Lecture 4: Distribution Shift

CIS 7000: Trustworthy Machine Learning

Spring 2024

Agenda

- **Robustness to distribution shift**
  - Basic examples
  - Definitions
  - Unsupervised domain adaptation setting

- **Algorithms for distributional robustness**
  - Importance weighting
  - Application to label shift
  - Application to covariate shift

Importance Weights for Covariate Shift

- In the covariate shift setting, we have

$$w(x, y)$$

Importance Weights for Covariate Shift

- In the covariate shift setting, we have

$$
\begin{aligned}
w(x,y) &= \frac{q(x,y)}{p(x,y)} \\
&= \frac{q(y|x)q(x)}{p(y|x)p(x)} \\
&= \frac{q(x)}{p(x)} \\
&\coloneqq w(x)
\end{aligned}
$$

Importance Weights for Covariate Shift

- If we know $w(x)$, then we have

$$\mathbb{E}_Q[\ell(\theta; x, y)]$$

Importance Weights for Covariate Shift

- If we know $w(x)$, then we have

$$\mathbb{E}_Q[\ell(\theta; x, y)] = \mathbb{E}_P[\ell(\theta; x, y) \cdot w(x, y)]$$

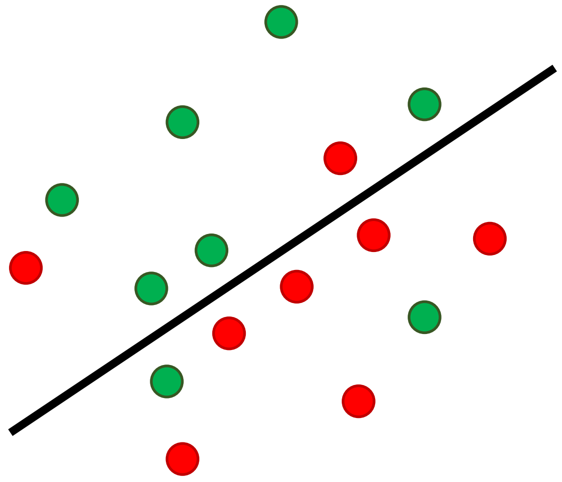
Importance Weights for Covariate Shift

- If we know $w(x)$, then we have

$$\mathbb{E}_Q[\ell(\theta; x, y)] = \mathbb{E}_P[\ell(\theta; x, y) \cdot w(x, y)] = \mathbb{E}_P[\ell(\theta; x, y) \cdot w(x)]$$

Importance Weights for Covariate Shift

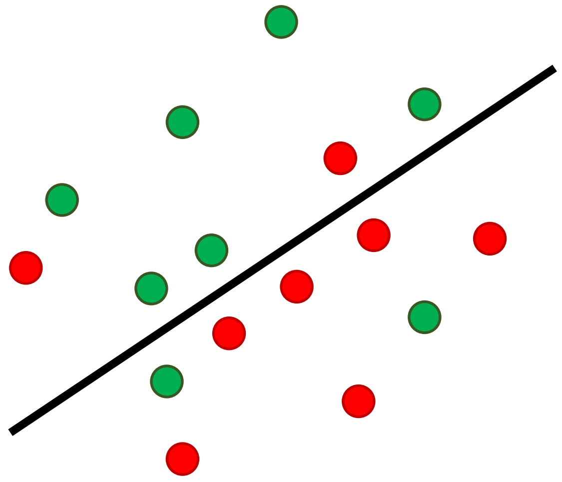
- Define a new distribution R over $\{0,1\} \times \mathcal{X}$:
  - Sample $b \sim \text{Bernoulli}\left(\frac{1}{2}\right)$
  - If $b = 0$, then sample $(x, \cdot) \sim P$
  - If $b = 1$, then sample $(x, \cdot) \sim Q$



- Suppose we know $r(b \mid x)$

Importance Weights for Covariate Shift

- Define a new distribution R over $\{0,1\} \times \mathcal{X}$:
  - Sample $b \sim \mathrm{Bernoulli}\left(\frac{1}{2}\right)$
  - If $b = 0$, then sample $(x, \cdot) \sim P$
  - If $b = 1$, then sample $(x, \cdot) \sim Q$

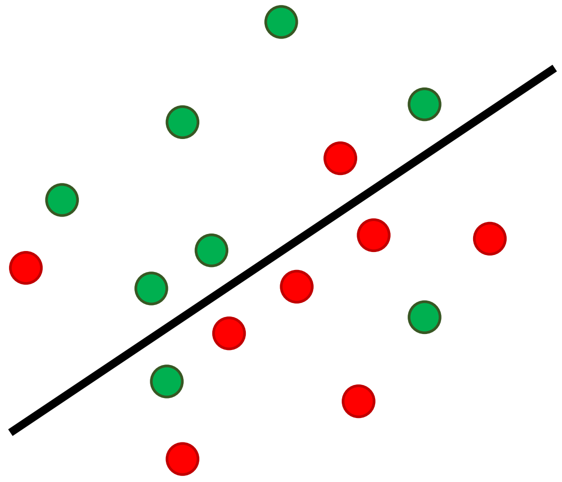


- Suppose we know $r(\, b \mid x \,)$, then by Bayes' rule, we have

$$r(\, b = 0 \mid x \,)$$

Importance Weights for Covariate Shift

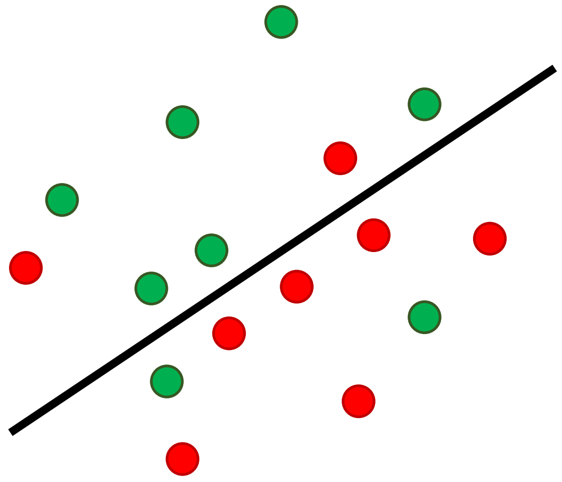
- Define a new distribution R over $\{0,1\}\times\mathcal{X}$:
  - Sample $b \sim \text{Bernoulli}\left(\frac{1}{2}\right)$
  - If $b = 0$, then sample $(x,\cdot) \sim P$
  - If $b = 1$, then sample $(x,\cdot) \sim Q$



- Suppose we know $r(\, b \mid x\,)$, then by Bayes' rule, we have

$$r(\, b = 0 \mid x\,) = \frac{r(\, x \mid b = 0\,)r(b = 0)}{r(\, x \mid b = 0\,)r(b = 0) + r(\, x \mid b = 1\,)r(b = 1)}$$

Importance Weights for Covariate Shift



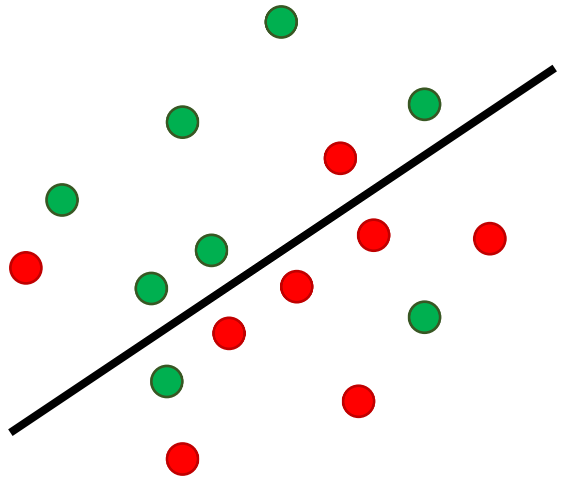
- Define a new distribution R over $\{0,1\} \times \mathcal{X}$:
  - Sample $b \sim \mathrm{Bernoulli}\left(\frac{1}{2}\right)$
  - If $b = 0$, then sample $(x, \cdot) \sim P$
  - If $b = 1$, then sample $(x, \cdot) \sim Q$

- Suppose we know $r(\, b \mid x \,)$, then by Bayes' rule, we have

$$r(\, b = 0 \mid x \,) = \frac{r(\, x \mid b = 0 \,) r(b = 0)}{r(\, x \mid b = 0 \,) r(b = 0) + r(\, x \mid b = 1 \,) r(b = 1)}$$

