ANOMALY DETECTION

Lecture 89: Problem Motivation

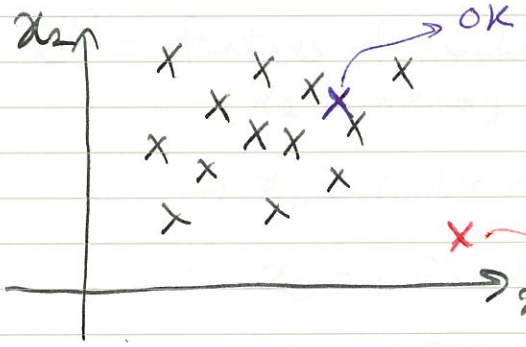
Consider an aircraft manufacturer that wants to make sure no defective aircrafts are given to customes. We can dessibe an aircraft by a given set of features:

x_1 = heat generated

x_2 = vibration intensity

⋮

Suppose we plot $x_2 - x_1$ for m aircrafts that we know are good:



If we have a new aircraft, can we tell if it's anomalous by comparing it to the good ones?

More gensally, we're given a data set $\{x^{(1)}, x^{(2)}, \dots, x^{(m)}\}$ which we assume is normal (not anomalous), and try to determine if a new example x_{test} is anomalous. We do this by modeling the probabity $p(x)$ of an example x occuring. Having that at our disposal, we can make decisions for new examples:

$$\begin{cases} p(x_{test}) < \varepsilon \Rightarrow \text{Anomalous} \\ p(x_{test}) \geq \varepsilon \Rightarrow \text{OK} \end{cases}$$

ε → small threshold

Anomaly detection examples

* Fraud detection:
  - $x^{(i)}$ = features of user i's activities:
    - → x_1 = How often does user log in?
    - → x_2 = Number of webpages visited.
    - → x_3 = Number of transactions
    - → x_4 = Number of posts on a form
    - → x_5 = Typing speed in characters per second
    - ⋮
  - Model $p(x)$ from data and use it to identify unusual users which have $p(x) < \varepsilon$.
* Manufacturing (our aircraft example, for instance)
* Monitoring computers in a data centre:
  - $x^{(i)}$ = features of machine i:
    - → x_1 = memory usage
    - → x_2 = number of disk accesses / sec
    - → x_3 = CPU load
    - → x_4 = CPU load / network traffic
    - ⋮
  - If $p(x) < \varepsilon$ for a given machine, we can flag it for inspection since it might be going down.

Lecture 90: Gaussian Distribution

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(x-\mu)^2/2\sigma^2}$$

& we write $x \sim N(\mu, \sigma^2)$ (distributed as; mean; variance)

This is also written as $p(x; \mu, \sigma^2)$ sometimes.

Parameter estimation

Suppose we have a data set $\{x^{(1)}, x^{(2)}, \dots, x^{(m)}\}$ where $x^{(i)} \in \mathbb{R}$ and we suspect $x^{(i)}$ are samples of a normal distribution $N(\mu, \sigma^2)$.

How do we estimate μ & σ^2?

- $\hat{\mu} = \frac{1}{m} \sum_{i=1}^{m} x^{(i)}$
- $\hat{\sigma}^2 = \frac{1}{m} \sum_{i=1}^{m} (x^{(i)} - \hat{\mu})^2$

Let's remind ourselves how these estimates are obtained. If $x^{(1)}, \dots, x^{(m)}$ are drawn independently from $N(\mu, \sigma^2)$, then the probability density of their observation is given by:

$$P = \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{(x^{(i)}-\mu)^2}{2\sigma^2}\right] = \frac{1}{(2\pi)^{m/2}\sigma^m} e^{-\frac{\sum_{i=1}^{m}(x^{(i)}-\mu)^2}{2\sigma^2}}$$

Let's pick the values of μ & σ which maximize P.

$$\frac{\partial P}{\partial \mu} = 0 \Rightarrow P \cdot \frac{-1}{2\sigma^2} \cdot \sum_{i=1}^{m} 2(x^{(i)} - \mu) \cdot x - 1 = 0 \Rightarrow \hat{\mu} = \frac{1}{m} \sum_{i=1}^{m} x^{(i)}.$$

Now suppose we pick m samples & compute $\hat{\mu}$, then pick another

m samples & compute $\hat{\mu}$, and so on. Obviously, each time we'd obtain a different number. What would be the mean of these numbers? Remember that we're assuming $x^{(i)} \sim N(\mu, \sigma^2)$, so:

$$\langle \hat{\mu} \rangle = \frac{1}{m} \sum_{i=1}^{m} \langle x^{(i)} \rangle = \frac{1}{m} \sum_{i=1}^{m} \mu = \mu.$$

So $\langle \hat{\mu} \rangle = \mu$, the true mean. For this reason, $\hat{\mu}$ is called an unbiased estimator.

