CIS 7000-005: Debugging Data & Models Fall 2022

Distribution shift — September 8

Prof. Eric Wong

Another area in which the ML pipeline can go wrong is *distribution shift*.

- What is a distribution shift?
- Why does distribution shift happen?
- How can you measure or detect shifts?

1 Types of distribution shift

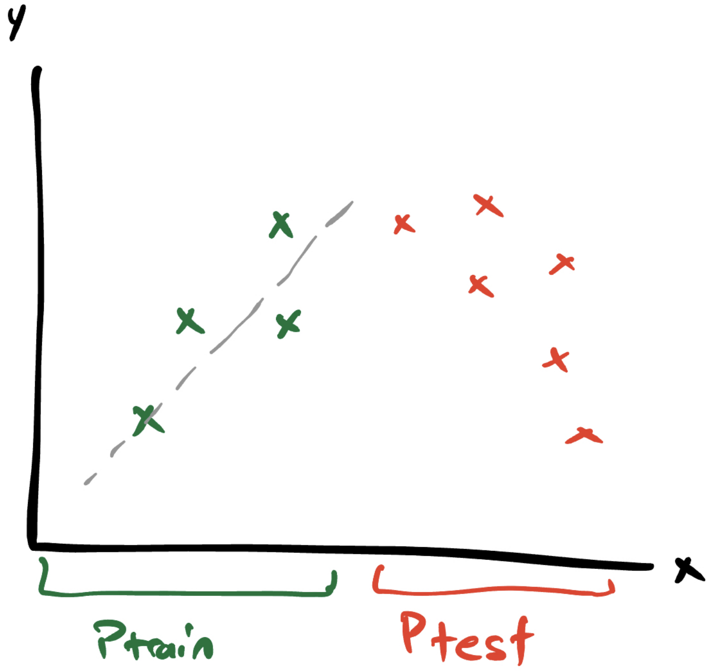
Typically in ML, we assume that the data is independent and identically distributed (i.i.d.), and that the test distribution matches the train distribution. So if our training data is drawn from p_{train}, and our test data is drawn from p_{test}, we assume that all samples are drawn i.i.d. and that $p_{train} = p_{test}$.

Distribution shift describes what happens when this assumption no longer holds, or $p_{train} \neq p_{test}$. How this occurs is usually described in one of three settings, depending on whether it affects the features, labels, or both:

1. Covariate shift: a change in $p(x)$. In this case, the feature distribution changes, while the label distribution does not. This includes subproblems such as domain shift and subpopulation shift, or sampling bias and representation bias. For example, for a cats vs. dogs classifier, in one location, dogs may be primarily photographed outdoors, while cats are primarily photographed indoors, whereas in another location, cats may be more commonly seen outdoors. The label of the cat and dog does not change whether it is indoors or outdoors, so the features shift in isolation. Another example is changes in languages across regions (i.e. dialects or slang, such as soda vs. pop).
2. Label shift: a change in $p(y)$. This can occur when, for example, our knowledge about the world changes (either over time or via expert knowledge). In the cats vs. dogs example, a contrived example for this could be when experts realize we've been mis-identifying a dog as a cat this entire time. A more natural example is the COVID-19 pandemic: a model predicting the positivity rate of COVID tests will experience label shift as the pandemic goes through waves over time. In this case, we could (roughly) assume that the population isn't drastically changing, and that the infection mechanism linking people to positive tests remains the same, so the labels shift in isolation.
3. Concept shift: a change in $p(y|x)$. This can occur when the mechanism, or underlying model linking our predictions and features, changes. For example, a model predicting sentiment or emotions on social media can have different mechanisms that affect people that change, such as shifts in political climates or major world events.

2 Covariate shift

Covariate shift is often studied under *domain adaptation*, which studies how to train models that remain robust or accurate when covariate shift occurs. For example, in supervised learning, we first train a model f on training data D_{train}, and then test our model on shifted data D_{test}.