Importance Weights for Covariate Shift

- Define a new distribution R over $\{0,1\} \times \mathcal{X}$:
  - Sample $b \sim \mathrm{Bernoulli}\left(\frac{1}{2}\right)$
  - If $b = 0$, then sample $(x, \cdot) \sim P$
  - If $b = 1$, then sample $(x, \cdot) \sim Q$

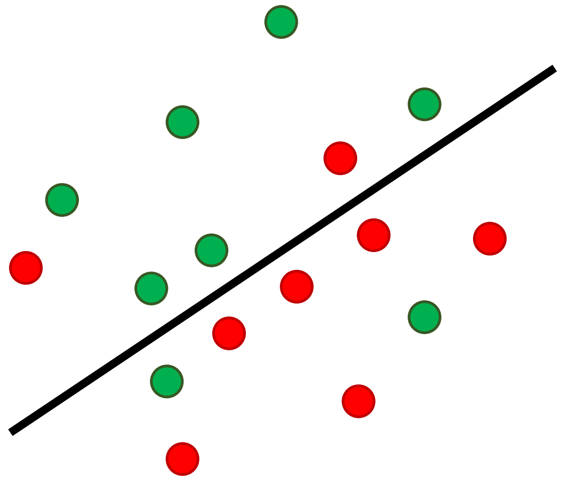


- Suppose we know $r(\, b \mid x \,)$, then by Bayes' rule, we have

$$r(\, b = 0 \mid x \,) = \frac{r(\, x \mid b = 0 \,) \cdot \frac{1}{2}}{r(\, x \mid b = 0 \,) \cdot \frac{1}{2} + r(\, x \mid b = 1 \,) \cdot \frac{1}{2}}$$

Importance Weights for Covariate Shift

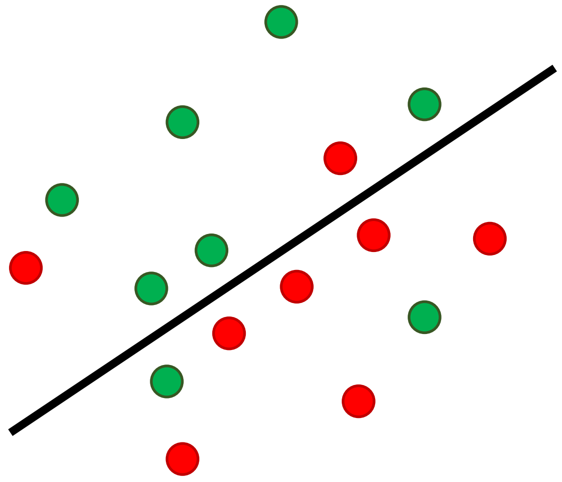
- Define a new distribution R over $\{0,1\} \times \mathcal{X}$:
  - Sample $b \sim \mathrm{Bernoulli}\left(\frac{1}{2}\right)$
  - If $b = 0$, then sample $(x, \cdot) \sim P$
  - If $b = 1$, then sample $(x, \cdot) \sim Q$



- Suppose we know $r(\, b \mid x\,)$, then by Bayes' rule, we have

$$r(\, b = 0 \mid x\,) = \frac{r(\, x \mid b = 0\,)}{r(\, x \mid b = 0\,) + r(\, x \mid b = 1\,)}$$

Importance Weights for Covariate Shift

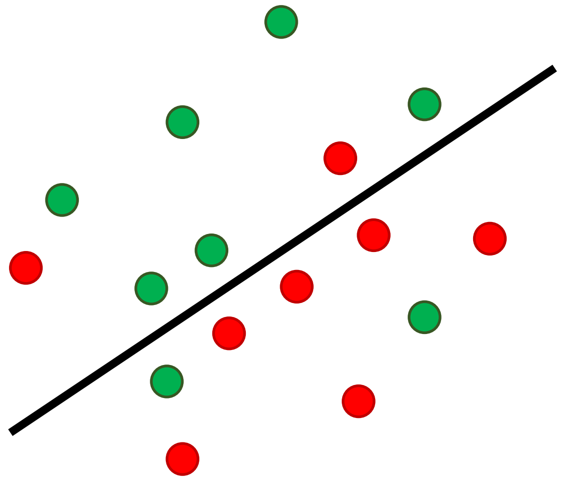


- Define a new distribution R over $\{0,1\}\times\mathcal{X}$:
  - Sample $b \sim \mathrm{Bernoulli}\left(\frac{1}{2}\right)$
  - If $b = 0$, then sample $(x,\cdot) \sim P$
  - If $b = 1$, then sample $(x,\cdot) \sim Q$

- Suppose we know $r(\, b \mid x \,)$, then by Bayes' rule, we have

$$r(\, b = 0 \mid x \,) = \frac{p(x)}{p(x) + q(x)}$$

Importance Weights for Covariate Shift



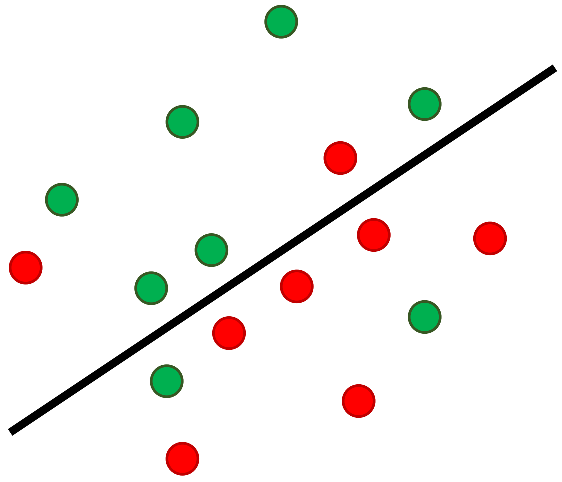
- Define a new distribution R over $\{0,1\} \times \mathcal{X}$:
  - Sample $b \sim \text{Bernoulli}\left(\frac{1}{2}\right)$
  - If $b = 0$, then sample $(x, \cdot) \sim P$
  - If $b = 1$, then sample $(x, \cdot) \sim Q$

- Suppose we know $r(\, b \mid x \,)$, then by Bayes' rule, we have

$$r(\, b = 0 \mid x \,) = \frac{1}{1 + \frac{q(x)}{p(x)}}$$

Importance Weights for Covariate Shift

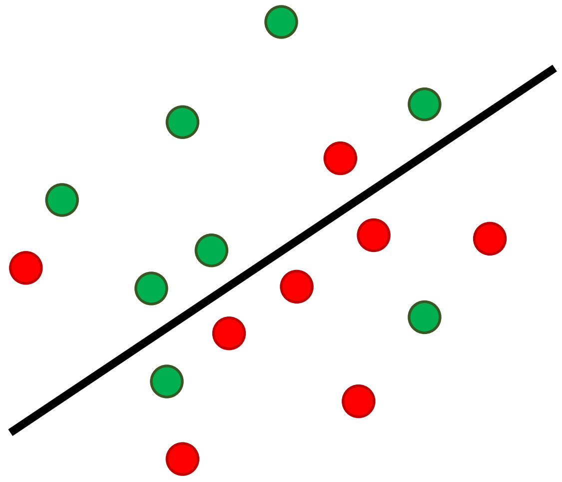
- Define a new distribution R over $\{0,1\}\times\mathcal{X}$:
  - Sample $b \sim \mathrm{Bernoulli}\left(\frac{1}{2}\right)$
  - If $b = 0$, then sample $(x,\cdot) \sim P$
  - If $b = 1$, then sample $(x,\cdot) \sim Q$