Let's go through the same procedure for σ:

$$\frac{\partial P}{\partial \sigma} = 0 \Rightarrow \frac{1}{(2\pi)^{m/2}} \times -m\sigma^{-m-1} \exp\left[-\frac{1}{2\sigma^2} \sum_{i=1}^{m} (x^{(i)} - \mu)^2\right]$$

$$+ P \cdot \frac{1}{\sigma^3} \sum_{i=1}^{m} (x^{(i)} - \mu)^2 = 0$$

$$\Rightarrow \left[-\frac{m}{\sigma} + \frac{1}{\sigma^3} \sum_{i=1}^{m} (x^{(i)} - \mu)^2\right] P = 0$$

$$\Rightarrow \hat{\sigma}^2 = \frac{1}{m} \sum_{i=1}^{m} (x^{(i)} - \mu)^2.$$

Of course, we only have an estimate of μ (i.e $\hat{\mu}$), so let's take our variance estimator to be: $\hat{\sigma}^2 = \frac{1}{m} \sum_{i=1}^{m} (x^{(i)} - \hat{\mu})^2$.

Is $\hat{\sigma}^2$ unbiased?

$$\hat{\sigma}^2 = \frac{1}{m} \sum_{i=1}^{m} \left[x^{(i)} - \frac{1}{m} \sum_{i'=1}^{m} x^{(i')}\right]^2$$

$$= \frac{1}{m} \sum_{i=1}^{m} \left[x^{(i)} x^{(i)} - \frac{2}{m} x^{(i)} \sum_{i'=1}^{m} x^{(i')} + \frac{1}{m^2} \sum_{i',i''=1}^{m} x^{(i')} x^{(i'')}\right]$$

$$\hat{\sigma}^2 = \frac{1}{m}\sum_{i=1}^{m}(x^{(i)})^2 - \frac{1}{m^2}\sum_{i,i'=1}^{m} x^{(i)}x^{(i')}$$

$$\langle\hat{\sigma}^2\rangle = \frac{1}{m}\sum_{i=1}^{m}\langle(x^{(i)})^2\rangle - \frac{1}{m^2}\sum_{i,i'=1}^{m}\langle x^{(i)}x^{(i')}\rangle$$

$$\langle(x^{(i)})^2\rangle = \langle(x^{(i)}-\mu+\mu)^2\rangle = \langle(x^{(i)}-\mu)^2 + 2\mu(x^{(i)}-\mu) + \mu^2\rangle$$

$$= \langle(x^{(i)}-\mu)^2\rangle + 2\mu\langle(x^{(i)}-\mu)\rangle + \mu^2 = \sigma^2 + \mu^2$$

$$\langle x^{(i)}x^{(i')}\rangle = \langle(x^{(i)}-\mu+\mu)(x^{(i')}-\mu+\mu)\rangle = \sigma^2\delta_{ii'} + \mu^2$$

(we've used $\langle(x^{(i)}-\mu)(x^{(i')}-\mu)\rangle = \delta_{ii'}\sigma^2$ → different examples are independent)

$$\langle\hat{\sigma}^2\rangle = \frac{1}{m}\sum_{i=1}^{m}(\sigma^2+\mu^2) - \frac{1}{m^2}\sum_{i,i'=1}^{m}\sigma^2\delta_{ii'} + \mu^2$$

$$= \sigma^2 + \mu^2 - \frac{1}{m^2}(\sigma^2 \times m + \mu^2 \times m^2)$$

$$= \sigma^2\left(1 - \frac{1}{m}\right) = \frac{m-1}{m}\sigma^2$$

We see that $\hat{\sigma}^2$ is not unbiased. That's why sometimes people use $\hat{\sigma}^2 = \frac{1}{m-1}\sum_{i=1}^{m}(x^{(i)}-\hat{\mu})^2$ which would be unbiased. For large values of m this won't make any difference in practice. Andrew Ng claims that in machine learning the biased estimator is used, i.e. $\hat{\sigma}^2 = \frac{1}{m}\sum_{i=1}^{m}(x^{(i)}-\hat{\mu})^2$.

Lecture 9: Algorithm

Consider the training set $\{x^{(1)}, \ldots, x^{(m)}\}$ where $x^{(i)} \in \mathbb{R}^n$. We will assume that each feature is distributed normally & that the different features are not correlated:

$$x_1 \sim \mathcal{N}(\mu_1, \sigma_1^2), \ldots, x_n \sim \mathcal{N}(\mu_n, \sigma_n^2)$$

We can then write $p(x)$ as:

$$p(x) = p(x_1; \mu_1, \sigma_1^2) \times p(x_2; \mu_2, \sigma_2^2) \times \cdots \times p(x_n; \mu_n, \sigma_n^2)$$

where $p(x_j; \mu_j, \sigma_j^2) = \frac{1}{\sqrt{2\pi}\,\sigma_j} \exp\left[-\frac{1}{2}\frac{(x_j - \mu_j)^2}{\sigma_j^2}\right]$

μ_j, σ_j^2 are estimated as follows:

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

Later we will talk about generalizing this result so that correlations are captured, via the multivariate Gaussian. Let's summarize our model of the probability distribution $p(x)$:

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

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

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

Then given a new example x_{test}, we say x_{test} is anomalous if $p(x_{test}) < \varepsilon$ for small ε.

