Stanford University

CS 329P : Practical Machine Learning (2021 Fall)

Lecture 7 - Two Sample Tests and Label Shift

Qingqing Huang, Mu Li, Alex Smola

https://c.d2l.ai/stanford-cs329p

Training ≠ Testing

- **Generalization performance**
  (the empirical distribution lies) $p_{\mathrm{emp}}(x, y) \neq p(x, y)$
- **Covariate shift**
  (the covariate distribution lies) $p(x) \neq q(x)$
- **Adversarial data**
  (the support of the distribution lies) $\mathrm{supp}(p) \neq \mathrm{supp}(q)$
- **Label shift**
  (the label distribution lies) **How?** $p(y) \neq q(y)$


AND NOW FOR SOMETHING
COMPLETELY DIFFERENT.

Training ≠ Testing

- **Generalization performance**
(the empirical distribution lies)

$$p_{\text{emp}}(x, y) \neq p(x, y)$$

- **Covariate shift**
(the covariate distribution lies)

$$p(x) \neq q(x)$$

- **Adversarial data**
(the support of the distribution lies)

$$\text{supp}(p) \neq \text{supp}(q)$$

- **Two-Sample Tests**
(distributions don't match)

$$p \neq q$$

- **Label shift**
(the label distribution lies)

$$p(y) \neq q(y)$$

Comparing Distributions

vs.

Two Sample Test

- **Definition**
  Given data $X = \{x_1, \ldots x_m\}$ drawn from p and $X' = \{x'_1, \ldots x'_{m'}\}$ drawn from q test whether $p = q$.
- **Algorithms**
  - Train classifier. If it can distinguish datasets we have $p \neq q$
  - Find biggest difference between expectations via
    $\mathrm{MMD}(p, q) := \sup_{f \in \mathcal{F}} \left[\mathbf{E}_p[f(x)] - \mathbf{E}_q[f(x)] \right]$. If big, we have $p \neq q$
  - Estimate Kullback-Leibler divergence
    $D(p \| q) := \mathbf{E}_p \left[\log p(x) - \log q(x) \right]$

Classifier

```
>>> from autogluon.tabular import TabularPredictor
>>> predictor = TabularPredictor(label=COLUMN_NAME).fit(train_data=TRAIN_DATA.csv)
>>> predictions = predictor.predict(TEST_DATA.csv)
```

Classifier - Gory Math

- Classifier objective

$$\mathbf{E}_p[\log \pi(y=1|x)] + \mathbf{E}_q[\log \pi(y=-1|x)]$$

is minimized for $\pi(y=1|x) = \dfrac{p(x)}{p(x)+q(x)}$.

