Change detection in multi-dimensional datasets and time series

Andrea De Simone
andrea.desimone@sissa.it

Univ. Camerino, 2019-02-26

Outline

1 Two-Sample Test: Intro & Motivation
2 Nearest Neighbors Two-Sample Test (NN2ST)
3 Gaussian Examples
4 Outlook: Time Series Data
Two-Sample Test

Two sets:

**Trial:** \( \mathcal{T} \equiv \{ x_1, \ldots, x_{N_T} \} \overset{\text{iid}}{\sim} p_T \), \( x_i, x'_i \in \mathbb{R}^D \)

**Benchmark:** \( \mathcal{B} \equiv \{ x'_1, \ldots, x'_{N_B} \} \overset{\text{iid}}{\sim} p_B \), \( p_B, p_T \) unknown

Andrea De Simone
Univ. Camerino, 2019-02-26
Two-Sample Test

Two sets:

Trial: \( \mathcal{T} \equiv \{ x_1, \ldots, x_{N_T} \} \overset{iid}{\sim} p_T, \quad x_i, x'_i \in \mathbb{R}^D \)

Benchmark: \( \mathcal{B} \equiv \{ x'_1, \ldots, x'_{N_B} \} \overset{iid}{\sim} p_B. \quad p_B, p_T \text{ unknown} \)

« Are \( \mathcal{B}, \mathcal{T} \) drawn from the same probability distribution? »

easy...
Two-Sample Test

Two sets:

Trial: \( \mathcal{T} \equiv \{ x_1, \ldots, x_{N_T} \} \overset{iid}{\sim} p_T \), \( x_i, x'_i \in \mathbb{R}^D \)

Benchmark: \( \mathcal{B} \equiv \{ x'_1, \ldots, x'_{N_B} \} \overset{iid}{\sim} p_B \) \( p_B, p_T \) unknown

« Are \( \mathcal{B}, \mathcal{T} \) drawn from the same probability distribution? »

Benchmark and Trial samples

... hard
Two-Sample Test

Why is it important?

• detect departures from benchmark
• find anomalous points (outliers)
• check if observed data are compatible with expectations
• detect changes in underlying distributions
• real-time detect events/shifts in time series
Two-Sample Test

Desiderata for a statistical test

(1) model-independent
   no assumption about underlying physical model to interpret data
      \[\rightarrow\] more general

(2) non-parametric
   compare two samples as a whole (not just their means, etc.)
      \[\rightarrow\] fewer assumptions, no max likelihood estim.

(3) un-binned
   high-dim feature space partitioned without rectangular bins
      \[\rightarrow\] retain full multi-dim info of data
Two-Sample Test

Recipe

1. **Density Estimator**
   → reconstruct PDF from samples

2. **Test Statistic (TS)**
   → “measure distance” between PDFs

3. **TS distribution**
   → associate probabilities to TS
   under null hypothesis $H_0 : p_B = p_T$

4. **p-value**
   → if $p < \alpha$ then reject $H_0$

Let’s build the **Nearest Neighbors Two-Sample Test (NN2ST)**
1. Density Estimator

Divide space in square bins?

- easy
- can use simple statistics (e.g. \( \chi^2 \))
- hard/slow/impossible in high-\( D \)

Need un-binned, multi-variate approach

Find PDF estimators \( \hat{p}_B, \hat{p}_T \), e.g. based on density of points

\[
\hat{p}_{B,T}(\mathbf{x}) = \frac{\rho_{B,T}(\mathbf{x})}{N_{B,T}}
\]

Nearest Neighbors!