This algorithm makes a few questionable assumptions: we're basically fitting for n uncorrelated Gaussians. What if our features are not normally distributed? What if they're correlated? We'll talk about these issues & more next.

Lecture 92: Developing and Evaluating an Anomaly Detection System

In lecture 67 (see pages 27-28 of Notebook #2) we talked about the importance of having a single real number to evaluate a learning algorithm. For instance, when picking one feature over another, or deciding what threshold to use in logistic regression for predictions, it'd be useful to have one metric that can tell us whether our learning algorithm is doing better or worse. In our discussion of supervised learning, we developed some metrics: cross validation error $J_{cv} = \frac{1}{2m_{cv}} \sum_{i=1}^{m_{cv}} [h_\theta(x_{cv}^{(i)}) - y_{cv}^{(i)}]^2$ for linear regression, F_1 score (see Lecture 69 - pgs 30-33 of Notebook #2) for classification problems (especially skewed datasets), etc.

How can we evaluate our anomaly detection algorithm in a similar way? Let's assume our data set is labeled & $y=0$ if normal, $y=1$ if anomalous.

For instance, in our aircraft manufacturing example, we might have some examples of aircrafts with defective engines which had to be fixed. In this case, we can split our data as follows:

* Training set: $\{x^{(1)}, x^{(2)}, \ldots, x^{(m)}\}$ where all examples are normal. This makes sense since we want the probability distribution of non-anomalous examples. Andrew Ng mentions it's okay if a bunch of anomalous examples find their way in the training set, but we should try to keep them out.
* Cross-validation set: $\{(x_{cv}^{(1)}, y_{cv}^{(1)}), \ldots, (x_{cv}^{(m)}, y_{cv}^{(m)})\}$. where we have both anomalous & normal examples.
* Test set: $\{(x_{test}^{(1)}, y_{test}^{(1)}), \ldots, (x_{test}^{(m)}, y_{test}^{(m)})\}$ where we have both anomalous & normal examples.

In our aircraft manufacturing example, suppose we have the following data:

* 10,000 good (normal) engines
* 20 flawed engines (anomalous)

This is a typical anomalous/normal ratio in anomaly detection problems. We can construct the above data sets as follows:

* Training set: 6000 good engines
* Cross-validation set: 2000 good engines ($y=0$), 10 anomalous ($y=1$)
* Test set: 2000 good engines ($y=0$), 10 anomalous ($y=1$).

Equipped with this, we can proceed as follows:

(i) Fit model $p(x)$ on training set $\{x^{(1)}, \ldots, x^{(m)}\}$

(ii) On a cross-validation/test set example x, predict:

$$y = \begin{cases} 1 & \text{if } p(x) < \varepsilon \quad \text{(anomaly)} \\ 0 & \text{if } p(x) \geq \varepsilon \quad \text{(normal)} \end{cases}$$

(iii) Use an evaluation metric to determine how well anomalies are detected:

* True positive, false positive, false negative, true negative
* Precision / Recall
* F_1-score (single #)

Note that the dataset is very skewed, so we should be using precision/recall or F_1-score.

We can also choose ε using the validation set: pick ϵ that results in the highest F_1-score. What happens if we take ε to be very small? Precision will be high since we're predicting very few anomalies that are likely to be actual anomalies, but Recall will be low because of all actual anomalies, we've caught only a few. The opposite is true if ϵ is too large. We can use F_1-score to pick the optimal ε.

Lecture 93: Anomaly Detection vs. Supervised Learning

In the previous lecture we assumed we have a labeled dataset to be able to evaluate anomaly detection. If we have a labeled dataset, however, why wouldn't we just use a supervised classification algorithm? The crucial determinant is the number of positive examples. If we have a very small number of positive example (e.g. 0 - 20), it will be hard for a supervised learning algorithm to learn what positive examples are generally like. This is especially so when there are many different types of anomalies, as is the case with aircraft engine failure. There are so many potential causes of engine failure, that it will be hard for any algorithm to learn from a handful of positive examples what the anomalies look like; future anomalies may look nothing like any of the anomalous examples we've seen so far. Therefore, in such cases, it makes more sense to model the negative examples, of which there are often many, and use that to detect anomalies. As we discussed in the previous lecture, the handful of positive examples can be used to evaluate the anomaly detection algorithm. For supervised learning algorithms to work well, we often want a large number of positive and negative examples. In the case of spam classification, there are many types of spam emails, but

our training sets often have many spam examples, so it's quite likely that an unseen spam email is similar to some examples in the training set.