- Plugging this into the objective yields

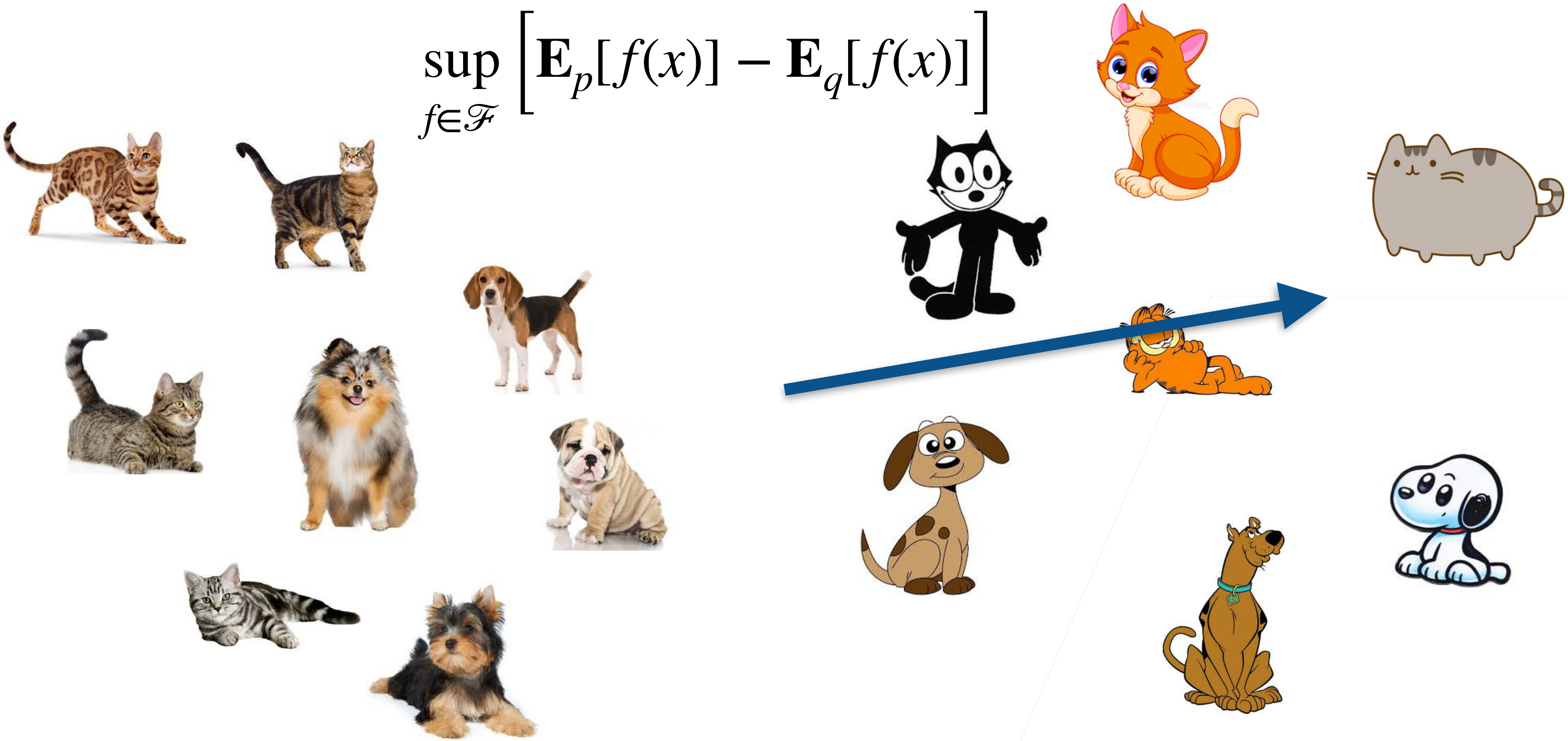
$$\mathbf{E}_p[\log p(x) - \log(p(x)+q(x))] + \mathbf{E}_q[\log q(x) - \log(p(x)+q(x))]$$

