

# Introduction to Deep Learning

## 9. Environment and Covariate Shift

STAT 157, Spring 2019, UC Berkeley

Alex Smola and Mu Li

[courses.d2l.ai/berkeley-stat-157](https://courses.d2l.ai/berkeley-stat-157)

# Training ≠ Testing

- **Generalization performance**  
(the empirical distribution lies)
- **Covariate shift**  
(the covariate distribution lies)
- **Logistic regression**  
(tools to fix shift)
- **Covariate shift correction**
- **Label shift**  
(the label distribution lies)
- **Nonstationary Environments**

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

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

$$\log(1 + \exp(-yf(x)))$$

$$\frac{1}{2} (p(x)\delta(1, y) + q(x)\delta(-1, y))$$

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



# Generalization performance

# Generalization performance

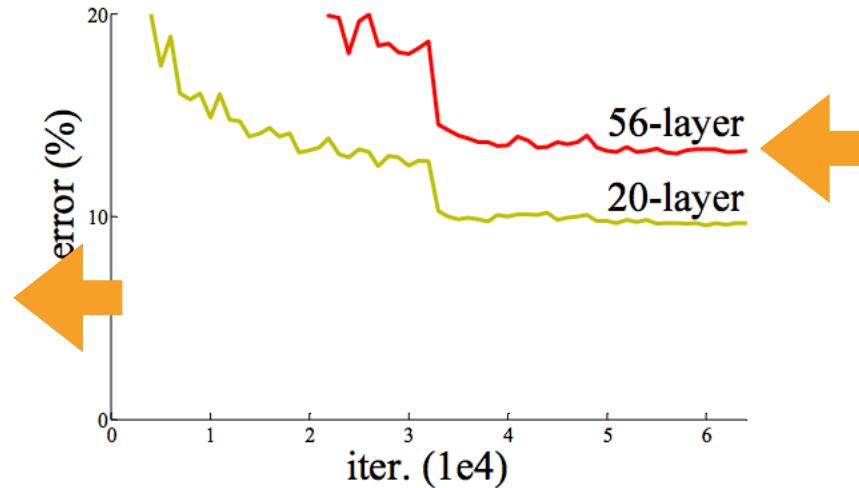
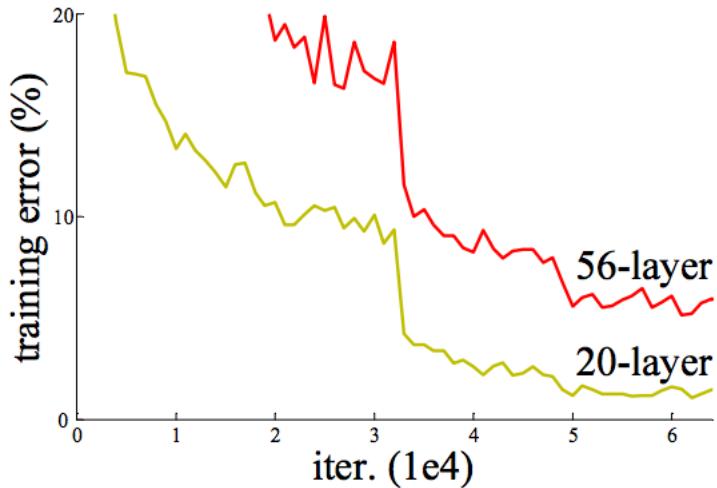


# Generalization performance



# Only cats and dogs?

- Images, too (e.g. He et al., 2015, ResNet paper)



- Alexa  
(‘Please turn off the coffee machine’ vs. ‘coffee machine off’)