- Suppose we know $r(\, b \mid x \,)$, then by Bayes' rule, we have

$$r(\, b = 0 \mid x \,) = \frac{1}{w(x) + 1}$$

Importance Weights for Covariate Shift



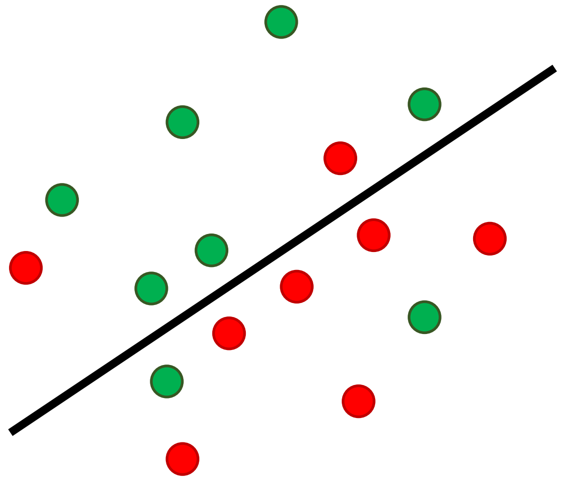
- Define a new distribution R over $\{0,1\}\times\mathcal{X}$:
  - Sample $b \sim \mathrm{Bernoulli}\left(\frac{1}{2}\right)$
  - If $b = 0$, then sample $(x,\cdot) \sim P$
  - If $b = 1$, then sample $(x,\cdot) \sim Q$

- Suppose we know $r(\,b \mid x\,)$, then by Bayes' rule, we have

$$w(x) + 1 = \frac{1}{r(\,b = 0 \mid x\,)}$$

Importance Weights for Covariate Shift

- Define a new distribution R over $\{0,1\}\times\mathcal{X}$:
  - Sample $b \sim \mathrm{Bernoulli}\left(\frac{1}{2}\right)$
  - If $b = 0$, then sample $(x,\cdot) \sim P$
  - If $b = 1$, then sample $(x,\cdot) \sim Q$



- Suppose we know $r(\, b \mid x\,)$, then by Bayes' rule, we have

$$w(x) = \frac{1}{r(\, b = 0 \mid x\,)} - 1$$

Estimating Source-Target Probability

- We can construct a dataset of i.i.d. samples $(x, b) \sim R$
  - For simplicity, assume that $|X| = |Z|$
  - Then, consider

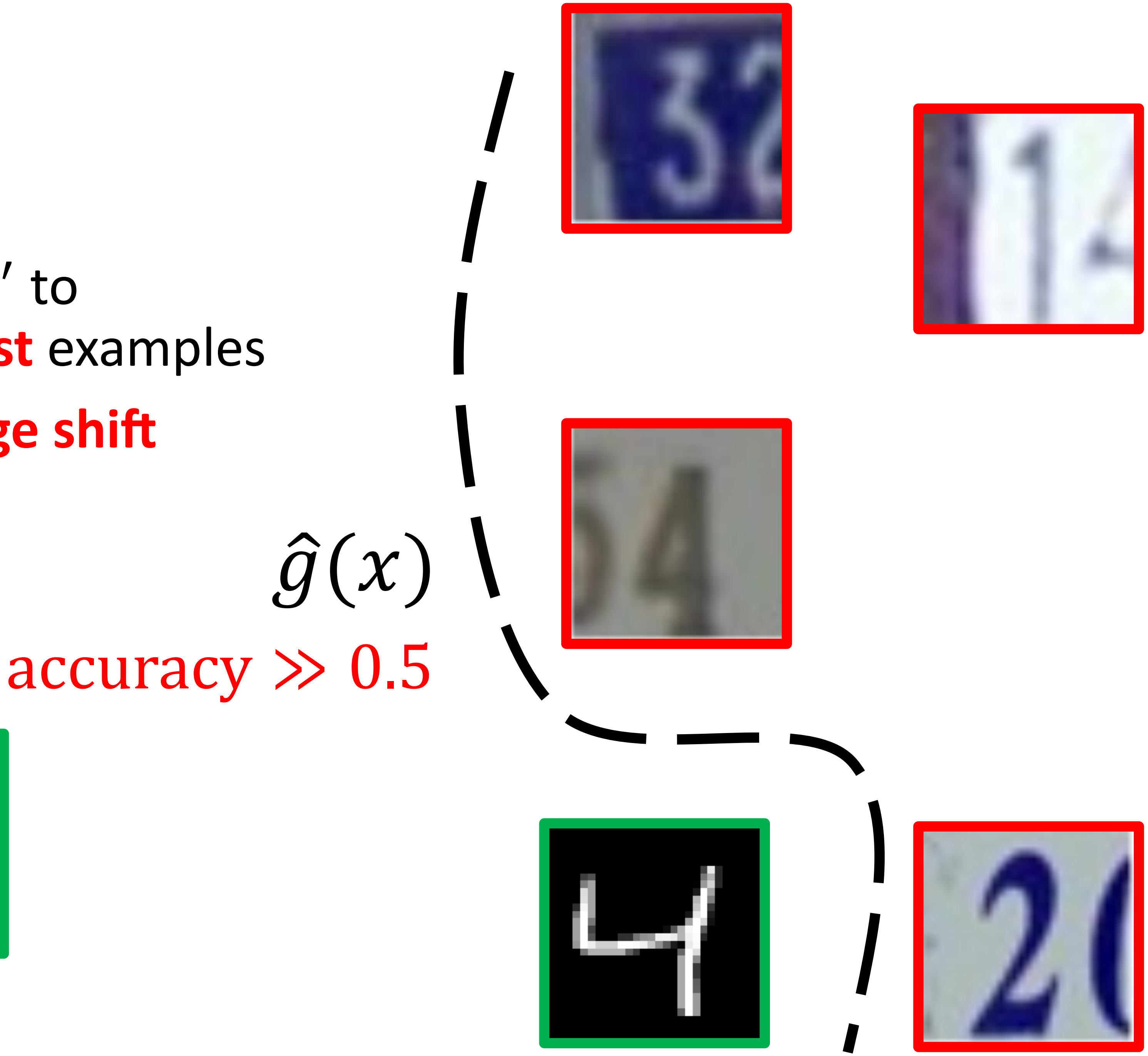
$$X' = \{ (x, 0) \mid (x, y) \in Z \} \cup \{ (x, 1) \mid x \in X \}$$

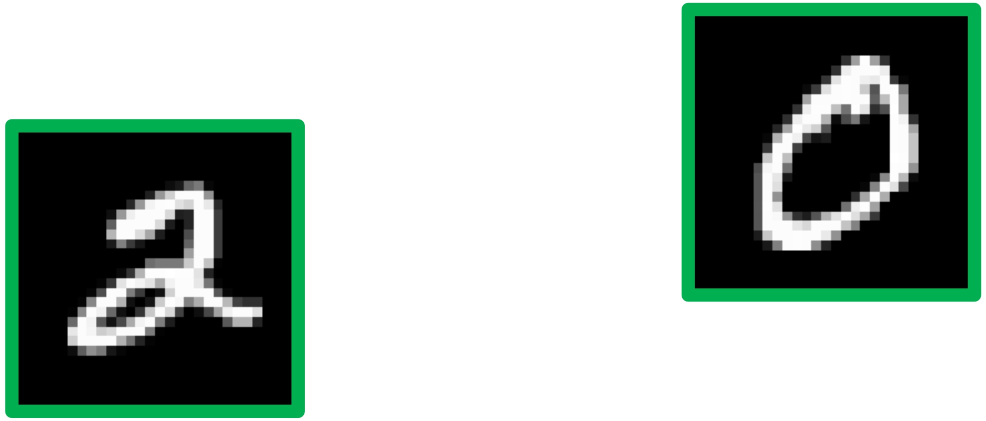
  - This dataset consists of i.i.d. samples $(x, b) \sim R$

- Given i.i.d. samples $(x, b) \sim R$, then $r(b = 1 \mid x)$ is the same as the probability of "label" b given "input" x
  - **Idea:** Train a model (called a **discriminator**) on X' to predict b given x