Usually this assume a factorization of the distribution as $p(x, y) = p(x)p(y|x)$. Learning under covariate shifts can be viewed as a form of *causal learning*, i.e. learning the effect, or the underlying causal mechanism $p(y|x)$ that remains correct even when $p(x)$ changes.

$$\begin{aligned} &\int_{x,y} f(x,y)p(x,y)dxdy \\ =&\int_{x,y} f(x,y)p(y|x)p(x)dxdy \\ =&\int_{x,y} f(x,y)p(y|x)\frac{p(x)}{q(x)}q(x)dxdy \\ =&\int_{x,y} f(x,y)p(y|x)\alpha(x)q(x)dxdy \end{aligned} \tag{1}$$

Types of covariate shift can usually be categorized into one of the two:

1. Domain shift: different distributions are organized according several *domains*, i .e. p_i for $i = 1 \dots k$. For example, clipart vs. real photos vs. paintings. In this case, the entire

distribution typically changes. A distribution shift is thus a change from one domain to another, i.e. training on p_1 and testing on p_k.

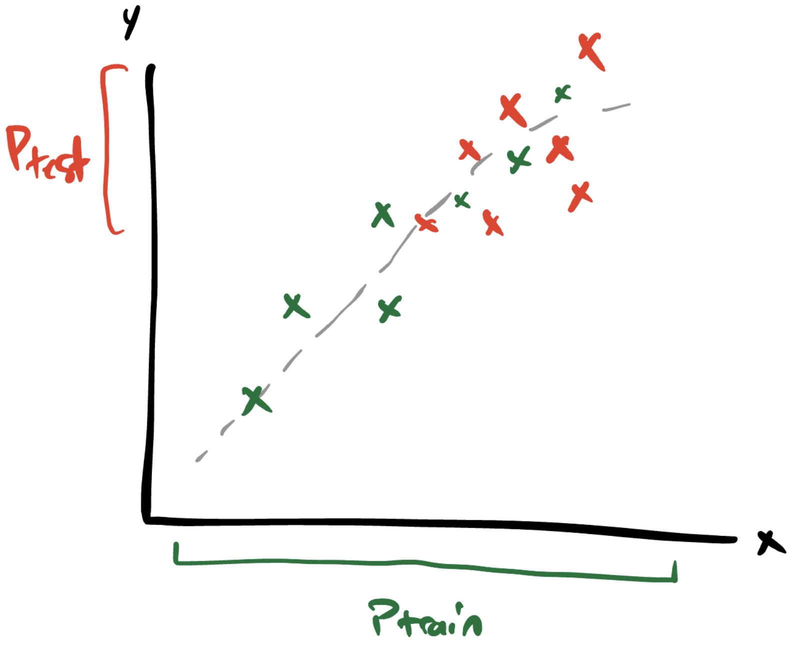
2. Subpopulation shift: the support of the distribution does not change, but the distribution over subgroups or subpopulations does. For example, going from 50% dogs and 50% cats to 90% dogs and 10% cats.

In the future we will discuss how to train models robust to these shifts.

3 Label shift

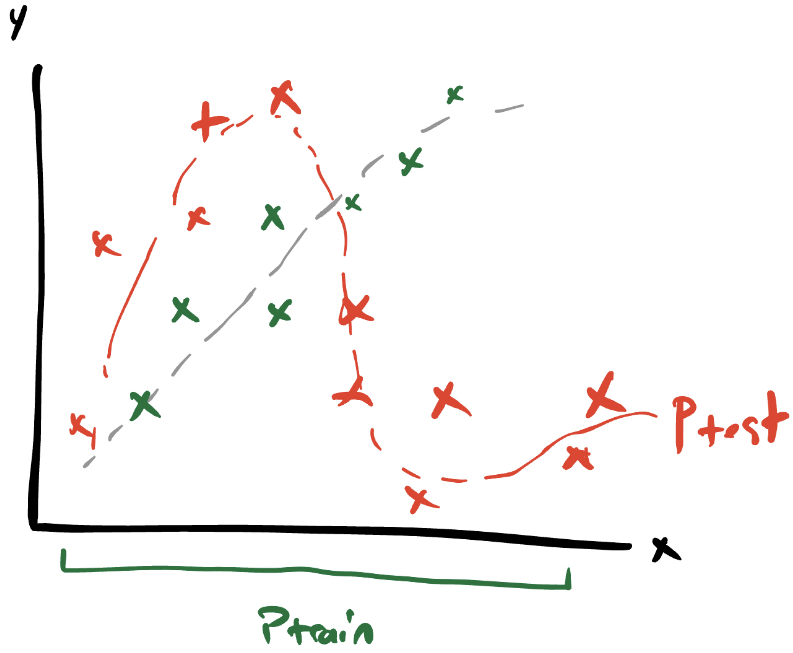
In contrast to covariate shift, assumes that the the label distribution $p(y)$ changes. This is based on a factorization of the distribution as $p(x, y) = p(y)p(x|y)$, which can be viewed as a form of anti-causal learning, i.e. predicting the cause of the label.