$$= 2H[(p+q)/2] - H[p] - H[q] + 2\log 2$$

- By convexity of entropy minimized for $p = q$

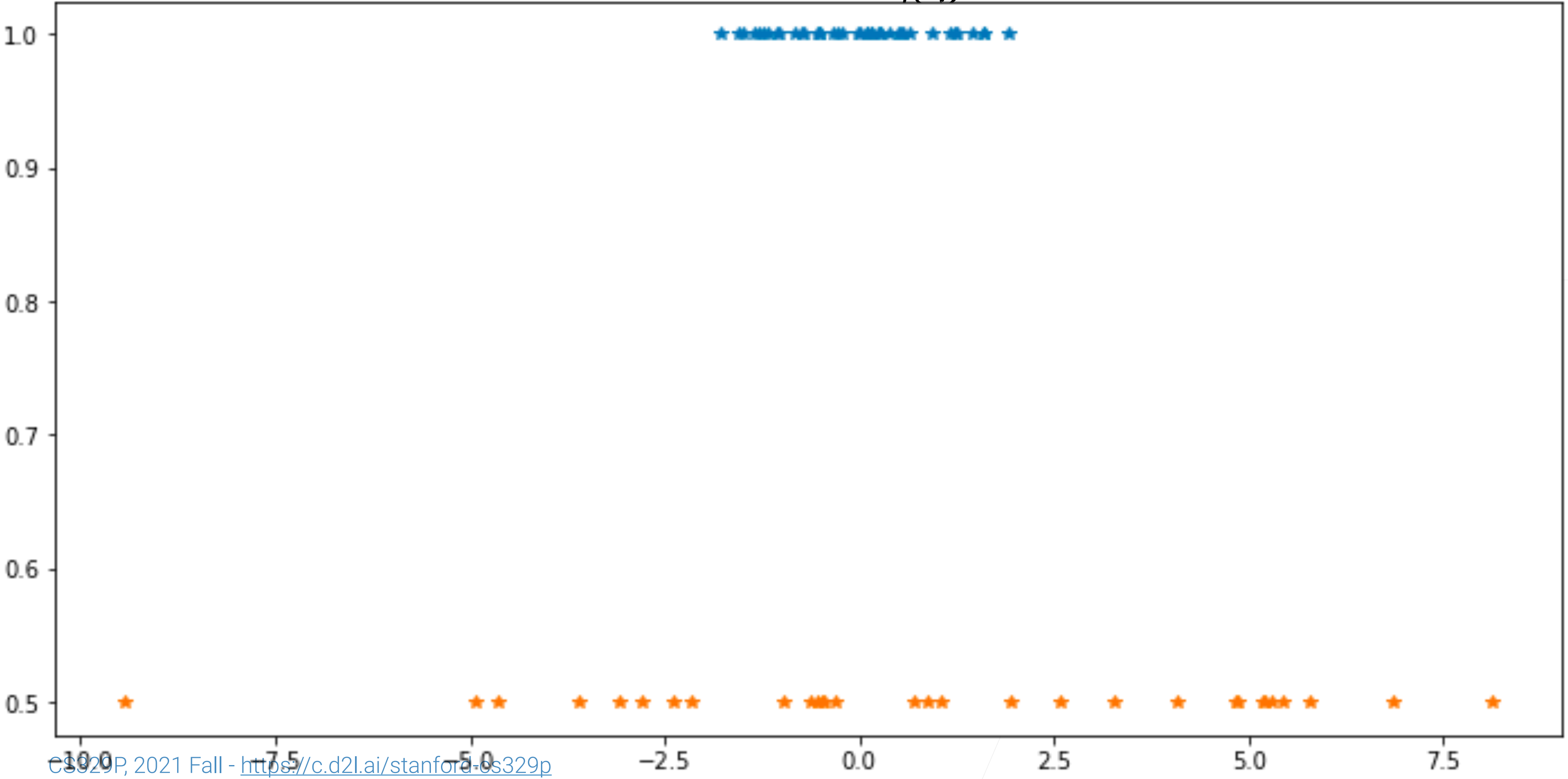
Maximum Mean Discrepancy

$$\sup_{f \in \mathcal{F}} \left[\mathbf{E}_p[f(x)] - \mathbf{E}_q[f(x)] \right]$$