Discriminators

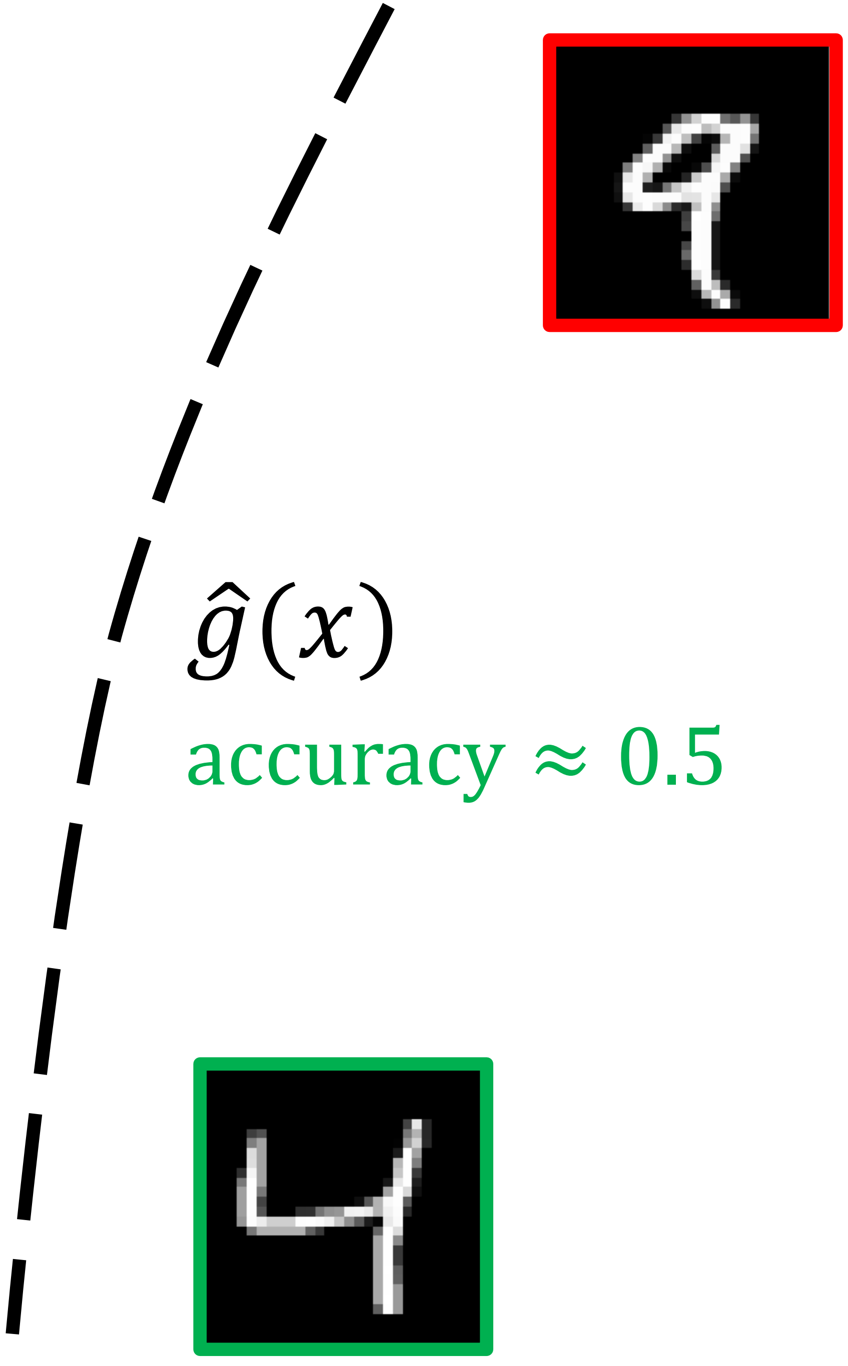
- Train **discriminator** $\hat{g}$ on X' to distinguish **training** and **test** examples
- $\hat{g}$ has **high accuracy** $\Rightarrow$ **large shift**





Discriminators

- Train **discriminator** $\hat{g}$ on X' to distinguish **training** and **test** examples
- $\hat{g}$ has **high accuracy** $\Rightarrow$ **large shift**
- $\hat{g}$ has **low accuracy** $\Rightarrow$ **small shift** (assuming sufficient capacity)



Supervised Learning with Covariate Shift

- **Input:** Training dataset Z, unlabeled test dataset X
- **Step 1:** Construct $X' = \{ (x, 0) \mid (x, y) \in Z \} \cup \{ (x, 1) \mid x \in X \}$ and train $\hat{g}$ on X' to predict b given x
- **Step 2:** Compute $w(x) = \frac{1}{\hat{g}(b=1|x)} - 1$
- **Step 3:** Compute $\hat{\theta} = \arg\min_{\theta} \sum_{(x,y)\in Z} \ell(\theta; x, y) \cdot w(x)$

Importance Weights

- **Pros:**
  - Principled technique for addressing distribution shift
  - “Granular” quantification of shift (obtain an estimate of the shift for each example, not just just the overall shift)

- **Cons:**
  - Does not work when support of Q is not contained in support of P
  - Even if the above is satisfied, importance weights are large if $P(x, y)$ is small

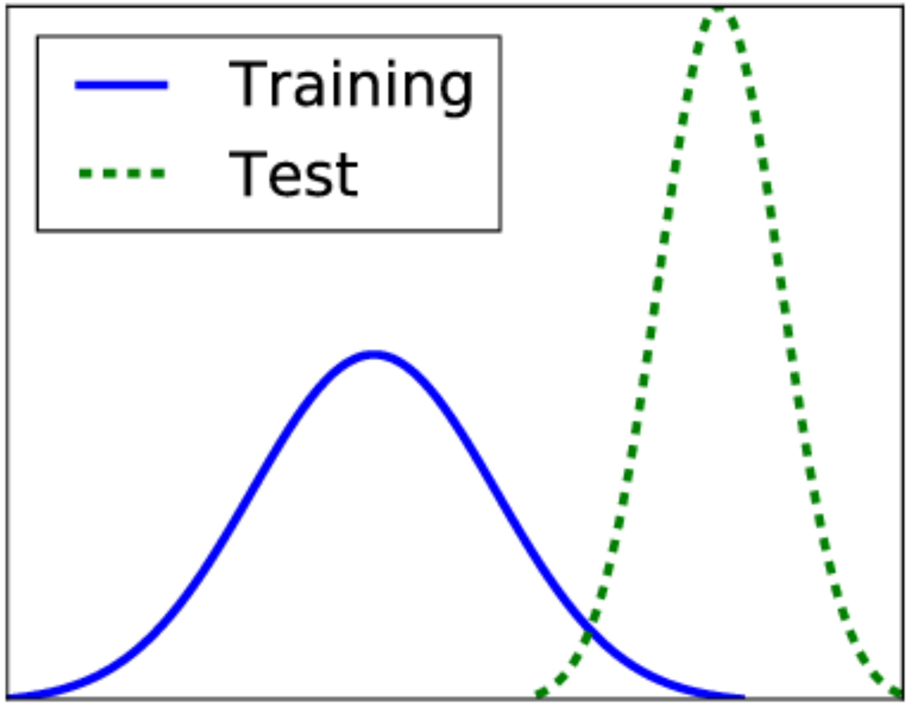
Agenda

- **Robustness to distribution shift**
  - Basic examples
  - Definitions
  - Unsupervised domain adaptation setting

- **Algorithms for distributional robustness**
  - Importance weighting
  - Application to label shift
  - Application to covariate shift

Support of Shifted Data

- **Assumption:** Support of Q is not contained in support of P
- However, this is **necessary** since we do not know anything about data outside of the support of P
- Need additional assumptions to do better
  - Focus on covariate shift

