the next \(\nu\) samples,
10- \textbf{while} \((s(t + \nu)\) arrives) \textbf{do}
11- extract the patch \(s_t\) as in (1),
12- define the search region \(R_{t,\phi,\delta}\) as in (8),
13- compute the patch most similar to \(s_t\) in \(R_{t,\phi,\delta}\), (7),
14- compute the change indicator \(x(t)\) as in (6),
15- \textbf{if} \((CDT(\{x(\tau), \tau = M, \ldots, t\}) == 1)\) \textbf{then}
16- detect a structural change in \(S\) at \(\hat{T} = t\).
17- \textbf{return}.
18- \(t = t + 1;\)
end
end

Run the CDT until a change is detected
EXPERIMENTS
The Data Set

- Flow measured in Barcelona Water Distribution Networks
  - Measurements from different DMA inlets
  - One measure every 10 minutes, daily period
The DataSet

- Flow measured in Barcelona Water Distribution Networks
  - Measurements from different DMA inlets
  - One measure every 10 minutes, daily period
- 10 sequences synthetically adding a change after 41 days
  - Offset: \( s(t) = s(t) + o, \ t > T^*, \ o \in \{0.25a, 0.5a\} \)
**The DataSet**

- Flow measured in Barcelona Water Distribution Networks
  - Measurements from different DMA inlets
  - One measure every 10 minutes, daily period
- 10 sequences synthetically adding a change after 41 days
  - **Offset:** \( s(t) = s(t) + o, \ t > T^*, \ o \in \{0.25a, 0.5a\} \)

![Graph showing data pattern over time](attachment:image.png)
The DataSet

- Flow measured in Barcelona Water Distribution Networks
  - Measurements from different DMA inlets
  - One measure every 10 minutes, daily period
- 10 sequences synthetically adding a change after 41 days
  - **Offset**: \( s(t) = s(t) + o, t > T^*, o \in \{0.25a, 0.5a\} \)
  - **Sensor Degradation**: \( s(t) = s(t) + \eta(t), t > T^*, \eta \sim N(0, \sigma^2) \)
The DataSet

- Flow measured in Barcelona Water Distribution Networks
  - Measurements from different DMA inlets
  - One measure every 10 minutes, daily period
- 10 sequences synthetically adding a change after 41 days
  - **Offset**: $s(t) = s(t) + o, t > T^*, o \in \{0.25a, 0.5a\}$
  - **Sensor Degradation**: $s(t) = s(t) + \eta(t), t > T^*, \eta \sim N(0, \sigma^2)$
  - **Source Change** $s(t) = s_1(t), t > T^*$ from a different DMA

![Graph showing flow measurements over time]
The Data Set

- Flow measured in Barcelona Water Distribution Networks
  - Measurements from different DMA inlets
  - One measure every 10 minutes, daily period
- 10 sequences synthetically adding a change after 41 days
  - **Offset:** \( s(t) = s(t) + o, \quad t > T^* \), \( o \in \{0.25a, 0.5a\} \)
  - **Sensor Degradation:** \( s(t) = s(t) + \eta(t), \quad t > T^* \), \( \eta \sim N(0, \sigma^2) \)
  - **Source Change** \( s(t) = s_1(t), \quad t > T^* \) from a different DMA
  - **Stack-at:** \( s(t) = k, \quad t > T^* \)
The Considered CDTs

- **Residual-based**: a predictive model $f_{\hat{\theta}}$ of a nonlinear ARX (wavelet network) is used to compute

$$r(t) = s(t) - f_{\hat{\theta}}(t)$$
The Considered CDTs

- **Residual-based**: a predictive model $f_{\theta}$ of a nonlinear ARX (wavelet network) is used to compute

$$r(t) = s(t) - f_{\theta}(t)$$

- **Template-based**: compute the difference w.r.t. template, i.e., the average flow profile in the $n$ periods in the TS