$$\begin{aligned}
&\int_{x,y} f(x,y)p(x,y)dxdy \\
&= \int_{x,y} f(x,y)p(x|y)p(y)dxdy \\
&= \int_{x,y} f(x,y)p(x|y)\frac{p(y)}{q(y)}q(y)dxdy \\
&= \int_{x,y} f(x,y)p(x|y)\alpha(y)q(y)dxdy
\end{aligned} \tag{2}$$



4 Concept shift

Finally, concept shift assumes that the mechanism connecting the features to the label changes. This assumes a forward causal relationship as in the covariate setting.



5 Other types of shifts

You may have heard of other kinds of shifts:

- Temporal - changes over time
- Environmental - changes in the environment
- Group - changes over subgroups

For the most part these are an orthogonal property of possible shifts, and could be in addition to any of the previous categories, or in isolation.

6 Detecting distribution s hift

Distribution shift can be monitored anywhere in the ML pipeline—from the data down to the predictions. But how do we determine when this occurs? There are a ton of methods for detecting distribution shift. In these notes we'll cover three statistical methods. These methods test the question "What is the probability that two sets of samples were drawn from the same (unknown) probability distribution"?

6.1 Kolmogorov-Smirnov

The Kolmogorov-Smirnov (KS) test is a nonparametric statistical test. It makes no assumptions, but only works for one-dimensional data. When it is applicable, it is one of the most useful and general tests as it can detect differences in both support and shape of the distribution.

Let's begin with the standard KS test statistic. For a sample $x_1, \ldots, x_n$ drawn from a distribution with CDF $F(x)$, we do the following:

1. Calculate the empirical CDF $F_n(x) = \frac{\sum_i 1(x_i \leq x)}{n}$
2. Calculate the KS statitistic $D_n = \sup_x |F_n(x) - F(x)|$

Via the fundamental theorem of statistics (or the Glivenko-Cantelli theorem), we know that $F_n(x)$ converges almost surely to $F(x)$ if x_i are drawn from $F(x)$ (the null hypothesis). The goodness-of-fit test, which tests of the empirical CDF matches the true CDF rejects the null hypothesis if

$$\sqrt{n} D_n > K_\alpha \tag{3}$$

where $K_\alpha = P(K \leq K_\alpha) = 1 - \alpha$ and K is a random variable drawn from the Kolmogorov distribution. Note that $K = \sup_{[0,1]} |B(t)|$ where $B(t)$ is a Brownian bridge.

To make this a two-sample test, we simply use the empirical CDF of both datasets. If we no longer have the true distribution F but a sample z_i, we can do the following:

1. Calculate the empirical CDF $F_{1,n}(x) = \frac{\sum_i 1(x_i \leq x)}{n}$ and $F_{2,m}(z) = \frac{\sum_i 1(z_i \leq z)}{m}$
2. Calculate the KS statitistic $D_{n,m} = \sup_x |F_{1,n}(x) - F_{2,m}(x)|$

where we reject the null hypothesis if

$$D_{n,m} > c(\alpha) \sqrt{\frac{n+m}{nm}} \tag{4}$$

where $c(\alpha) = \sqrt{-\ln(\alpha/2) \cdot \frac{1}{2}}$.