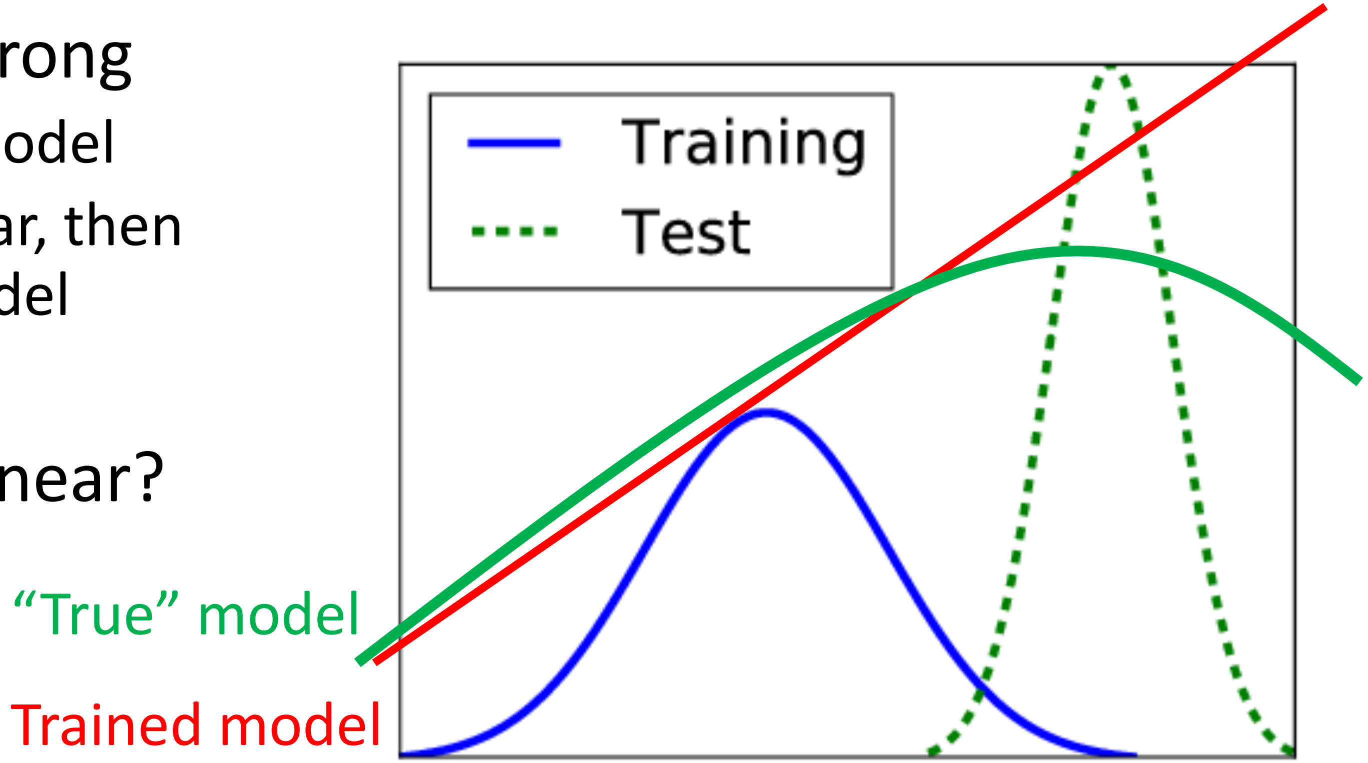


Image; Glauner et al., 2018

Support of Shifted Data

- Closer look at what goes wrong
  - Suppose we train a linear model
  - If the true model is nonlinear, then it may diverge from our model

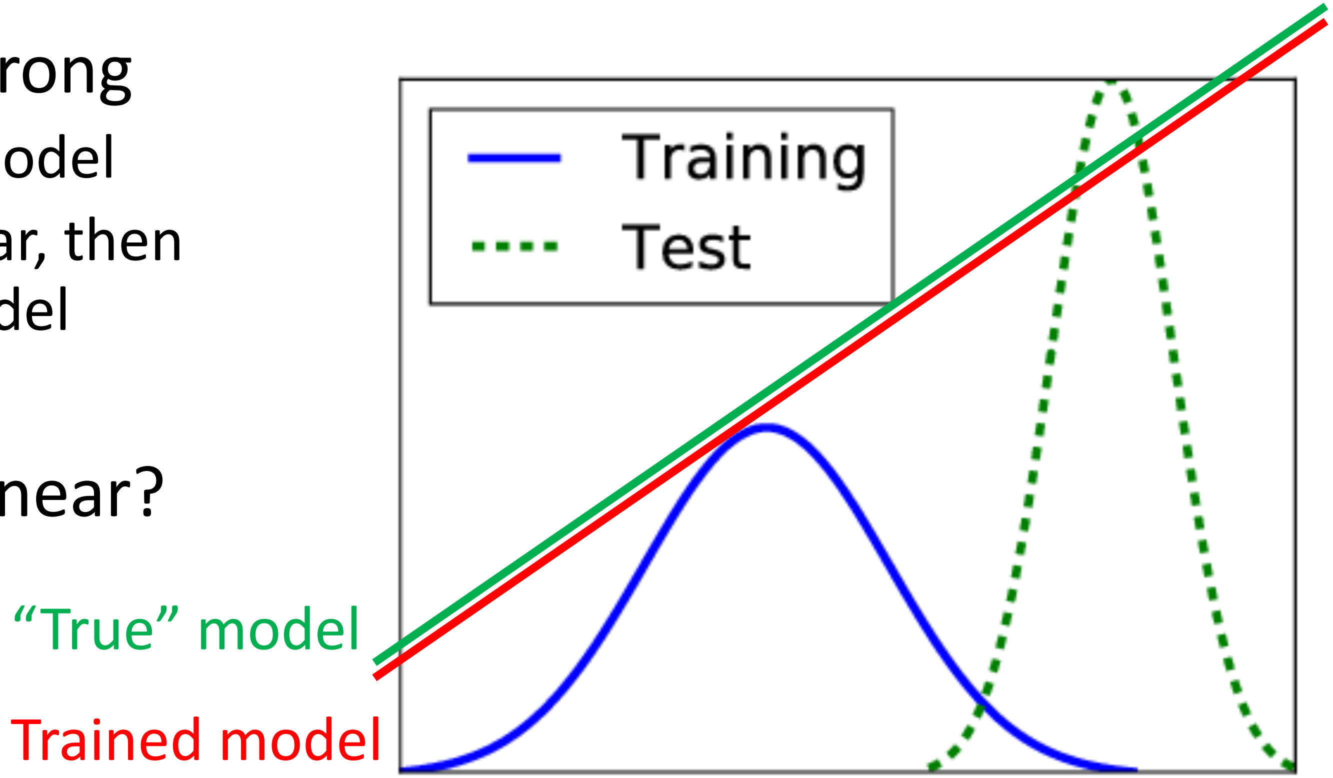
- What if the true model is linear?



Image; Glauner et al., 2018

Support of Shifted Data

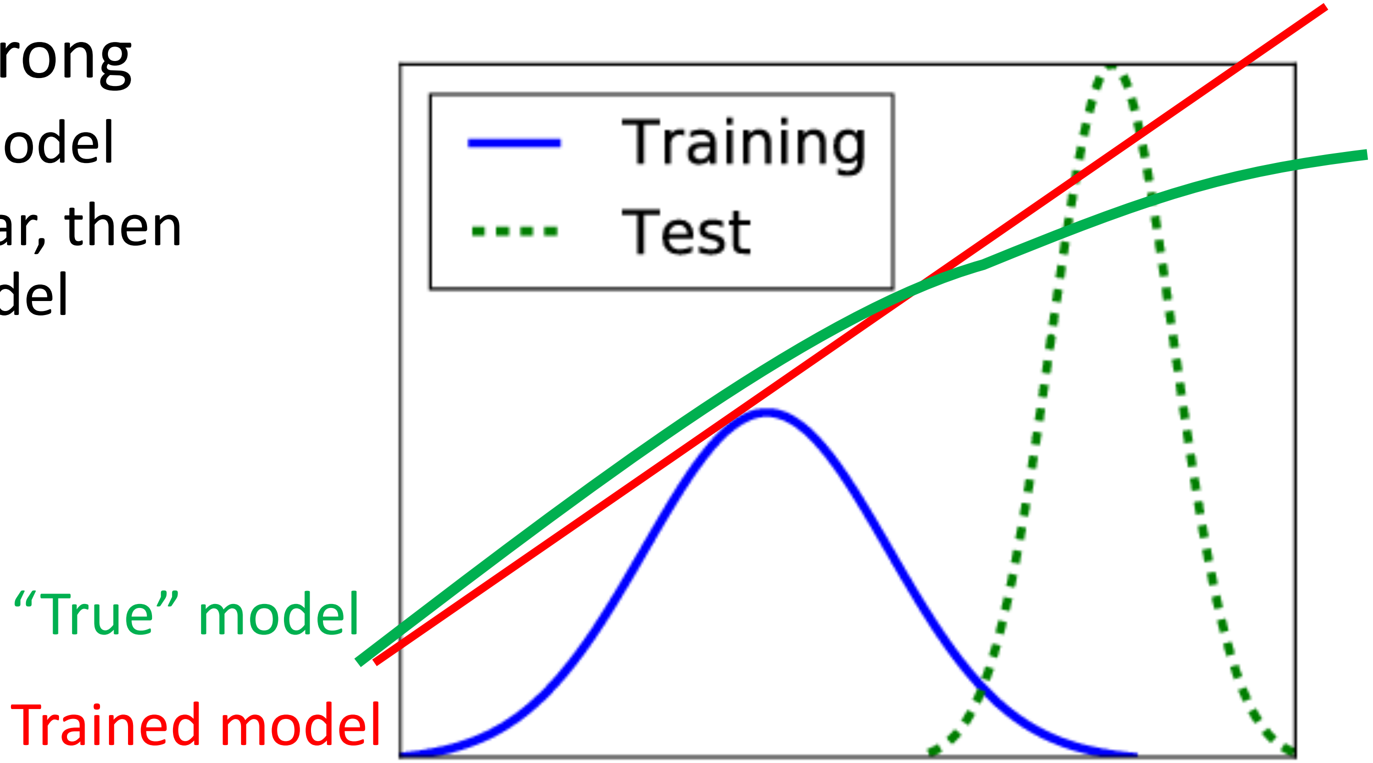
- Closer look at what goes wrong
  - Suppose we train a linear model
  - If the true model is nonlinear, then it may diverge from our model
- What if the true model is linear?
  - Everything is OK!
  - “Well-specified”
  - Rarely holds in practice



Image; Glauner et al., 2018

Support of Shifted Data

- Closer look at what goes wrong
  - Suppose we train a linear model
  - If the true model is nonlinear, then it may diverge from our model
- What is the true model is approximately linear?
  - OK if a “little” off
  - Can we use this fact?