[Schilling 1986, Henze 1988]
[Wang et al. 2005-2006, Perez-Cruz 2008]
1. Density Estimator

- Fix integer $K$.
- Choose query point $x_j$ in $\mathcal{T}$ and draw it in $\mathcal{B}$. 

\[ \hat{p}_B(x_j) = \frac{K}{N_B} \frac{1}{\omega(D r_{j,B})} \]
\[ \hat{p}_T(x_j) = \frac{K}{N_T} \frac{1}{\omega(D r_{j,T})} \]
1. Density Estimator

- Fix integer $K$.
- Choose query point $x_j$ in $\mathcal{T}$ and draw it in $\mathcal{B}$.
- Find the distance $r_{j,B}$ of the $K^{th}$-NN of $x_j$ in $\mathcal{B}$.
1. Density Estimator

- Fix integer $K$.
- Choose query point $x_j$ in $\mathcal{T}$ and draw it in $\mathcal{B}$.
- Find the distance $r_{j,B}$ of the $K^{th}$-NN of $x_j$ in $\mathcal{B}$.
- Find the distance $r_{j,T}$ of the $K^{th}$-NN of $x_j$ in $\mathcal{T}$.

\[ \hat{p}_B(x_j) = \frac{K}{N_B} \frac{1}{\omega_D r_{j,B}} \]
\[ \hat{p}_T(x_j) = \frac{K}{N_T - 1} \frac{1}{\omega_D r_{j,T}} \]
1. Density Estimator

- Fix integer $K$.
- Choose query point $x_j$ in $\mathcal{T}$ and draw it in $\mathcal{B}$.
- Find the distance $r_{j,B}$ of the $K$th-NN of $x_j$ in $\mathcal{B}$.
- Find the distance $r_{j,T}$ of the $K$th-NN of $x_j$ in $\mathcal{T}$.
- Estimate PDFs:

  \[
  \hat{p}_B(x_j) = \frac{K}{N_B} \frac{1}{\omega_D r_{j,B}^D}
  \]

  \[
  \hat{p}_T(x_j) = \frac{K}{N_T - 1} \frac{1}{\omega_D r_{j,T}^D}
  \]
2. Test Statistic

- Measure the “distance” between 2 PDFs
- Define **Test Statistic** (to detect under-/over-densities)

\[
TS(\mathcal{T}) \equiv \frac{1}{N_T} \sum_{j=1}^{N_T} \log \frac{\hat{p}_T(x_j)}{\hat{p}_B(x_j)}
\]

- Form NN-estimated PDFs:

\[
TS(\mathcal{T}) = \frac{D}{N_T} \sum_{j=1}^{N_T} \log \frac{r_{j,B}}{r_{j,T}} + \log \frac{N_B}{N_T - 1}
\]

- Related to *Kullback-Leibler* divergence as: \( TS(\mathcal{T}) = \hat{D}_{KL}(\hat{p}_T||\hat{p}_B) \)

\[
D_{KL}(p||q) \equiv \int_{\mathbb{R}^D} p(x) \log \frac{p(x)}{q(x)} \, dx
\]

- **Theorem:**
  this estimator converges to \( D_{KL}(p_B||p_T) \), in the large sample limit

[Wang et al. – 2005, 2006]
3. Test Statistic Distribution

How is TS distributed? **Permutation test!**

Assume $p_B = p_T$. Union set $\mathcal{U} = \mathcal{T} \cup \mathcal{B}$.

Repeat many times.

Distribution of TS under $H_0$: $f(TS|H_0) \leftarrow \{TS_n\}$ [asymptotically normal with zero mean]
4. \( p \)-value

- Find \( \hat{\mu}, \hat{\sigma} \): mean, variance of \( f(TS|H_0) \)

- Standardize the TS:

\[
TS \rightarrow TS' \equiv \frac{TS - \hat{\mu}}{\hat{\sigma}}
\]

- \( TS' \) distributed according to \( f'(TS'|H_0) = \hat{\sigma} f(\hat{\mu} + \hat{\sigma} TS'|H_0) \)

- Two-sided \( p \)-value

\[
p = 2 \int_{|TS_{obs}|}^{\infty} f'(TS'|H_0) dTS'
\]
\textbf{NN2ST: Summary}

\textbf{INPUT:}

\begin{itemize}
    \item Trial sample: $\mathcal{T} \equiv \{x_1, \ldots, x_{N_T}\} \overset{iid}{\sim} p_T,$
    \item Benchmark sample: $\mathcal{B} \equiv \{x'_1, \ldots, x'_{N_B}\} \overset{iid}{\sim} p_B$
    \item $K$: number of nearest neighbors
    \item $N_{\text{perm}}$: number of permutations
    \item $x_i, x'_i \in \mathbb{R}^D$
    \item $p_B, p_T$ unknown
\end{itemize}

