Hidden Markov Anomaly Detection

Nico Görnitz, Mikio Braun, Marius Kloft

Berlin Institute of Technology & Humboldt University Berlin
I CAN WORK WITH THE MEANS.

BUT I'D RATHER PARTY WITH THE OUTLIERS.
Supervised vs Unsupervised

Anomaly Detection

Hidden Markov Anomaly Detection Results

Software

Train

Test

novel anomalies

AUC\([0,0.01]\) [in %]

% of labeled data

unsupervised

supervised

Train Test novel anomalies

Nico ()
One-class Classification

The OC-SVM employs linear models $f_{w,\rho}(x) = \langle w, \phi(x) \rangle - \rho$. It subsequently separates a fraction of $1 - \nu$ many inputs from the origin with maximum margin:

$$\max_{w,\rho,\xi \geq 0} \|w\|^2 - \rho + \frac{1}{\nu n} \sum_{i=1}^{n} \xi_i$$

(OC-SVM)

s.t. $\xi_i \geq -f_{w,\rho}(x_i)$ $\forall i = 1, \ldots, n.$
Latent Anomaly Detection employs models of the form:

$$f_{\mathbf{w}, \rho}(x) = \max_{z \in \mathcal{Z}} \langle \mathbf{w}, \Psi(x, z) \rangle + \delta(z) - \rho$$
Latent Anomaly Detection

\[ f_{\psi} (x) = \max_{z \in \mathcal{Z}} \left( \langle w, \psi(x, z) \rangle + \delta(z) \right) - \rho \]

- Dual
- Properties
- Learning Theory

Latent Anomaly Detection Framework

Fixing the loss

Latent One-class SVM

Fixing the joint feature map \( \psi \)

Hidden Markov Anomaly Detection (HMAD)
Latent Anomaly Detection

Given a monotonically non-decreasing loss function \( l: \mathbb{R} \to \mathbb{R} \), minimize, with respect to \( \mathbf{w} \in \mathcal{H} \) and \( \rho \in \mathbb{R} \),

\[
\frac{1}{2} \| \mathbf{w} \|^2 - \rho + \frac{1}{\nu n} \sum_{i=1}^{n} l(\rho - \max_{z \in \mathcal{Z}} (\langle \mathbf{w}, \Psi(x_i, z) \rangle + \delta(z))) .
\]

Given the monotonically non-decreasing hinge loss function \( l: \mathbb{R} \to \mathbb{R}, l(t) = \max(0, t) \), we arrive at the Latent One-class SVM.
Generalization Bound

Let \( l : \mathbb{R} \rightarrow \mathbb{R} \) be a non-negative and \( L \)-Lipschitz continuous loss function. Denote 
\[
A := \max_{z \in \mathcal{Z}} |\delta(z)| \quad \text{and} \quad B := \max_{x \in \mathcal{X}, z \in \mathcal{Z}} \|\Psi(x, z)\|.
\]
With probability at least \( 1 - \epsilon \) over the draw of the sample, the generalization error is bounded as:
\[
E l(\hat{f}) - E l(f^*) \leq 8L \frac{1 + A + BC |\mathcal{Z}|}{\sqrt{n}} + L(1 + A + BC) \sqrt{\frac{2 \log(2/\epsilon)}{n}}.
\]
Hidden Markov Joint Feature Map

Given a feature map \( \phi: X \rightarrow \mathcal{F} \), define the hidden Markov joint feature map \( \Psi: X \times Z \rightarrow \mathcal{H} \) as

\[
\Psi(x, z) = \left( (\sum_{t=2}^{T} 1[z^t = i \land z^{t-1} = j])_{i,j \in Y}, \left( \sum_{t=1}^{T} 1[z^t = i] \phi(x^t) \right)_{i \in Y} \right).
\]

so the scoring function becomes

\[
\langle w, \Psi(x, z) \rangle = \sum_{t=2}^{T} \sum_{i,j \in Y} 1[z^t = i \land z^{t-1} = j] w^{\text{trans}}_{i,j} + \sum_{t=1}^{T} \sum_{i \in Y} 1[z^t = i] \langle w^\text{em}_i, \phi(x^t) \rangle,
\]

\[
Nico ()
\]

Hidden Markov Anomaly Detection

Results

Software
Optimization

input data $x_1, \ldots, x_n$
put $t = 0$ and initialize $w^t$
repeat
  $t := t + 1$
  for $i = 1, \ldots, n$ do
    $z_i^t := \text{argmax}_{z \in Z} \langle w^{t-1}, \Psi(x_i, z) \rangle + \delta(z)$
    (i.e. use Viterbi algorithm)
  end for
let $(w^t, \rho^t)$ be the optimal arguments when solving OC-SVM with $\phi(x_i) := \Psi(x_i, z_i^t)$
until $\forall i = 1, \ldots, n : z_i^t = z_i^{t-1}$
Return optimal model parameters $w := w^t, \rho = \rho^t$
and $z_i := z_i^t \ \forall i = 1, \ldots, N$
Toy Data - Setting

![Toy Data - Setting](image_url)

- Noisy observations
- True state sequence

Percentage of disorganization

Sequence position
Toy Data – Accuracy I

![Graph showing detection accuracy for different methods vs percentage of disorganization.]

- Bayes (Linear)
- HMAD
- OC-SVM (RBF 1.0)
- OC-SVM (Hist 8)
- OC-SVM (Linear)
Toy Data – Accuracy II

![Graph showing detection accuracy for different methods and varying percentages of anomalous data.](image)

Detection accuracy [in AUC] vs. Percentage of anomalous data.
Toy Data – Accuracy III

Detection accuracy [in AUC]

- HMAD
- Fisher Kernel Upper Bound
- Fisher Kernel Lower Bound

Percentage of anomalous data

- 2.5%
- 5%
- 10%
- 15%
- 20%
- 30%
Anomaly Detection Hidden Markov Anomaly Detection Results Software

Toy Data – Runtime

![Graph showing runtime comparison for different methods. The x-axis represents the number of training examples ranging from 100 to 1000. The y-axis represents time in seconds, with values ranging from 10^-5 to 10^0. The methods compared include Bayes (Linear), HMAD, OC-SVM (RBF 1.0), OC-SVM (Hist 8), and OC-SVM (Linear). The graph shows how the time increases with the number of training examples for each method.]
Toy Data – Stability

Detection accuracy [in AUC]

Number of hidden states

HMAD

Nico ()
Procaryotic Gene Finding - Setting
Procaryotic Gene Finding – Results

Detection accuracy [in AUC]

- OC-SVM Spectrum (1)
- OC-SVM Spectrum (2)
- OC-SVM Spectrum (3)
- OC-SVM Spectrum (FS)
- HMAD (FS)
- HMAD

Percentage of anomalous data

2.5% 5% 10% 15% 20% 30%
Wind Turbines - Results

![Graph showing detection accuracy for different methods and percentage of anomalous data.]

- Detection accuracy [in AUC]:
  - OC-SVM (Hist 4)
  - OC-SVM (Hist 8)
  - OC-SVM (Hist 16)
  - HMAD

- X-axis: Percentage of anomalous data
- Y-axis: Detection accuracy [in AUC]
Python Code available at...

https://github.com/nicococo/LAD.git
- 2015-icml branch
- master branch