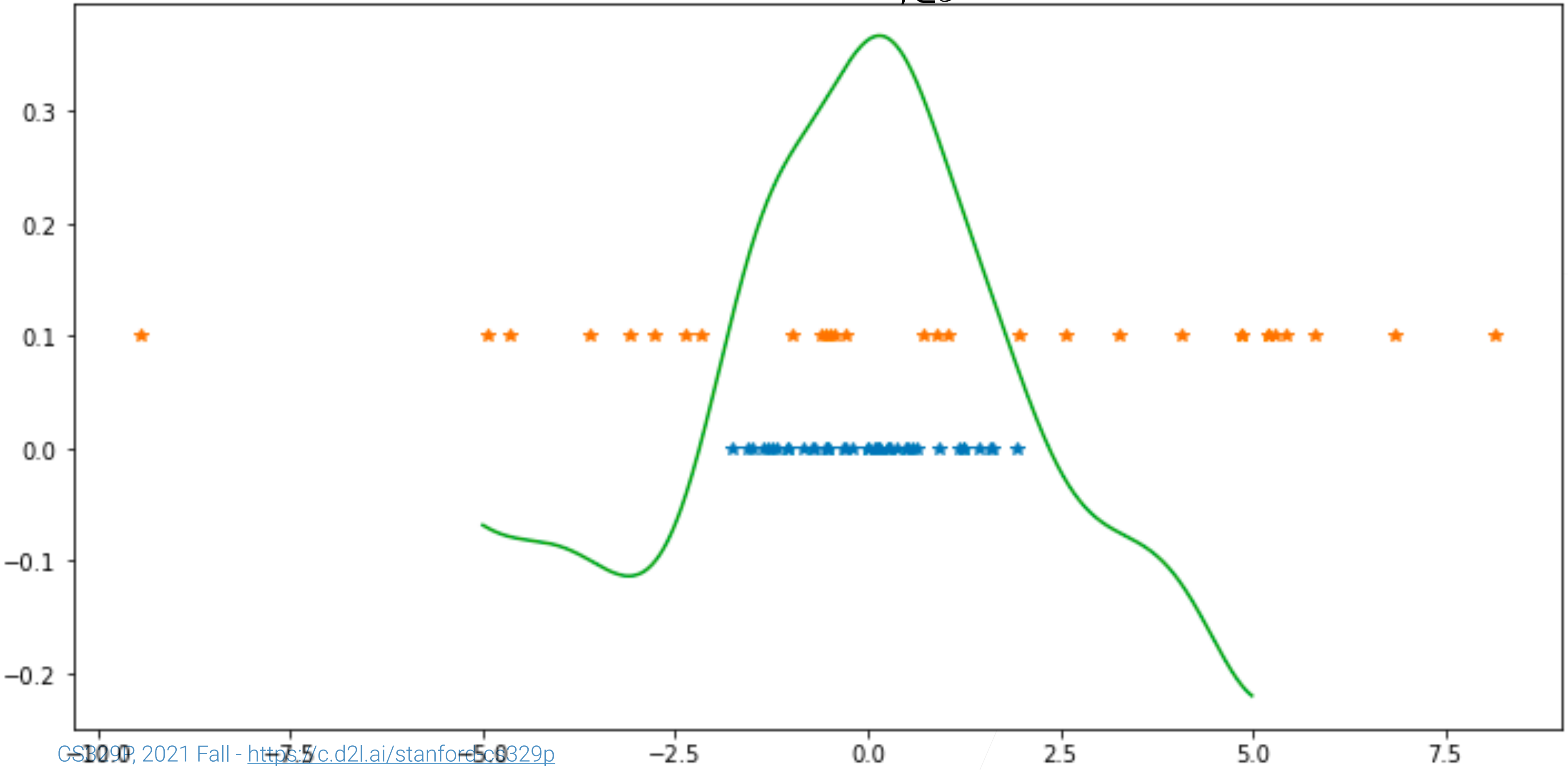
Maximum Mean Discrepancy

$$\sup_{f \in \mathcal{F}} \left[\mathbf{E}_p[f(x)] - \mathbf{E}_q[f(x)] \right]$$



Maximum Mean Discrepancy

$$\sup_{f \in \mathcal{F}} \left[\mathbf{E}_p[f(x)] - \mathbf{E}_q[f(x)] \right]$$



Maximum Mean Discrepancy - Gory Math

- Find function with largest difference in expectation between two distributions

$$\sup_{f \in \mathscr{F}} \left[\mathbf{E}_p[f(x)] - \mathbf{E}_q[f(x)] \right]$$

- For linear functions (in Banach space) this is

$$\sup_{\|w\| \leq 1} \left[\mathbf{E}_p[\langle \phi(x), w \rangle] - \mathbf{E}_q[\langle \phi(x), w \rangle] \right] =$$

$$\sup_{\|w\| \leq 1} \left\langle \mathbf{E}_p[\phi(x)] - \mathbf{E}_q[\phi(x)], w \right\rangle = \left\| \mathbf{E}_p[\phi(x)] - \mathbf{E}_q[\phi(x)] \right\|_*$$

Maximum Mean Discrepancy - More Gory Math

- Using kernels (Reproducing Kernel Hilbert Space)

$$k(x, x') = \langle \phi(x), \phi(x') \rangle$$

- Discriminant function (adversary)

$$f(x') = \left\langle \mathbf{E}_p[\phi(x)] - \mathbf{E}_q[\phi(x)], \phi(x') \right\rangle = \mathbf{E}_p[k(x, x')] - \mathbf{E}_q[k(x, x')]$$

- On finite sample

$$f(x) = \frac{1}{m} \sum_{i=1}^{m} k(x_i, x) - \frac{1}{m'} \sum_{i=1}^{m'} k(x_i', x)$$

$$\frac{1}{m(m-1)} \sum_{i \neq j} \left(k(x_i, x_j) + k(x_i', x_j') - k(x_i, x_j') - k(x_i', x_j) \right)$$