$$p(t) = s(t) - \frac{1}{n} \sum_{i=1}^{n} s(t_0 + i\phi)$$
The Considered CDTs

- **Residual-based**: a predictive model $f_{\theta}$ of a nonlinear ARX (wavelet network) is used to compute

$$r(t) = s(t) - f_{\theta}(t)$$

- **Template-based**: compute the difference w.r.t. template, i.e., the average flow profile in the $n$ periods in the TS

$$p(t) = s(t) - \frac{1}{n} \sum_{i=1}^{n} s(t_0 + i\phi)$$

**Self-similarity**: the proposed solution, $\nu = 5$, $\delta = 5$

**Details:**

2 weeks of recordings used for building $P$ / model fitting / template estimation, 400 samples for CDT configuration.
Change indicators in normal conditions

- The autocorrelation of the considered change indicators in normal conditions.
Change indicators in normal conditions

- The autocorrelation of the considered change indicators in normal conditions
Change indicators in normal conditions

- The autocorrelation of the considered change indicators in normal conditions

**Autocorrelation of Template-based Change Index p in (10)**

- Average over 10 time series
- Individual values

![Graph showing autocorrelation](image)
## Change Detection Performance

- **FPR**: False Positive Rate
- **FNR**: False Negative Rate
- **DD**: Expected Detection Delay

<table>
<thead>
<tr>
<th></th>
<th>Self-Similarity based</th>
<th>Residuals-based</th>
<th>Template-based</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FPR</td>
<td>FNR</td>
<td>DD</td>
</tr>
<tr>
<td>offset 0.5</td>
<td>0.1</td>
<td>0.0</td>
<td>156.4</td>
</tr>
<tr>
<td>offset 0.25</td>
<td>0.1</td>
<td>0.0</td>
<td>914.2</td>
</tr>
<tr>
<td>sensor degradation 0.5</td>
<td>0.1</td>
<td>0.0</td>
<td>174.2</td>
</tr>
<tr>
<td>sensor degradation 0.25</td>
<td>0.1</td>
<td>0.0</td>
<td>336.4</td>
</tr>
<tr>
<td>source change</td>
<td>0.1</td>
<td>0.0</td>
<td>103.1</td>
</tr>
<tr>
<td>stack-at</td>
<td>0.1</td>
<td>0.0</td>
<td>169.8</td>
</tr>
</tbody>
</table>
Offset of +50% the average flow value

Time series affected by an offset at $T^* = 5904$

Change indicator $x$ in (6) $\hat{T} = 6056$

Change indicator $r$ in (9) $\hat{T} = 2416$

Change indicator $p$ in (10) $\hat{T} = 6876$
Time series affected by a source change at $T^* = 5904$

Change indicator $x$ in (6) $\hat{T} = 5916$

Change indicator $r$ in (9) $\hat{T} = 6556$

Change indicator $p$ in (10) $\hat{T} = 5936$
Offset of +25% the mean flow (False Positive)

Time series affected by an offset at $T^* = 5904$

Change indicator $x$ in (6) $\hat{T} = 4516$

Change indicator $r$ in (9) $\hat{T} = 8256$

Change indicator $p$ in (10) $\hat{T} = 6576$
CONCLUDING REMARKS
Concluding Remarks

- Self similarity seems a promising approach for detecting changes in the structure of a self-similar datastream
  - Detection performance and autocorrelation show that $x$ is very good at assessing self similarity
  - Detection performance indicates that $x$ reliably reacts to changes

- Ongoing Works
  - Investigating different change indicators for assessing self similarity.
  - Exploiting self similarity in a collaborative manner (multichannel observations)
  - Self similarity when data are not periodic
  - Automatic criteria to identify the best patch size
Many thanks to Prof. Vicenç Puig from Universitat Politécnica de Catalunya, Barcelona, Spain, for providing us the datasets and for the meaningful discussions.

This work was supported by the FP7 EU project i-Sense, *Making Sense of Nonsense*, Contract No: INSFO-ICT-270428.
Questions?

Codes will be soon available for download at
http://home.deib.polimi.it/boracchi/Projects/