Machine Learning of Andrew Ng
pdf by CarolusRex

Anomaly detection

- Problem motivation
  - Model
  - Gaussian distribution
  - Anomaly detection algorithm
  - Developing and Evaluating an Anomaly Detection System
  - compared with supervised learning
  - choosing what features to use
  - Multivariate Gaussian (Normal) distribution

Problem motivation

Dataset $\{x^{(1)}, ..., x^{(m)}\}$
New engine: x_{test}

Model

Fraud detection:

1. $x^{(i)}$ = features of user i's activities;
2. Model $p(x)$ from data;
3. Identify unusual users by checking which have $p(x) < \epsilon$

Gaussian distribution

$x \sim \mathbb{N}(\mu, \sigma^2)$
$p(x; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} exp(-\frac{(x-\mu)^2}{2\sigma^2})$

or $p(x) = \prod p(x_i; \mu_i, \sigma_i^2)$

Anomaly detection algorithm

1. Choose features x_i that you think might be indicative of anomalous examples;

2. Fit parameters $\mu_1, ..., \mu_n, \sigma_1^2, ..., \sigma_n^2$
   $\mu_j = \frac{1}{m}\sum_{i=1}^m x_j^{(i)}$
   $\sigma_j^2 = \frac{1}{m}\sum_{i=1}^m (x_j^{(i)} - \mu_j)^2$
3. Given new examples x, compute $p(x)$
   $p(x) = \prod_{j=1}^n p(x_j; \mu_j, \sigma_j^2) = \prod_{j=1}^n \frac{1}{\sqrt{2\pi}\sigma_j} exp(-\frac{(x_j-\mu_j)^2}{2\sigma_j^2})$

Anomaly if $p(x) < \epsilon$

Developing and Evaluating an Anomaly Detection System

Split the data 60/20/20 training/CV/test and then split the anomalous examples 50/50 between the CV and test sets.

possible evaluation matrics:

1. True Positive, False Positive, False Negative, True Negative
2. Precision/Recall
3. F1-score

to choose parameter ϵ

compared with supervised learning

Anomaly Detection	Supervised Learning
Very small number of positive examples ($y == 1$). (0--20 is common) Larger number of negative($y == 0$) examples.	Larger number of positive and negative examples
Many different "types" of anomalies. Hard for any algorithm to learn from positives examples what the anomalies look like future anomalies may look nothing like any of the anomalous examples we've seen so far	Enough positive examples for algorithm to get a sense of what positive examples are like, future positive examples likely to be similar to ones in training set.

choosing what features to use

non-gaussian features
$log(x), log(x+1), x^{\frac{1}{2}}, x^{\frac{1}{3}}$

Multivariate Gaussian (Normal) distribution

Parameters μ, Σ
$p(x;\mu,\Sigma) = \frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma|^{\frac{1}{2}}} exp(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu))$

Parameter fitting:
$\mu = \frac{1}{m}\sum_{i=1}^{m} x^{(i)}$
$\Sigma = \frac{1}{m}\sum_(i=1)^m(x^{(i)}-\mu)(x^{(i)}-\mu)^T$

Relationship to original model
Σ is a diagonal matrix.

Original model	Multivariate Gaussian
Manually create features to capture anomlies where x_1, x_2 take unusual combinations of values	Automatically captures correlations between features
computationally cheaper(alternatively, scales better to large n)	Computationally more expensive
OK even if m(training set size) is small	Must have $m > n$ or else Σ is non-invertible