

Anomaly Detection Algorithm: Building and Evaluating

Overview

In building an anomaly detection algorithm, we aim to identify anomalies in a dataset. The algorithm is based on the Gaussian distribution and involves the following steps:

Step 1: Choose Features

1. Training Set (x_1 through x_m):

- Each example x has n features.
- Example: In aircraft engine manufacturing, features could include heat and vibrations.

Step 2: Density Estimation

2. Model for $p(x)$:

- Assume features are statistically independent (though the algorithm can work even if they aren't).
- Probability $p(x)$ modeled as the product of individual feature probabilities:

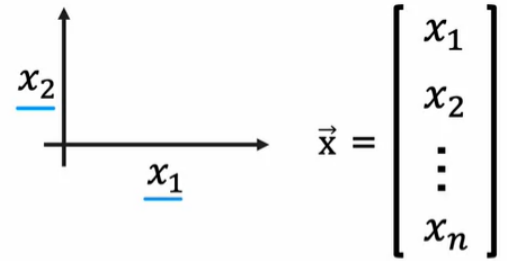
$$p(x) = \prod_{j=1}^n p(x_j)$$

- Each feature x_j is modeled by a Gaussian distribution:

$$p(x_j) = \frac{1}{\sqrt{2\pi}\sigma_j} e^{-\frac{(x_j - \mu_j)^2}{2\sigma_j^2}}$$

Density estimation

Training set: $\{\vec{x}^{(1)}, \vec{x}^{(2)}, \dots, \vec{x}^{(m)}\}$
Each example $\vec{x}^{(i)}$ has n features


$$\vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

$$p(\vec{x}) = p(x_1; \mu_1, \sigma_1^2) * p(x_2; \mu_2, \sigma_2^2) * p(x_3; \mu_3, \sigma_3^2) * \dots * p(x_n; \mu_n, \sigma_n^2)$$

$$= \prod_{j=1}^n p(x_j; \mu_j, \sigma_j^2) \quad \sum \quad \Pi$$

"add"

$$p(x_1 = \text{high temp}) = 1/10$$

$$p(x_2 = \text{high vibra}) = 1/20$$

$$p(x_1, x_2) = p(x_1) * p(x_2)$$

$$= \frac{1}{10} * \frac{1}{20} = \frac{1}{200}$$

Step 3: Parameter Estimation

3. Estimate Parameters (μ_j, σ_j^2):

- μ_j is the average of feature x_j over all examples.
- σ_j^2 is the average squared difference between x_j and μ_j .

$$\mu_j = \frac{1}{m} \sum_{i=1}^m x_j^{(i)}$$

$$\sigma_j^2 = \frac{1}{m} \sum_{i=1}^m (x_j^{(i)} - \mu_j)^2$$

Step 4: Anomaly Detection

4. Decision Rule:

- Given a new example x_{test} :
 - Compute $p(x_{\text{test}})$ using the product of individual feature probabilities.
 - If $p(x_{\text{test}}) < \epsilon$, flag as an anomaly.

Anomaly detection algorithm

1. Choose n features x_i that you think might be indicative of anomalous examples.
2. Fit parameters $\mu_1, \dots, \mu_n, \sigma_1^2, \dots, \sigma_n^2$

$$\mu_j = \frac{1}{m} \sum_{i=1}^m x_j^{(i)} \quad \sigma_j^2 = \frac{1}{m} \sum_{i=1}^m (x_j^{(i)} - \mu_j)^2$$

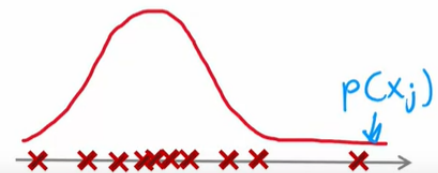
Vectorized formula

$$\vec{\mu} = \frac{1}{m} \sum_{i=1}^m \vec{x}^{(i)} \quad \vec{\mu} = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \dots \\ \mu_n \end{bmatrix}$$

3. Given new example x , compute $p(x)$:

$$p(x) = \prod_{j=1}^n p(x_j; \mu_j, \sigma_j^2) = \prod_{j=1}^n \frac{1}{\sqrt{2\pi}\sigma_j} \exp\left(-\frac{(x_j - \mu_j)^2}{2\sigma_j^2}\right)$$

Anomaly if $p(x) < \epsilon$



Parameter Choices

- **Choosing Features:**
 - Select features believed to indicate anomalous behavior.
- **Choosing ϵ :**
 - ϵ is a threshold to determine when an example is flagged as anomalous.
 - Typically chosen based on cross-validation on a labeled validation set.

Evaluation

- **How to know if the algorithm is working well?**
 - Use labeled data (with anomalies marked) to evaluate algorithm performance.
 - Metrics:
 - True Positive (TP), False Positive (FP), False Negative (FN), True Negative (TN).
 - Precision, Recall, F1 Score.
 - Adjust ϵ to balance precision and recall based on application requirements.

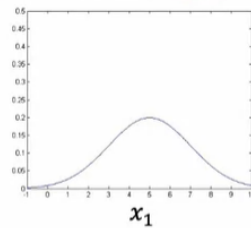
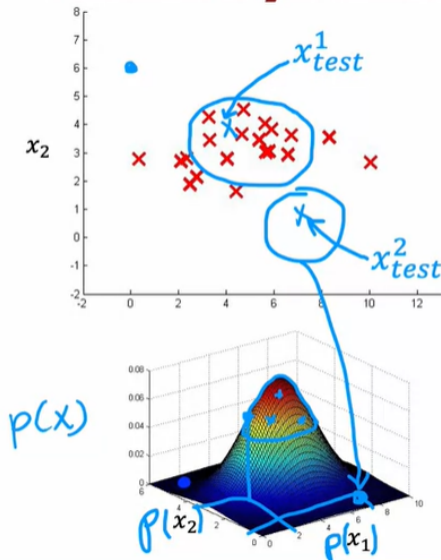
Example Evaluation

- **Parameters:**
 - $\epsilon = 0.02$
- **Test Examples:**
 - $x_{\text{test}_1} : p(x_{\text{test}_1}) \approx 0.4$ (Not flagged as anomaly)
 - $x_{\text{test}_2} : p(x_{\text{test}_2}) \approx 0.0021$ (Flagged as anomaly)

- **Conclusion:**

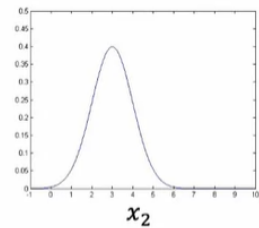
- Algorithm correctly identifies x_{test2} as an anomaly and ignores x_{test1} .

Anomaly detection example



$$\mu_1 = 5, \sigma_1 = 2$$

$$\underline{p(x_1; \mu_1, \sigma_1^2)}$$



$$\mu_2 = 3, \sigma_2 = 1$$

$$\underline{p(x_2; \mu_2, \sigma_2^2)}$$

$$\varepsilon = 0.02$$

$$p(x_{test}^{(1)}) = \underline{0.0426} \longrightarrow \text{"ok"}$$

$$p(x_{test}^{(2)}) = \underline{0.0021} \longrightarrow \text{anomaly}$$

Next Steps

In the next video, we'll explore parameter tuning, choosing an appropriate ϵ , and refining the anomaly detection system for optimal performance.