6.2 Maximum Mean Discrepancy

The MMD is a two sample test statistic that represents distance between distributions as the distance between the mean embedding. It also makes no assumption, and can apply to multivariate data.

The MMD is typically calculated in a reproducing kernel Hilbert space (RKHS). Let $\mathcal{H}$ be a Hilbert space of real valued functions on X. An RKHS is a space of functions where closeness in RKHS implies pointwise closeness in the functions, and that the space has a reproducing property. By the Riesz representation theorem, for all x, there exists a unique $k_x \in \mathcal{H}$ such that

$$f(x) = \langle f, k_x \rangle \tag{5}$$

Then we can say the reproducing kernel is $k(x, z) = \langle k_x, k_x \rangle$.

For two distributions p, q and a feature map ϕ, we have

$$MMD(P, Q) = \|\mathbb{E}_p[\phi(x)] - \mathbb{E}_q[\phi(z)]\|_{\mathcal{H}} \tag{6}$$

To compute this, we can use what is called the kernel trick:

$$\begin{aligned} MMD^2(P, Q) &= \|\mathbb{E}_p[\phi(x)] - \mathbb{E}_q[\phi(z)]\|^2_{\mathcal{H}}c \\ &= \langle \mathbb{E}_p[\phi(x)], \mathbb{E}_p[\phi(x)]\rangle + \langle \mathbb{E}_q[\phi(z)], \mathbb{E}_q[\phi(z)]\rangle - 2\langle \mathbb{E}_q[\phi(z)], \mathbb{E}_p[\phi(x)]\rangle \\ &= \mathbb{E}_{x,x'\sim p}[k(x, x')] + \mathbb{E}_{z,z'\sim q}[k(z, z')] - 2\mathbb{E}_{x,z\sim p,q}[k(x, z)] \end{aligned} \tag{7}$$

One kernel that leads to an RKHS is the Gaussian kernel, $k(x, z) = e^{\frac{-\|x-y\|^2}{2\sigma^2}}$. There is no theoretically nice threshold, however we can bootstrap the distribution of MMD distances under the null hypothesis via resampling to estimate the distribution under the null (also known as permutation testing). Then, accept/reject based on a 95% threshold.

6.3 Least-squares density difference

Here, the goal is to estimate the dfiference between two densities, $f(x) = p(x) - q(x)$. Here we will fit a linear model $g(x)$ where

$$g(x) = \theta^T \psi(x) \tag{8}$$

where $\psi(x)$ is a vector of basis functions, i.e. the Gaussian kernel $\exp(-\|x - c_l\|^2/(2\sigma^2))$ for centers c_l. Then, the optimal linear classifier can be fit by minimizing the least squares difference:

$$\begin{aligned} \min_\theta \int (g(x) - f(x))^2 dx &= \min_\theta \int (g(x)^2 dx - \int g(x) f(x) dx \\ &= \min_\theta \theta^T H\theta - 2\theta^T h \\ &= H^{-1}h \end{aligned} \tag{9}$$

where

$$\begin{aligned} H(x) &= \int \psi(x)\psi(x)^T dx \\ h(x) &= \int \psi(x)p(x)dx - \int \psi(x')q(x')dx' \end{aligned} \tag{10}$$

With this estimator, we can then estimate the least squares difference between the two densities as follows:

$$\begin{aligned}\int (p(x) - q(x))^2 dx &= intf(x)(p(x) - q(x))dx \\ &= intf(x)(p(x) - q(x))dx \\ &= int\theta^T \psi(x)(p(x) - q(x))dx \\ &= \theta^T h \\ &= h^T H^{-1} h\end{aligned} \tag{11}$$

We can then test significance with a permutation test as we did in the MMD setting.

7 References

Parts of these notes are drawn from Chelsea Finn's talk given at the UpML ICML 2022 workshop `https://upml2022.github.io/`.

Parts of these notes are drawn from Chip Huyen's course CS 329S: Machine Learning Systems Design `https://huyenchip.com/2022/02/07/data-distribution-shifts-and-monitoring.html`

Many distribution shift metrics are implemented in the open source alibi detect library `https://github.com/SeldonIO/alibi-detect`