Maximum Mean Discrepancy - More Gory Math

- Using kernels (Reproducing Kernel Hilbert Space)

$$k(x, x') = \langle \phi(x), \phi(x') \rangle$$

- Discriminant function (adversary)

$$f(x') = \Big\langle \mathbf{E}_p[\phi(x)] - \mathbf{E}_q[\phi(x)], \phi(x') \Big\rangle = \mathbf{E}_p[k(x, x')] - \mathbf{E}_q[k(x, x')]$$

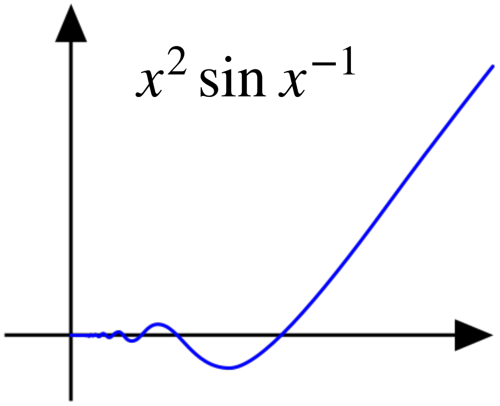
```
p = torch.randn(100)
q = torch.randn(100) * 4.0 + 0.4
x = torch.arange(-3,3,0.01)
k = gpytorch.kernels.RBFKernel()
f = k(x,p)@wp - k(x,q)@wq
```