Image; Glauner et al., 2018

Learning vs. Evaluation

- For this part, we will focus on **model evaluation**
  - **Learning:** Optimize $\mathbb{E}_Q[\ell(\theta; x, y)]$
  - **Evaluation:** Estimate $\mathbb{E}_Q[\ell(\theta; x, y)]$

- We will see why learning is harder later

Integral Probability Metrics

- The **total variation distance** is

$$\mathrm{TV}(P,Q) = \int_{\mathcal{X}\times\mathcal{Y}} |q(x,y) - p(x,y)| \cdot dx \cdot dy$$

- The **Wasserstein distance** is

$$W(P,Q) = \sup_{f:K_f\le 1} \int_{\mathcal{X}\times\mathcal{Y}} f(x,y) \cdot \big(q(x,y) - p(x,y)\big) \cdot dx \cdot dy$$

Evaluation Bounds

- Note that

$$\mathbb{E}_Q[\ell(\theta; x, y)]$$

Evaluation Bounds

- Note that

$$\begin{aligned}
\mathbb{E}_Q[\ell(\theta;x,y)] &= \int_{\mathcal{X}\times\mathcal{Y}} \ell(\theta;x,y)\cdot q(x,y)\cdot dx\cdot dy \\
&= \int_{\mathcal{X}\times\mathcal{Y}} \ell(\theta;x,y)\cdot \big(p(x,y)+q(x,y)-p(x,y)\big)\cdot dx\cdot dy \\
&= \int_{\mathcal{X}\times\mathcal{Y}} \ell(\theta;x,y)\cdot p(x,y)\cdot dx\cdot dy \\
&\qquad + \int_{\mathcal{X}\times\mathcal{Y}} \ell(\theta;x,y)\cdot \big(q(x,y)-p(x,y)\big)\cdot dx\cdot dy \\
&= \mathbb{E}_P[\ell(\theta;x,y)] + \int_{\mathcal{X}\times\mathcal{Y}} \ell(\theta;x,y)\cdot \big(q(x,y)-p(x,y)\big)\cdot dx\cdot dy \\
&\leq \mathbb{E}_P[\ell(\theta;x,y)] + \ell_{\max}\cdot \int_{\mathcal{X}\times\mathcal{Y}} |q(x,y)-p(x,y)|\cdot dx\cdot dy \\
&= \mathbb{E}_P[\ell(\theta;x,y)] + \ell_{\max}\cdot \mathrm{TV}(P,Q)
\end{aligned}$$

Evaluation Bounds

- Note that

$$
\begin{aligned}
\mathbb{E}_Q[\ell(\theta; x, y)] &= \int_{\mathcal{X}\times\mathcal{Y}} \ell(\theta; x, y) \cdot q(x, y) \cdot dx \cdot dy \\
&= \int_{\mathcal{X}\times\mathcal{Y}} \ell(\theta; x, y) \cdot \big(p(x, y) + q(x, y) - p(x, y)\big) \cdot dx \cdot dy \\
&= \int_{\mathcal{X}\times\mathcal{Y}} \ell(\theta; x, y) \cdot p(x, y) \cdot dx \cdot dy \\
&\qquad + \int_{\mathcal{X}\times\mathcal{Y}} \ell(\theta; x, y) \cdot \big(q(x, y) - p(x, y)\big) \cdot dx \cdot dy \\
&= \mathbb{E}_P[\ell(\theta; x, y)] + \int_{\mathcal{X}\times\mathcal{Y}} \ell(\theta; x, y) \cdot \big(q(x, y) - p(x, y)\big) \cdot dx \cdot dy \\
&\leq \mathbb{E}_P[\ell(\theta; x, y)] + K_\ell \cdot W(P, Q)
\end{aligned}
$$

Evaluation Bounds for Covariate Shift

- Note that

$$\int_{\mathcal{X}\times\mathcal{Y}} \ell(\theta; x, y) \cdot \big(q(x, y) - p(x, y)\big) \cdot dx \cdot dy$$

$$= \int_{\mathcal{X}\times\mathcal{Y}} \ell(\theta; x, y) \cdot \big(q(\, y \mid x\,)q(x) - p(\, y \mid x\,)p(x)\big) \cdot dx \cdot dy$$

$$= \int_{\mathcal{X}\times\mathcal{Y}} \ell(\theta; x, y) \cdot \big(p(\, y \mid x\,)q(x) - p(\, y \mid x\,)p(x)\big) \cdot dx \cdot dy$$

$$= \int_{\mathcal{X}\times\mathcal{Y}} \ell(\theta; x, y) \cdot p(\, y \mid x\,) \cdot \big(q(x) - p(x)\big) \cdot dx \cdot dy$$

$$= \int_{\mathcal{X}} \left(\int_{\mathcal{Y}} \ell(\theta; x, y) \cdot p(\, y \mid x\,) \cdot dy\right) \cdot \big(q(x) - p(x)\big) \cdot dx$$

$$= \int_{\mathcal{X}} \tilde{\ell}(\theta; x) \cdot \big(q(x) - p(x)\big) \cdot dx$$

$$\leq K_{\tilde{\ell}} \cdot W\big(P(x), Q(x)\big)$$

Evaluation Bounds for Covariate Shift

- Thus, we have

$$\mathbb{E}_{Q}[\ell(\theta; x, y)] \leq \mathbb{E}_{P}[\ell(\theta; x, y)] + K_{\tilde{\ell}} \cdot W\big(P(x), Q(x)\big)$$

Aside: What About Learning?

- Suppose we optimize the upper bound:

$$\mathbb{E}_Q[\ell(\theta; x, y)] \leq \mathbb{E}_P[\ell(\theta; x, y)] + K_{\tilde{\ell}} \cdot W\big(P(x), Q(x)\big)$$

- It is equivalent to optimizing $\mathbb{E}_P[\ell(\theta; x, y)]$, since the penalty is independent of θ

- Need new approaches to use such bounds for learning

Evaluation Bounds

- Need to evaluate the metric $\mathrm{TV}(P, Q)$ or $W(P, Q)$
  - $\mathrm{TV}(P, Q)$ is harder to estimate
  - $W(P, Q)$ can be estimated heuristically

- We focus on covariate shift

Evaluation Bounds

- **Basic idea:** Train a discriminator with bounded Lipschitz constant
  - Construct $X' = \{\,(x,0) \mid (x,y) \in Z\,\} \cup \{\,(x,1) \mid x \in X\,\}$
  - Train $\hat{g}$ on X' **but bound its Lipschitz constant $K_{\hat{g}} \leq 1$**

- Use the Wasserstein distance as the training loss:

$$\hat{g} = \sup_{f:K_f \leq 1} \int_{\mathcal{X}} f(x) \cdot \big(q(x) - p(x)\big) \cdot dx \cdot dy$$

$$= \sup_{f:K_f \leq 1} \left\{ \mathbb{E}_Q[f(x)] - \mathbb{E}_P[f(x)] \right\}$$

$$\approx \sup_{f:K_f \leq 1} \left\{ n^{-1} \sum_{(x,1) \in X'} f(x) - n^{-1} \sum_{(x,0) \in X'} f(x) \right\}$$