# Why?

- Data Distribution  $p(x,y)$
- Dataset drawn from  $p(x,y)$
- Training minimizes empirical risk (plus regularization)

$$\underset{w}{\text{minimize}} \frac{1}{m} \sum_{I=1}^m l(f(x_i, w), y_i)$$

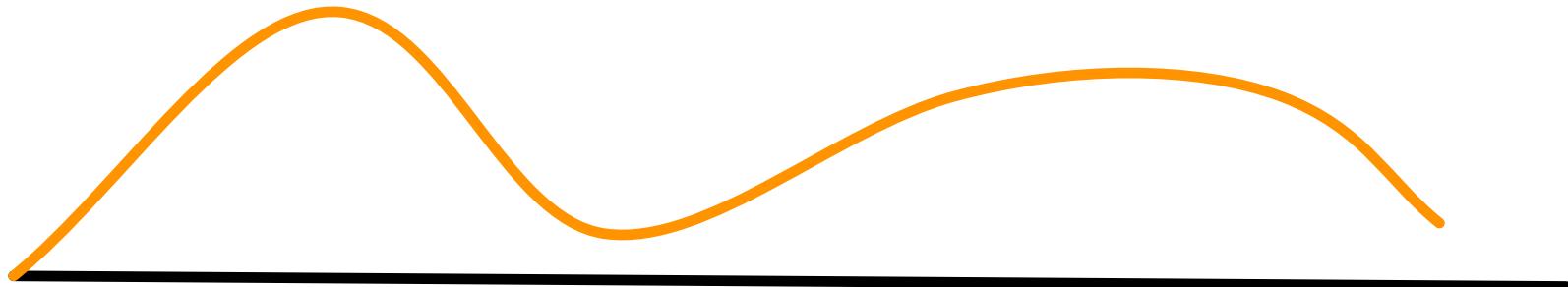
- At test time expected risk matters  
(all the other data we could have seen)