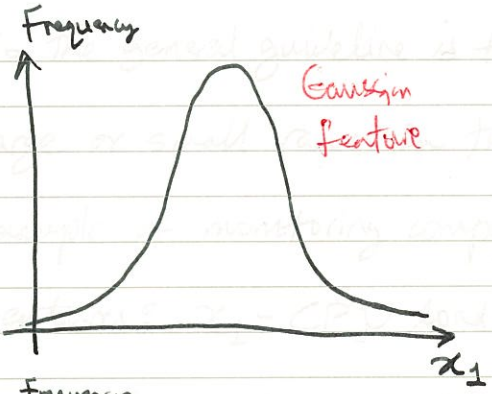
Typical examples of anomaly detection vs. supervised learning:

Anomaly detection	Supervised learning
* Fraud detection	* Email spam classification
* Manufacturing (e.g. aircraft engines)	* Weather prediction (sunny/rainy/etc)
* Monitoring machines in a data center.	* Cancer classification

When there's a lot of positive examples, some of these anomaly detection examples could be studied using supervised learning. For instance, if a giant online retailer has many cases of fraudulent behaviour, they could use supervised learning to detect anomalies.

Lecture 9.4: Choosing What Features to Use

In practice, the performance of the anomaly detection algorithm is affected quite a lot by the choice of features. We modeled our features as independent normally-distributed variables. The Gaussian assumption is a bold one and may not be true. Therefore, it's a good idea to draw the histogram of values of a given feature across our training set:



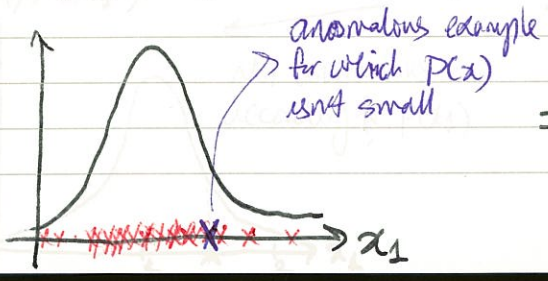
If we get a plot like this, then we know our assumption is probably okay.

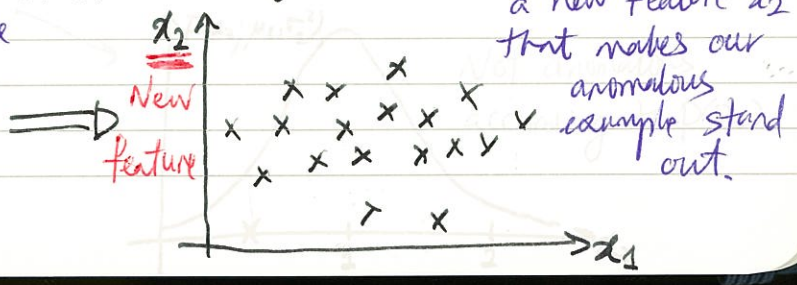
If we get a plot like this, we should transform our feature so that it becomes more Gaussian. For instance, we might try: $x_1 \to \log x_1$, which would do the trick if x_1 has a log-normal distribution. This is, of course, just one example. Here are some others:

$x_2 \to \sqrt{x_2}$, $x_3 \to x_3^{1/3}$, $x_4 \to \log(x_4 + c)$, etc.

For anomaly detection, we want our modeled probability distribution to have the following behaviour: $\begin{cases} p(x) \text{ large for normal examples } x. \\ p(x) \text{ small for anomalous examples } x. \end{cases}$

A common problem, however, is that $p(x)$ might end up being comparable for normal and anomalous examples. In such cases, we should study the anomalous examples for which $p(x)$ is not small and try to come up with features that make them stand out:

