# Anomaly Detection

## Density Estimation

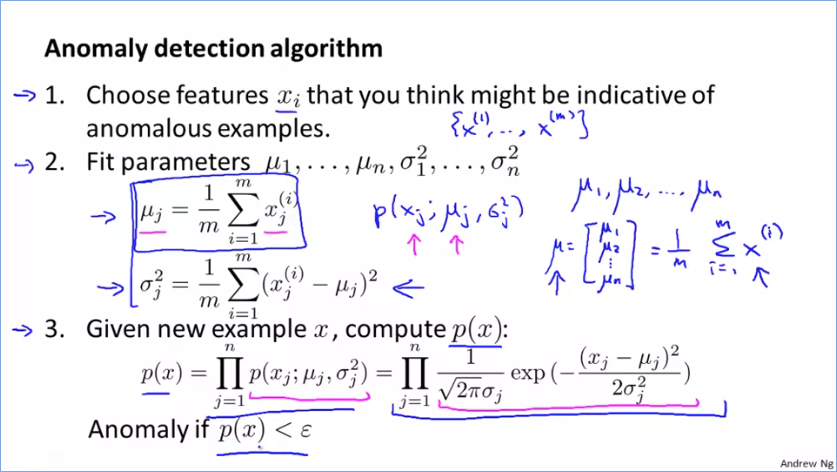
Problem Motivation

* Compare p to ε
* Examples/applications
* Fraud detection
* Manufacturing
* Machine performance

Gaussian Distribution

* Normal distribution
* x ~ *N*(μ, σ2)
* p(x; μ, σ2)
* Parameter estimation

Algorithm



## Building an Anomaly Detection System

Developing and Evaluating an Anomaly Detection System

* Training, CV and test sets
* Not a good idea to use accuracy as a measure due to skewness
* Precision, recall, F1

Anomaly Detection vs. Supervise Learning

* Numbers of positive and negative examples
* Types of anomalies

Choosing What Features to Use

* Non-Gaussian features
* Algorithm will work fine
* Can transform to log or exponent
* Error analysis

## Multivariate Gaussian Distribution

Multivariate Gaussian Distribution

* Model *p*(*x*) all in one go
* Capture feature correlation (*difference from extra correlation feature?*)

Anomaly Detection using the Multivariate Gaussian Distribution

* (similar to PCA)
* Original model is a special case of the multivariate on, where the covariance matric (Σ) would have zeros for non-diagonal elements (axis-aligned)
* Comparison
* Manual vs. automatic capturing of feature correlations (unusual combinations)
* Computational cost (wrt n)
* Must have m > n for Σ to be invertible (m >= 10n)
* Singularity causes
* Failure to satisfy m > n condition
* Redundant features

# Recommender Systems

## Predicting Movie Ratings

Problem Formulation

* Recommender system problem dataset notation
* nu, , nm, r(i, j), y(i, j)

Content-Based Recommendations

* Treat prediction of each user’s rating as a separate linear regression problem
* *Estimate θ’s from features*

## Collaborative Filtering

Collaborative Filtering

* Sum over all movies to learn all features
* *Estimate features from θ’s*

Collaborative Filtering Algorithm

* Minimize with respect to x’s and θ’s simultaneously

## Low Rank Matrix Factorization

Vectorization: Low Rank Matrix Factorization

* Essentially collaborative filtering algorithm
* Vectorization implementation
* Other applications

Implementational Detail: Mean Normalization

* Subtract mean from every movie rating; equivalent to assuming mean for no rating