

Your anomaly detection system flags x as anomalous whenever $p(x) \leq \epsilon$. Suppose your system is flagging too many things as anomalous that are not actually so (similar to supervised learning, these mistakes are called false positives). What should you do?

Try increasing ϵ .

Try decreasing ϵ . Correct

Given a training set $\{x^{(1)}, \dots, x^{(m)}\}$, how would you estimate each μ_j and σ_j^2 (Note $\mu_j \in \mathbb{R}, \sigma_j^2 \in \mathbb{R}$.)

- ☐ $\mu_j = \frac{1}{m} \sum_{i=1}^m x^{(i)}, \sigma_j^2 = \frac{1}{m} \sum_{i=1}^m (x^{(i)} - \mu)^2$
- ☐ $\mu_j = \frac{1}{m} \sum_{i=1}^m (x_j^{(i)})^2, \sigma_j^2 = \frac{1}{m} \sum_{i=1}^m (x_j^{(i)} - \mu_j)^2$
- ☐ $\mu_j = \frac{1}{m} \sum_{i=1}^m x_j^{(i)}, \sigma_j^2 = \frac{1}{m} \sum_{i=1}^m (x^{(i)} - \mu)^2$
- ☒ $\mu_j = \frac{1}{m} \sum_{i=1}^m x_j^{(i)}, \sigma_j^2 = \frac{1}{m} \sum_{i=1}^m (x_j^{(i)} - \mu_j)^2$

Correct

Which of the following problems would you approach with an anomaly detection algorithm (rather than a supervised learning algorithm)? Check all that apply.

You run a power utility (supplying electricity to customers) and want to monitor your electric plants to see if any one of them might be behaving strangely.

Correct

You run a power utility and want to predict tomorrow's expected demand for electricity (so that you can plan to ramp up an appropriate amount of generation capacity).

Un-selected is correct

A computer vision / security application, where you examine video images to see if anyone in your company's parking lot is acting in an unusual way.

Correct

A computer vision application, where you examine an image of a person entering your retail store to determine if the person is male or female.

Un-selected is correct

Suppose your anomaly detection algorithm is performing poorly and outputs a large value of $p(x)$ for many normal examples and for many anomalous examples in your cross validation dataset. Which of the following changes to your algorithm is most likely to help?

Try using fewer features.

Try coming up with more features to distinguish between the normal and the anomalous examples. Correct

Get a larger training set (of normal examples) with which to fit $p(x)$.

Try changing ϵ .

In the algorithm we described, we initialized $x^{(1)}, \dots, x^{(n_m)}$ and $\theta^{(1)}, \dots, \theta^{(n_u)}$ to small random values. Why is this?

This step is optional. Initializing to all 0's would work just as well.

Random initialization is always necessary when using gradient descent on any problem.

This ensures that $x^{(i)} \neq \theta^{(j)}$ for any i, j .

This serves as symmetry breaking (similar to the random initialization of a neural network's parameters) and ensures the algorithm learns features $x^{(1)}, \dots, x^{(n_m)}$ that are different from each other.

Correct