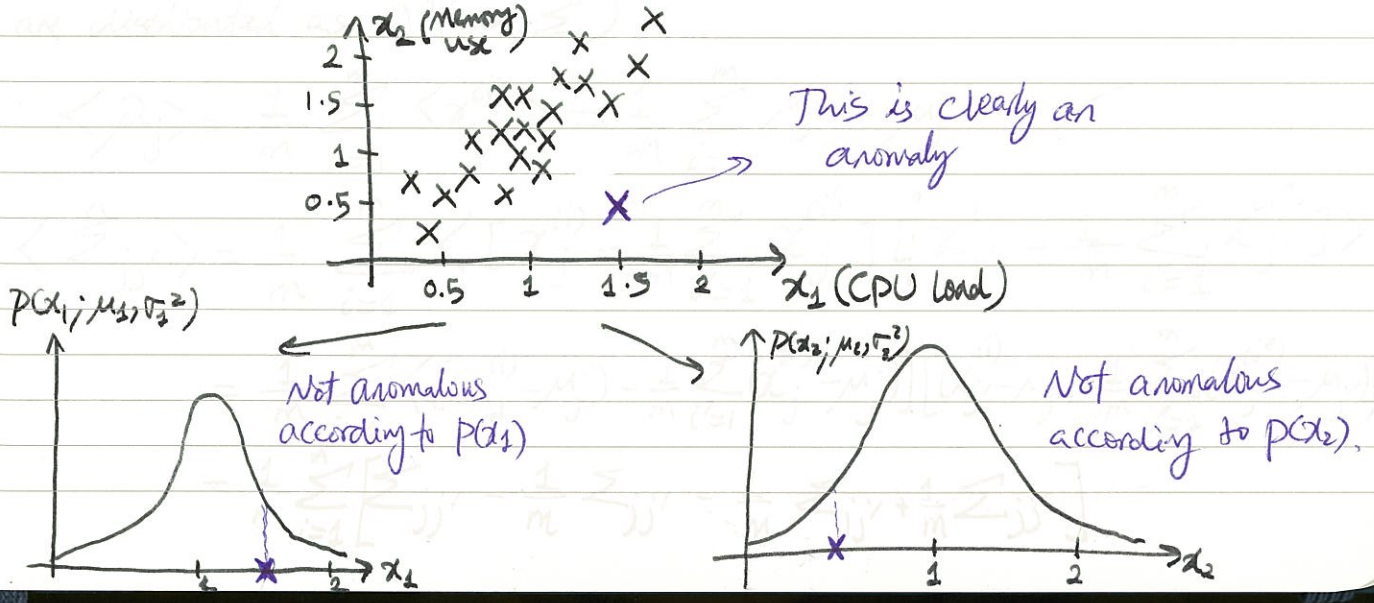




So the general guideline is to choose features that might take on unusually large or small values in the event of an anomaly. Let's go back to the example of monitoring computers in a data center. Consider the two features: x_1 = CPU load & x_2 = Network traffic. If the computers are running webservers, x_1 & x_2 are correlated: the more network traffic, the more CPU usage. If we want to distinguish computers for which CPU load is not growing with network traffic, for instance because they're stuck in an infinite loop, we can create a new feature $x_3 = x_1 / x_2$, which takes unusually high values if CPU usage is high despite modest or low network traffic.

Lecture 95: Multivariate Gaussian Distribution

The anomaly detection algorithm we described in Lecture 91 assumes different features are not correlated. This can cause issues:



Because our model is assuming x_1 & x_2 are not correlated, it will not detect the anomaly. To get around this, we'll model $x \in \mathbb{R}^n$ as a multivariate Gaussian parametrised by $\mu \in \mathbb{R}^n$ (mean) & $\Sigma \in \mathbb{R}^{n \times n}$ (covariance matrix):

$$p(x; \mu, \Sigma) = \frac{1}{(2\pi)^{n/2} |\Sigma|^{1/2}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)}$$

$|\Sigma|$ → determinant of Σ.

It can be shown that $\begin{cases} \Sigma_{jj'} = \langle (x_j - \mu_j)(x_{j'} - \mu_{j'}) \rangle \\ \mu_j = \langle x_j \rangle \end{cases}$

Let's go through our sample estimate arguments (see Lecture 20) again:

- $\hat{\mu}_j = \frac{1}{m} \sum_{i=1}^{m} x_j^{(i)}$
- $\hat{\Sigma}_{jj'} = \frac{1}{m} \sum_{i=1}^{m} (x_j^{(i)} - \hat{\mu}_j)(x_{j'}^{(i)} - \hat{\mu}_{j'})$

Let's check if $\hat{\mu}$ & $\hat{\Sigma}$ are unbiased estimates, remembering that $x^{(i)}$ are distributed as $\mathcal{N}(\mu, \Sigma)$.

$$\langle \hat{\mu}_j \rangle = \frac{1}{m} \sum_{i=1}^{m} \langle x_j^{(i)} \rangle = \frac{1}{m} \sum_{i=1}^{m} \mu_j = \mu_j \quad \checkmark$$

$$\langle \hat{\Sigma}_{jj'} \rangle = \frac{1}{m} \sum_{i=1}^{m} \left\langle \left[x_j^{(i)} - \frac{1}{m} \sum_{i'=1}^{m} x_j^{(i')} \right] \left[x_{j'}^{(i)} - \frac{1}{m} \sum_{i''=1}^{m} x_{j'}^{(i'')} \right] \right\rangle$$

$$= \frac{1}{m} \sum_{i=1}^{m} \left\langle \left[(x_j^{(i)} - \mu_j) - \frac{1}{m} \sum_{i'=1}^{m} (x_j^{(i')} - \mu_j) \right] \left[(x_{j'}^{(i)} - \mu_{j'}) - \frac{1}{m} \sum_{i''=1}^{m} (x_{j'}^{(i'')} - \mu_{j'}) \right] \right\rangle$$

$$= \frac{1}{m} \sum_{i=1}^{m} \left[\Sigma_{jj'} - \frac{1}{m} \Sigma_{jj'} - \frac{1}{m} \Sigma_{jj'} + \frac{1}{m} \Sigma_{jj'} \right]$$

where we've used that $\langle (x_j^{(i)} - \mu_j)(x_{j'}^{(i')} - \mu_{j'}) \rangle = \Sigma_{jj'} \delta_{ii'}$

different examples are independent

$\langle \hat{\Sigma}_{jj'} \rangle = \frac{m-1}{m} \Sigma_{jj'}$, much as it was the case in the case of a single Gaussian. For large values of m, $\frac{m-1}{m} \approx 1$ & this won't matter much.