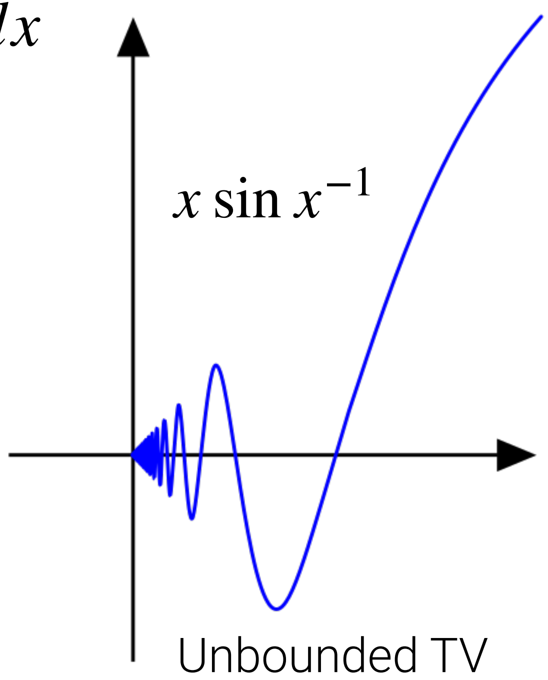
Kolmogorov Smirnov Statistic

- Functions with bounded total variation

$$\mathrm{TV}[f] := \int \left| \partial_x f(x) \right| dx$$

- Examples





Kolmogorov Smirnov Statistic

- Functions with bounded total variation

$$\mathrm{TV}[f] := \int \left|\partial_x f(x)\right| dx$$

- Maximizing the statistic

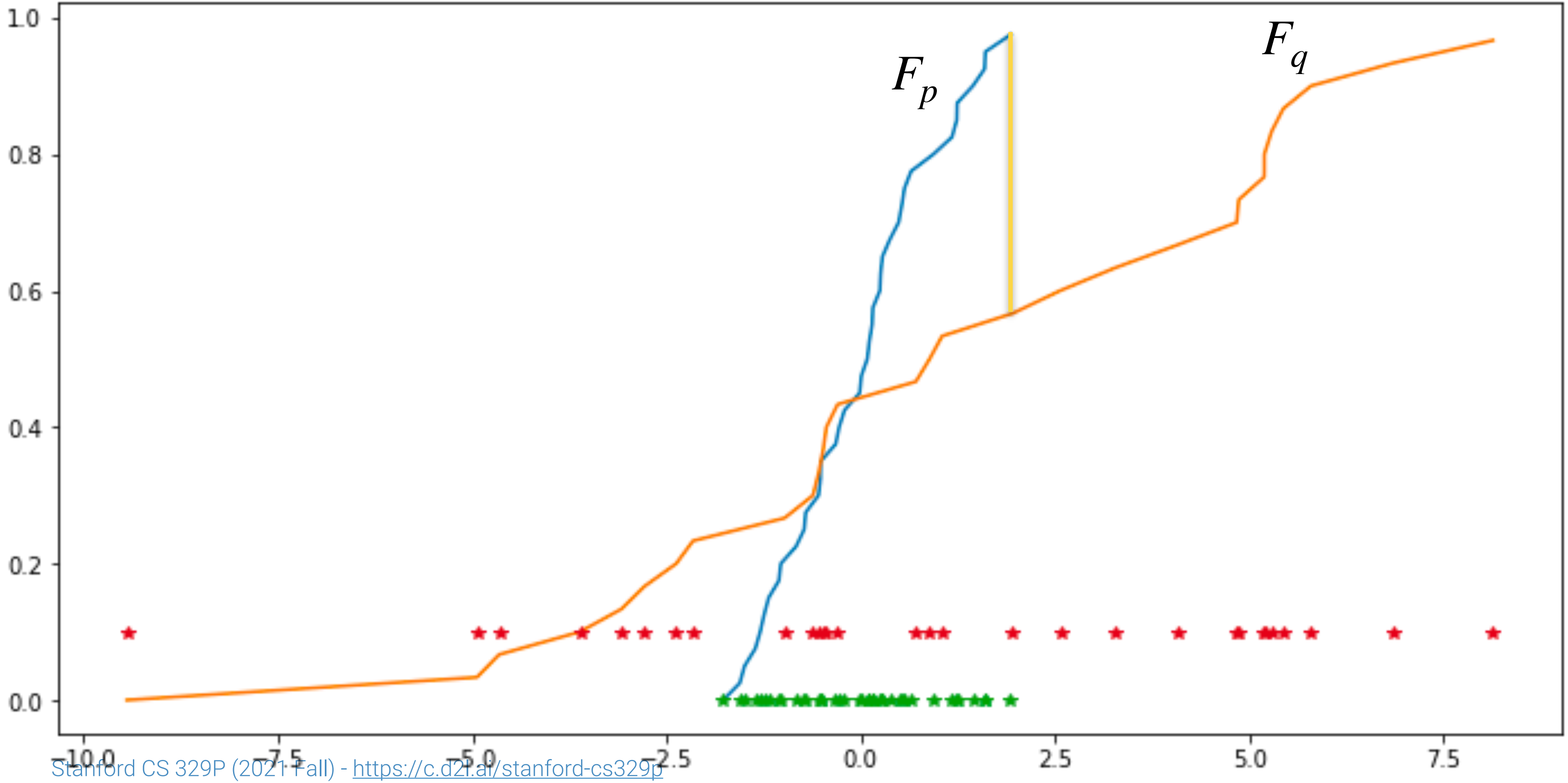
$$\sup_{\mathrm{TV}[f] \leq 1} \left[\mathbf{E}_p[f(x)] - \mathbf{E}_q[f(x)]\right] =$$

Cumulative Distribution Function

$$\sup_z \left|\mathbf{E}_p[\{x \leq z\}] - \mathbf{E}_q[\{x \leq z\}]\right| = \|F_p - F_q\|_\infty$$

$$F_p[z] = \int_{-\infty}^{z} p(x)dx$$

Kolmogorov Smirnov Statistic



Key Takeaways

Two sample tests

- Check whether X and X' are drawn from same distribution
- Tests
  - **Train classifier, if it works, the samples are different** (choose this one)
  - **Maximum Mean Discrepancy** (easy to generate discriminator without training)
  - **Kolmogorov Smirnov Test** (works great for 1D data)

Sanity check to confirm that distributions match!

Training ≠ Testing

- **Generalization performance**
  (the empirical distribution lies) $p_{\mathrm{emp}}(x, y) \neq p(x, y)$
- **Covariate shift**
  (the covariate distribution lies) $p(x) \neq q(x)$
- **Adversarial data**
  (the support of the distribution lies) $\mathrm{supp}(p) \neq \mathrm{supp}(q)$
- **Two-Sample Tests**
  (distributions don't match) $p \neq q$
- **Label shift**
  (the label distribution lies) $p(y) \neq q(y)$