Training Lipschitz Neural Networks

- **Simple strategy:** Bound weight matrices individually
  - For example, $g = g_m \circ g_{m-1} \circ \cdots \circ g_1$, then $K_g \leq K_{g_m} \cdot K_{g_{m-1}} \cdot \cdots \cdot K_{g_1}$
- For a single layer
  - If $g_j(x) = W_j x$ is linear, we have $K_{g_j} = \|W_j\|_1$
  - Here, $\|W\|_1$ is the operator norm $\|W\|_1 = \max_x \frac{\|Wx\|_1}{\|x\|_1}$
  - If $g_j(x) = \mathrm{ReLU}(x)$, we have $K_{g_j} = 1$

Training Lipschitz Neural Networks

- Use projected gradient descent

- For $t \in \{1, \ldots, T\}$ (or until convergence):
  - For $j \in \{1, \ldots, m\}$:

$$W_j \leftarrow W_j - \alpha \cdot \nabla_{W_j} L(W_j; Z)$$

$$W_j \leftarrow \frac{W_j}{\|W_j\|_1}$$

Integral Probability Metric Penalties

- **Pros:**
  - Can handle shifts without distribution overlap
- **Cons:**
  - Requires additional assumptions about the true function (e.g., Lipschitz)
  - Cannot be used for learning, only evaluation

Covariate Shift Detection

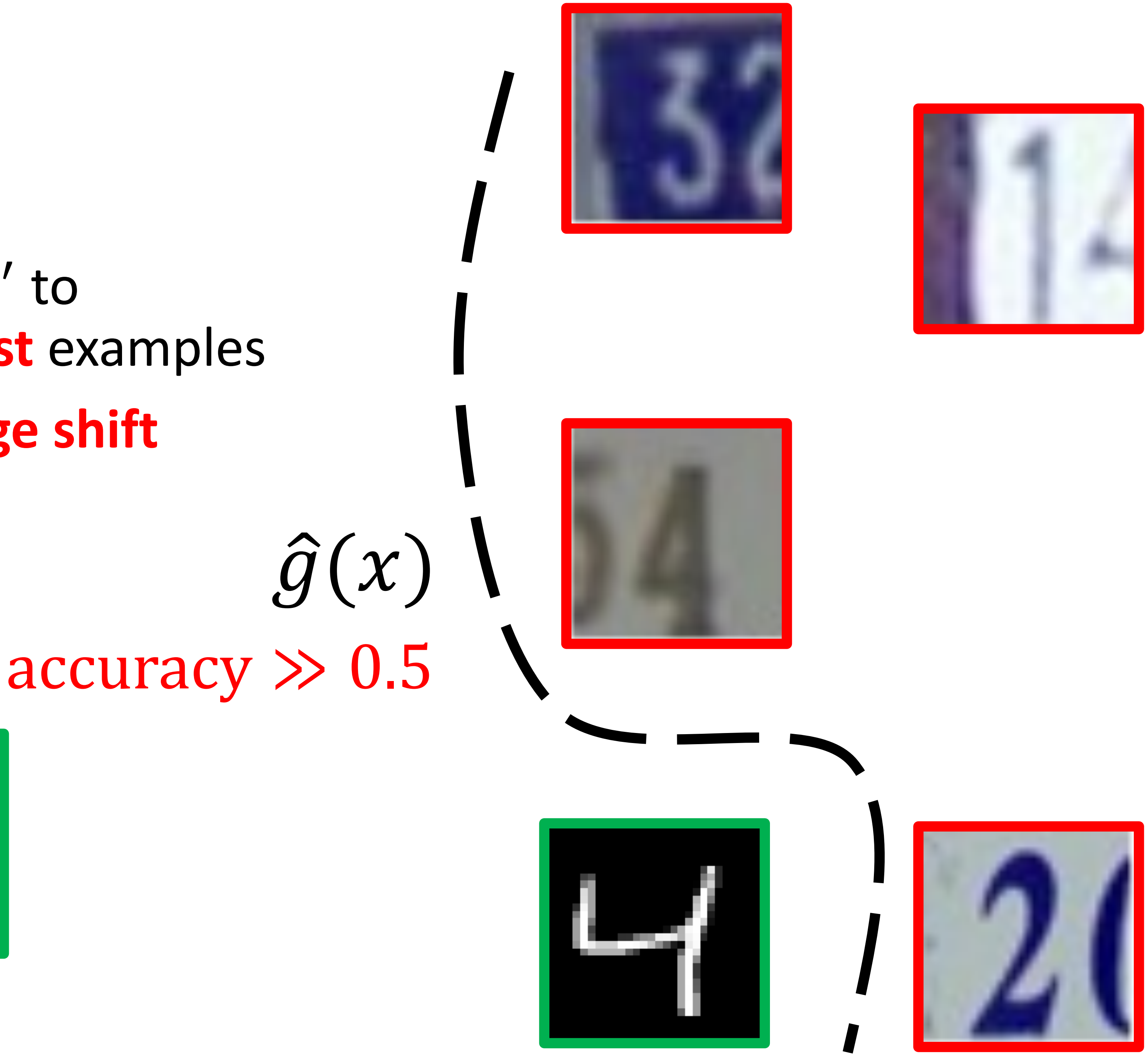
- **Alternative strategy:** Can we test for covariate shift?

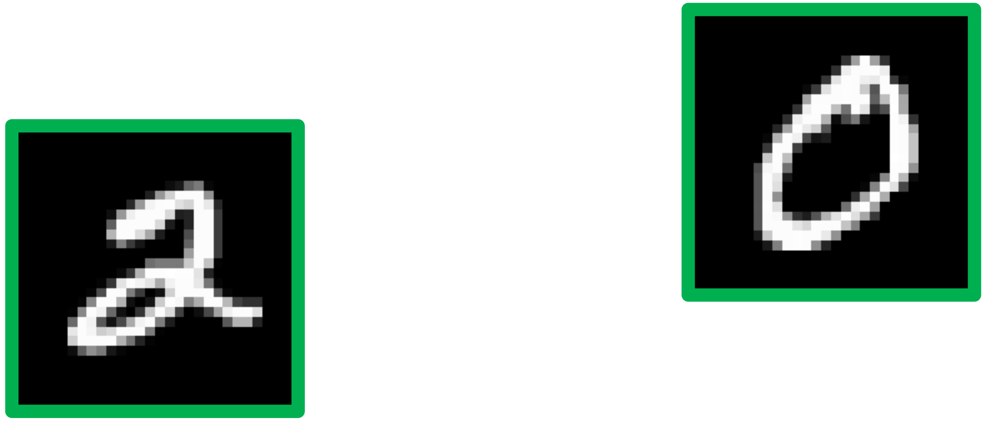
- **Problem setting**
  - **Given:** i.i.d. samples $x_1, \dots, x_n \sim P$ and $x'_1, \dots, x'_n \sim Q$ (denoted X_P and X_Q)
  - **Goal:** Determine whether $P = Q$

- This is a **two-sample test**
  - Lots of work on two-sample tests in the statistics literature
  - **Idea:** Can we leverage our source-target discriminator?
  - Yes! This is called a **classifier test**