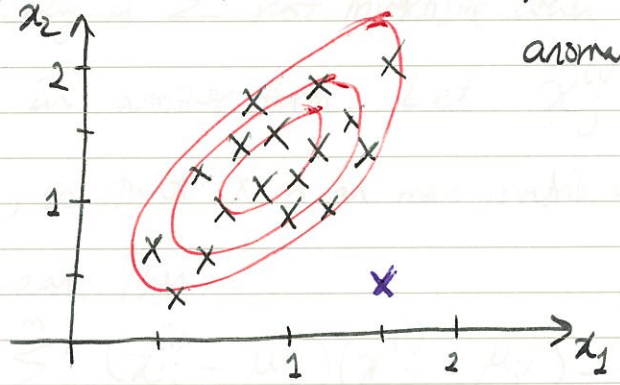
Lecture 96: Anomaly Detection Using the Multivariate Gaussian Distribution

Using the multivariate Gaussian to model $p(x)$, we have:

$$p(x) = \frac{1}{(2\pi)^{n/2} |\Sigma|^{1/2}} \exp\left[-\frac{1}{2} (x-\mu)^T \Sigma^{-1} (x-\mu) \right]$$

where $\mu = \frac{1}{m} \sum_{i=1}^{m} x^{(i)}$ & $\Sigma = \frac{1}{m} \sum_{i=1}^{m} (x^{(i)} - \mu)^T (x^{(i)} - \mu)^T$

For our example in the previous lecture, $p(x)$ would now be small for the anomalous example.



The model introduced in lecture 91 is a special case of the model above, where $\Sigma_{jj'} = \sigma_j^2 \delta_{jj'}$. As we're already mentioned, the multivariate

Gaussian model automatically captures correlations, whereas for the original model we need to manually create new features (like $\frac{\text{CPU load}}{\text{memory usage}}$) to handle correlations. The original model, however, is computationally cheaper and scales better for large values of n. The reason is that we need Σ^{-1}, which typically scales as n^3. For $n \gtrsim 10,000$, inverting Σ becomes too computationally expensive. We faced the same dilemma in linear regression, when choosing between the Normal Equation and Gradient Descent (see Lecture 24, pgs 22-24 of Notebook #1).

Another case where we cannot use the multivariate Gaussian model is when $m < n$, since in that case Σ is not invertible. We will soon prove this, but it shouldn't be that hard to believe if we're fitting about $\frac{n^2}{2}$ parameters, so we need a lot of data.

Okay, so why is Σ not invertible when $m < n$? Let's first rewrite Σ in another form. Let $\hat{x}_j^{(i)} \equiv x_j^{(i)} - \mu_j$ & $\hat{X}_{ij} \equiv \hat{x}_j^{(i)}$, so that $\hat{X}$ is an $m \times n$ matrix where every feature (i.e. column) has zero mean. Then:

$$\Sigma_{jj'} = \frac{1}{m} \sum_{i=1}^{m} (x_j^{(i)} - \mu_j)(x_{j'}^{(i)} - \mu_{j'}) = \frac{1}{m} \sum_{i=1}^{m} \hat{x}_j^{(i)} \hat{x}_{j'}^{(i)}$$

$$= \frac{1}{m} \sum_{i=1}^{m} \hat{X}_{ij} \hat{X}_{ij'} = \frac{1}{m} [\hat{X}^T \hat{X}]_{jj'} \Rightarrow \Sigma = \frac{1}{m} \hat{X}^T \hat{X}.$$

This is actually what we have already seen in PCA (see page 35 for instance). Let $A \equiv \hat{X}/\sqrt{m}$ so that $\Sigma = A^T A$.

Of course, Σ is symmetric & positive semi-definite (see page 32 for the proof) and as a result can be diagonalized and all of its eigenvalues are greater or equal to zero:

$$\Sigma v^{(i)} = \lambda_i v^{(i)} \qquad i = 1, \dots, n$$

where $v^{(i)} \cdot v^{(j)} = \delta_{ij}$ & $\lambda_i \geq 0$.

Now A is an $m \times n$ matrix, so $W \equiv AA^T$ is an $m \times m$ symmetric & positive definite matrix:

$$W u^{(i)} = \xi_i u^{(i)} \qquad i = 1, \dots, m$$

where $u^{(i)} \cdot u^{(j)} = \delta_{ij}$ & $\xi_i \geq 0$.

Claim I: $Av^{(i)}$ is an eigenvector of W with eigenvalue λ_i if $\lambda_i > 0$.