\textbf{OUTPUT:}

$p$-value of the null hypothesis $H_0 : p_B = p_T$

[check compatibility between 2 samples]
[detect changes in underlying distributions]
NN2ST: Summary

- K-NN density ratio estimation
- Test Statistic
- permutation test
- p value
- TS distribution

Python code:
[github.com/de-simone/NN2ST
NN2ST: Summary

- general, model-independent
- solid math foundations
- fast, no optimization
- sensitive to unspecified signals
- need to run for each sample pair
- permutation test is bottleneck
NN2ST on Gaussian Samples

Random samples from $D$-dimensional Gaussians

\[
p_B = \mathcal{N}(\mu_B, \Sigma_B), \quad p_T = \mathcal{N}(\mu_T, \Sigma_T).
\]

\[
D = 2, \quad \mu_B = \begin{pmatrix} 1.0 \\ 1.0 \end{pmatrix}, \quad \mu_T = \begin{pmatrix} 1.2 \\ 1.2 \end{pmatrix},
\]

\[
\Sigma_B = \Sigma_T = I_2.
\]

Convergence to exact KL divergence
NN2ST on Gaussian Samples

<table>
<thead>
<tr>
<th>Dataset</th>
<th>$\mu$</th>
<th>$\Sigma$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$B$</td>
<td>$1_D$</td>
<td>$I_D$</td>
</tr>
<tr>
<td>$T_{G0}$</td>
<td>$1_D$</td>
<td>$I_D$</td>
</tr>
<tr>
<td>$T_{G1}$</td>
<td>$1.12_D$</td>
<td>$I_D$</td>
</tr>
<tr>
<td>$T_{G2}$</td>
<td>$1_D$</td>
<td>$\begin{pmatrix} 0.95 &amp; 0.1 &amp; 0 \ 0.1 &amp; 0.8 &amp; 0 \ 0 &amp; 0 &amp; I_{D-2} \end{pmatrix}$</td>
</tr>
<tr>
<td>$T_{G3}$</td>
<td>$1.15_D$</td>
<td>$I_D$</td>
</tr>
</tbody>
</table>

$N_B = N_T = 20000$
$K = 5$
$N_{\text{perm}} = 1000$

more data, more power

higher $D$, more power
Real-time detection of changes in data streams: variation in underlying mechanism generating data.

\( \mathcal{T}, \mathcal{B} \) samples: windows of time series data, ending at discrete times \( t, t' \)

\[
\mathcal{T}_t = \{x_{t-N+1}, \ldots, x_t\}, \\
\mathcal{B}_{t'} = \{x_{t'-N+1}, \ldots, x_{t'}\}, \quad (N_B = N_T \equiv N).
\]

Trial window sliding forward with time. Benchmark window anchored or rolling.

- anchored \( \mathcal{B} \) window: \( t' = N \rightarrow \mathcal{B}_{t'} = \{x_1, \ldots, x_N\} \)
  Captures cumulative changes over time.

- adjacent windows: \( t' = t - N \rightarrow \mathcal{B}_{t'} = \{x_{t-2N+1}, \ldots, x_{t-N}\} \)
  Captures “rate of change” at current time.
Outlook: time series data

Feature space can be high-dimensional: prices (OHLC), prices of related markets, indicators, volumes, ...

Reduce false alarms with persistence factor $\gamma (\sim 1\%)$.

H$_0$ rejected $\gamma \cdot N$ times in a row $\rightarrow$ detected change in market conditions.
adjacent vs. anchored windows
Feature space can be high-dimensional: prices (OHLC), prices of related markets, indicators, volumes, . . .

Reduce false alarms with persistence factor $\gamma$ ($\sim 1$)\%.

$H_0$ rejected $\gamma \cdot N$ times in a row

$\rightarrow$ detected change in market conditions
Take-Home Messages

(1) Proposed a new statistical test: NN2ST

(2) Model-independent and suitable for high-$D$ data

(3) Excellent results on static datasets

(4) Promising applications for change detection in time series data