Discriminators

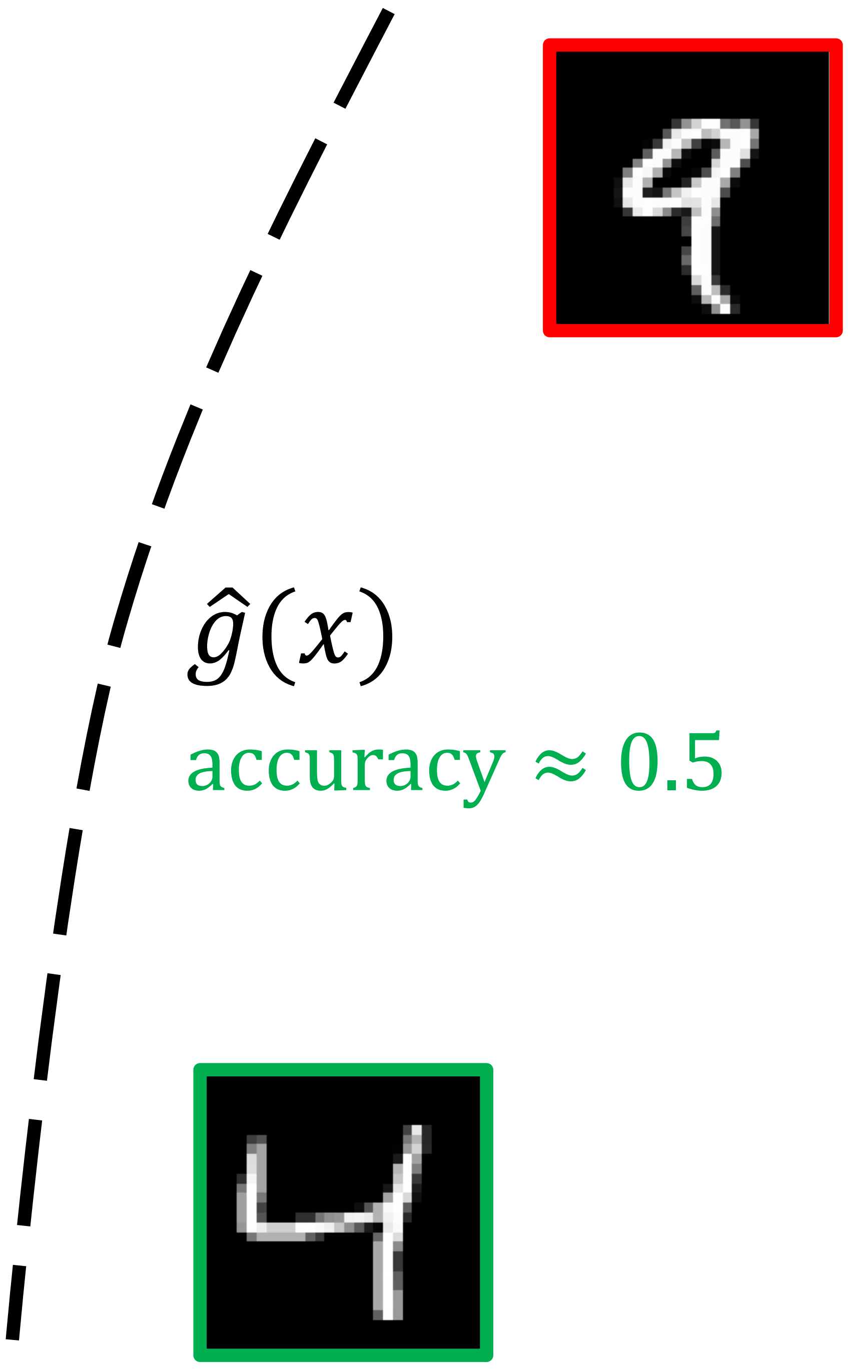
- Train **discriminator** $\hat{g}$ on X' to distinguish **training** and **test** examples
- $\hat{g}$ has **high accuracy** $\Rightarrow$ **large shift**

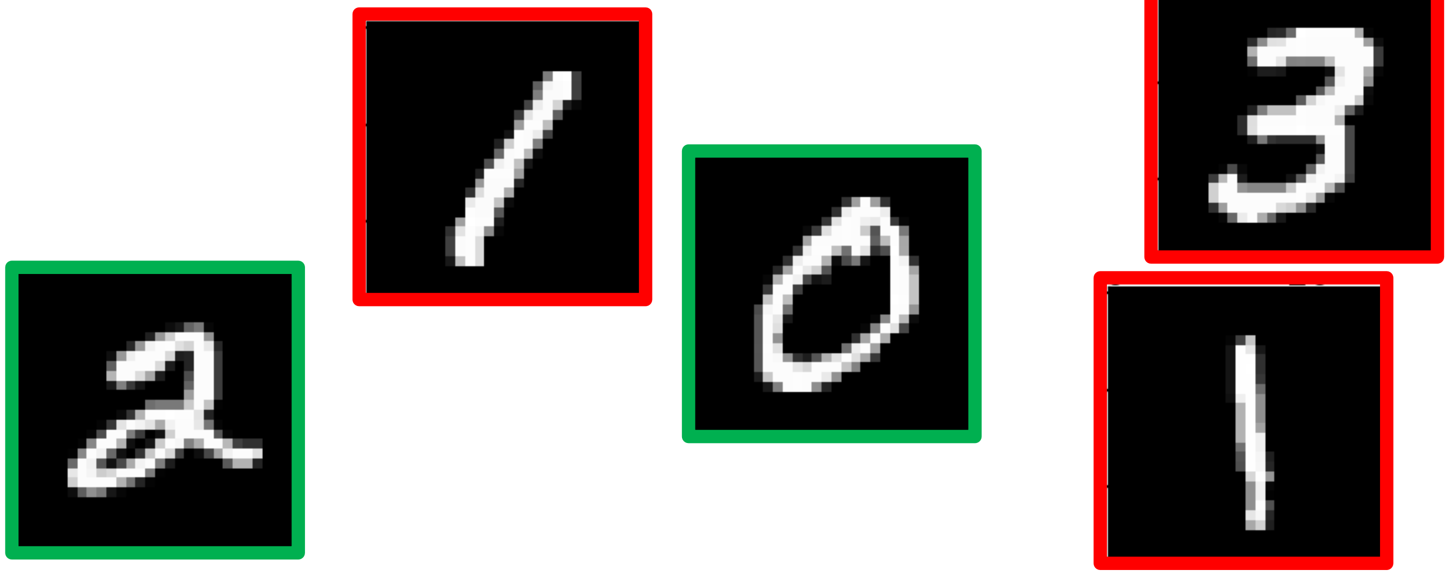




Discriminators

- Train **discriminator** $\hat{g}$ on X' to distinguish **training** and **test** examples
- $\hat{g}$ has **high accuracy** $\Rightarrow$ **large shift**
- $\hat{g}$ has **low accuracy** $\Rightarrow$ **small shift** (assuming sufficient capacity)





Covariate Shift Detection

- **Proposed approach**
  - Train discriminator $\hat{g}$ on $X' = \{ (x, 0) \mid x \in X_P \} \cup \{ (x, 1) \mid x \in X_Q \}$
  - Determine there is covariate shift if $\mathrm{Accuracy}(\hat{g}; X'') \geq \frac{1}{2} + \epsilon$
  - X'' is a held-out test set constructed the same way as X'

- **Question:** How do we choose ϵ?

- **Typical goal:** Choose ϵ so the probability of a false positive is bounded by a user provided error level α:

$$\mathbb{P}_{X''}[\, \mathrm{Detector}(X''; \hat{g}, \epsilon) = 1 \mid P = Q \,] \leq \alpha$$

Covariate Shift Detection

- Note that $\text{Accuracy}(\hat{g}; X) = n^{-1} \sum_{i=1}^{n} 1(\hat{g}(x_i) = b_i)$
- Assuming $P = Q$, then $z_i := 1(\hat{g}(x_i) = b_i)$ is a Bernoulli random variable with mean $\mathbb{E}[1(\hat{g}(x_i) = b_i)] = \mathbb{P}[\hat{g}(x_i) = b_i] = \frac{1}{2}$
- Thus, $\text{Accuracy}(\hat{g}; X) \sim \text{Binomial}\left(n, \frac{1}{2}\right)$, so

$$\mathbb{P}_{X''}[\,\text{Detector}(X''; \hat{g}, \epsilon) = 1 \mid P = Q\,] = \sum_{i=\lceil n\epsilon \rceil}^{n} \text{Binomial}\left(i; n, \frac{1}{2}\right)$$

Covariate Shift Detection

- **Step 1:** Train $\hat{g}$ on $X' = \{\,(x, 0) \mid x \in X_P\,\} \cup \left\{\,(x, 1) \mid x \in X_Q\,\right\}$

- **Step 2:** Compute ϵ so that $\sum_{i=\lceil n\epsilon \rceil}^{n} \mathrm{Binomial}\left(i; n, \frac{1}{2}\right) \leq \alpha$

- **Step 3:** Return "true" if $\mathrm{Accuracy}(\hat{g}; X'') \geq \frac{1}{2} + \epsilon$ else "false"
  - X'' is a held-out test set constructed the same way as X'

Key Takeaway

- We can get provable bounds on the true accuracy of a model $\mathbb{E}[1(\hat{g}(x_i) = b_i)]$ from the test set accuracy $n^{-1} \sum_{i=1}^{n} 1(\hat{g}(x_i) = b_i)$
- Later in the class, we will see how this idea can be used to obtain rigorous uncertainty quantification for machine learning models