Proof. $W(Av^{(i)}) = AA^T(Av^{(i)}) = A(A^TA)v^{(i)} = A\Sigma v^{(i)}$

$$= A(\lambda_i v^{(i)}) = \lambda_i (Av^{(i)}).$$

Claim II: $A^T u^{(i)}$ is an eigenvector of Σ with eigenvalue ξ_i if $\xi_i > 0$

Proof. $\Sigma(A^T u^{(i)}) = A^TA(A^T u^{(i)}) = A^T(AA^T)u^{(i)} = A^T W u^{(i)}$

$$= A^T(\xi_i u^{(i)}) = \xi_i (Au^{(i)}).$$

What about $Av^{(i)}$ when $\lambda_i = 0$?

$$(Av^{(i)})^T(Av^{(i)}) = v^{(i)T}A^TAv^{(i)} = v^{(i)T}\Sigma V^{(i)} = \lambda_i = 0$$

$$\Rightarrow Av^{(i)} = 0.$$

Similarly with $A^Tw^{(i)}$ when $\xi_i = 0$:

$$(A^Tw^{(i)})^T(A^Tw^{(i)}) = w^{(i)T}AA^Tw^{(i)} = w^{(i)T}Ww^{(i)} = \xi_i = 0$$

$$\Rightarrow A^Tw^{(i)} = 0.$$

Claim III: The maximum number of non-zero eigenvalues of both Σ & W is $\min(m,n)$. Also, Σ & W share the same set of eigenvalues.

proof. Assume $m > n$ & $\xi_i > 0$ $\forall$ $i = 1, \dots, m$. Let $z^{(i)} = \frac{1}{\sqrt{\xi_i}} A^Tw^{(i)}$.

Then $z^{(i)T}z^{(j)} = \frac{1}{\sqrt{\xi_i}\sqrt{\xi_j}}(A^Tw^{(i)})^T(A^Tw^{(j)}) = \frac{1}{\sqrt{\xi_i\xi_j}} w^{(i)T}(AA^T)w^{(j)}$

$$= \frac{1}{\sqrt{\xi_i\xi_j}} w^{(i)T}Ww^{(j)} = \frac{\xi_j}{\sqrt{\xi_i\xi_j}}\delta_{ij} = \delta_{ij} \quad \forall\, i,j = 1,\dots,m.$$

We've shown that $z^{(1)}, \dots, z^{(m)}$ form an orthonormal basis. Remember that $z^{(i)} \in \mathbb{R}^n$, and since $m > n$, this is impossible! Therefore W can only have n non-zero eigenvalues at maximum. In this case, $n = \min(n,m)$.

Assume $n > m$ & $\lambda_i > 0$ $\forall$ $i = 1, \dots, n$. Let $z^{(i)} = \frac{1}{\sqrt{\lambda_i}} Av^{(i)}$.

Then $z^{(i)T}z^{(j)} = \frac{1}{\sqrt{\lambda_i\lambda_j}} v^{(i)T}\underbrace{(A^TA)}_{\Sigma}v^{(j)} = \delta_{ij}$. Again, we cannot have n orthonormal vectors in $\mathbb{R}^m$ when $n > m$, therefore Σ can at most have $m = \min(n,m)$ non-zero eigenvalues.

That Σ & W share the same eigenvalues follows from claims I & II.

If λ_i is a non-zero eigenvalue of Σ, it's also an eigenvalue of W. But can W have more non-zero eigenvalues than Σ? No, because every non-zero eigenvalue of W is also an non-zero eigenvalue of W.

So, to get back to why Σ is non-invertible when $m<n$: this is because it will definitely have a zero eigenvalue, since it can have at most m non-zero eigenvalues.

With the results we've proven so far, it would be a shame not to discuss the Singular-Value Decomposition (SVD).

Claim IV: Every $m \times n$ matrix A can be written as:

$$A = U D V^T$$

where:
- U is $m \times m$ & satisfies $U^T U = U U^T = 1_m$
- V is $n \times n$ & satisfies $V^T V = V V^T = 1_n$
- D is $m \times n$ & satisfies $D_{ij} = \alpha_i \delta_{ij}$

where $i = 1, \dots, m$, $j = 1, \dots, n$ & $\alpha_i \geq 0$.

Proof: Let $\lambda_1, \dots, \lambda_k$ ($k \leq \min(n,m)$) denote the strictly positive eigenvalues of Σ & W. Let $v^{(1)}, \dots, v^{(k)}$ be the corresponding eigenvectors of Σ.

Define: $V = \left[\left[v^{(1)} \right] \cdots \left[v^{(k)} \right] \left[v^{(k+1)} \right] \cdots \left[v^{(n)} \right] \right]$

orthonormal eigenvectors corresponding to zero eigenvalues of Σ.

$\Sigma v^{(i)} = \lambda_i v^{(i)}$

& $v^{(i)} \cdot v^{(j)} = \delta_{ij}$

& $\lambda_1, \dots, \lambda_k > 0$

& $\lambda_{k+1} = \dots = \lambda_n = 0$

Similarly, let $w^{(1)} = \frac{1}{\sqrt{\lambda_1}} A v^{(1)}, \dots, w^{(k)} = \frac{1}{\sqrt{\lambda_k}} A v^{(k)}$

& pick $w^{(k+1)}, \dots, w^{(m)}$ to be orthonormal eigenvectors of W corresponding to zero eigenvalue s.t. $w^{(i)} \cdot w^{(j)} = \delta_{ij}$ & $W w^{(i)} = \xi_i w^{(i)}$

where as we've shown $\xi_1 = \lambda_1, \xi_2 = \lambda_2, \dots, \xi_k = \lambda_k$ & $\xi_{k+1} = \dots = \xi_m = 0$.

Define: $U = \begin{bmatrix} \begin{bmatrix} w^{(1)} \end{bmatrix} \cdots \begin{bmatrix} w^{(k)} \end{bmatrix} \begin{bmatrix} w^{(k+1)} \end{bmatrix} \cdots \begin{bmatrix} w^{(m)} \end{bmatrix} \end{bmatrix}$

$= \frac{1}{\sqrt{\lambda_1}} A v^{(1)}$ $= \frac{1}{\sqrt{\lambda_k}} A v^{(k)}$ → eigenvectors corresponding to zero eigenvalue of W.

Of course, V is nxn & U is mxm.

Let $D_{ij} = \sqrt{\lambda_i}\, \delta_{ij}$, $i = 1, \dots, m$ & $j = 1, \dots, n$.

So $D_{11} = \sqrt{\lambda_1}, \dots, D_{kk} = \sqrt{\lambda_k}$ & all other entries of D are zeros.

Then:

$$UDV^T = \begin{bmatrix} \begin{bmatrix} \frac{Av^{(1)}}{\sqrt{\lambda_1}} \end{bmatrix} \cdots \begin{bmatrix} \frac{Av^{(k)}}{\sqrt{\lambda_k}} \end{bmatrix} \begin{bmatrix} w^{(k+1)} \end{bmatrix} \cdots \begin{bmatrix} w^{(m)} \end{bmatrix} \end{bmatrix}_{m \times m} \begin{bmatrix} \sqrt{\lambda_1} & & & \\ & \sqrt{\lambda_2} & & \\ & & \ddots & \\ & & & \sqrt{\lambda_k} \\ & & & \end{bmatrix}_{m \times n} \begin{bmatrix} [- & v^{(1)T} & -] \\ & \vdots & \\ [- & v^{(k)T} & -] \\ [- & v^{(k+1)T} & -] \\ & \vdots & \\ [- & v^{(n)T} & -] \end{bmatrix}_{n \times n}$$

$$= \begin{bmatrix} \begin{bmatrix} \frac{Av^{(1)}}{\sqrt{\lambda_1}} \end{bmatrix} \cdots \begin{bmatrix} \frac{Av^{(k)}}{\sqrt{\lambda_k}} \end{bmatrix} \begin{bmatrix} w^{(k+1)} \end{bmatrix} \cdots \begin{bmatrix} w^{(m)} \end{bmatrix} \end{bmatrix}_{m \times m} \begin{bmatrix} [- & \sqrt{\lambda_1} v^{(1)T} & -] \\ & \vdots & \\ [- & \sqrt{\lambda_k} v^{(k)T} & -] \\ & \text{(zeros)} & \end{bmatrix}_{m \times n}$$

Carrying out the matrix multiplication leads to:

$$UDV^T = A\bar{V}\bar{V}^T \quad \text{where} \quad \bar{V} = \begin{bmatrix} \left[v^{(1)} \right] & \cdots & \left[v^{(k)} \right] \end{bmatrix}_{n \times k}$$

Let $\tilde{V} = \begin{bmatrix} \left[v^{(k+1)} \right] & \cdots & \left[v^{(n)} \right] \end{bmatrix}_{n \times (n-k)}$ be the collection of eigenvectors with zero eigenvalues.

It then follows from $VV^T = \mathbb{1}_n$ that $\bar{V}\bar{V}^T + \tilde{V}\tilde{V}^T = \mathbb{1}_n$.

Multiplying by A from left: $A\bar{V}\bar{V}^T + A\tilde{V}\tilde{V}^T = A$. But $A\tilde{V} = 0$ since $Av^{(k+1)} = Av^{(k+2)} = \cdots = Av^{(n)} = 0$ (see page 64), so: $A\bar{V}\bar{V}^T = A$.

This concludes the proof: $UDV^T = A$. Note that we also showed that d_i, which are called the singular values of A, are given by positive eigenvalues of A^TA & AA^T.

This was a long detour! Back to anomaly detection. Andrew Ng mentions that he would use the multivariate Gaussian model if $m \geq 10n$, just to be safe. Redundant feature could also cause Σ to be non-invertible.