$$\mathbf{E}_{(x,y) \sim p} [l(f(x, w), y)]$$



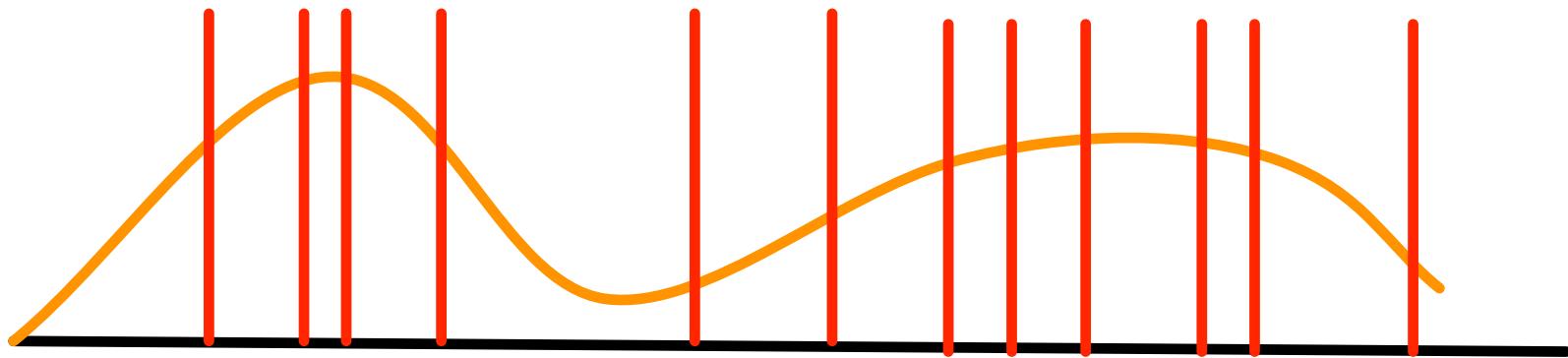
# Why

## Data Distribution



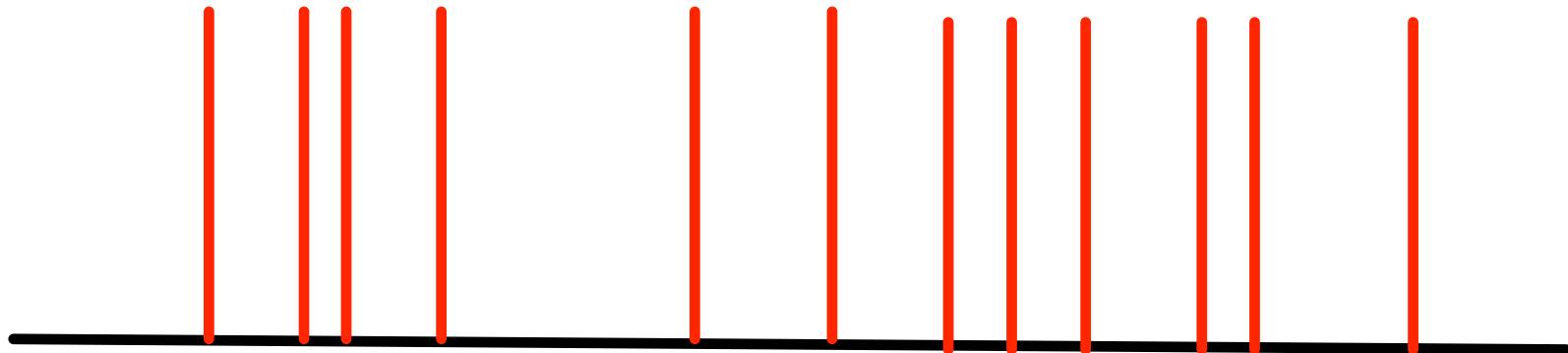
# Why

## Data Distribution with Empirical Sample



# Why

## Empirical Sample



# Fixing it

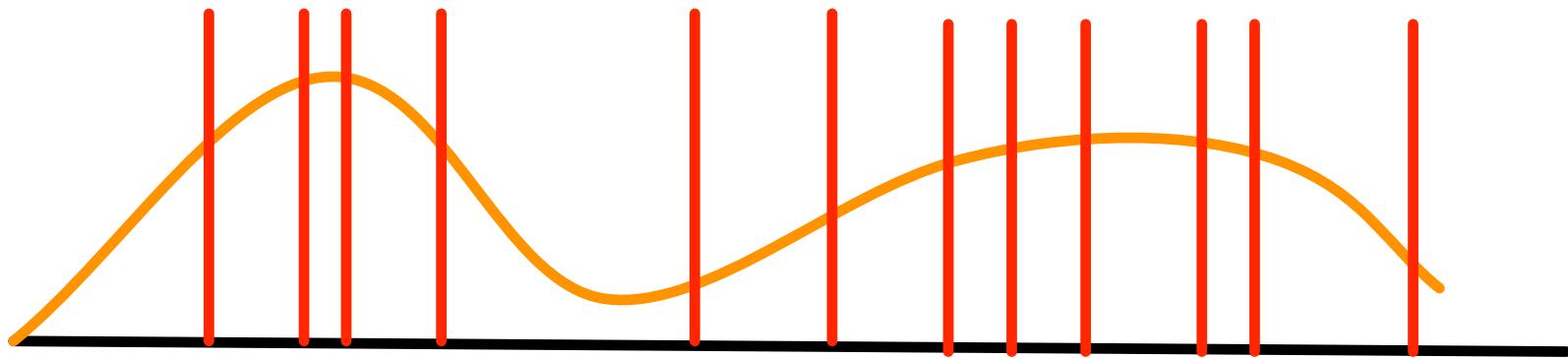
- **Validation set**  
(hold out **separate** data that is not used for training)
- **Chernoff bound**

$$\Pr \left\{ \frac{1}{m} \sum_{I=1}^m l(f(x_i), y_i) - \mathbf{E} [l(f(x), y)] > \epsilon \right\} \leq \exp(-2m\epsilon^2)$$

- **Why does it work?**
  - Validation set was never used for training  
(often violated)
  - Loss bounded within  $[0, 1]$  (otherwise rescale)

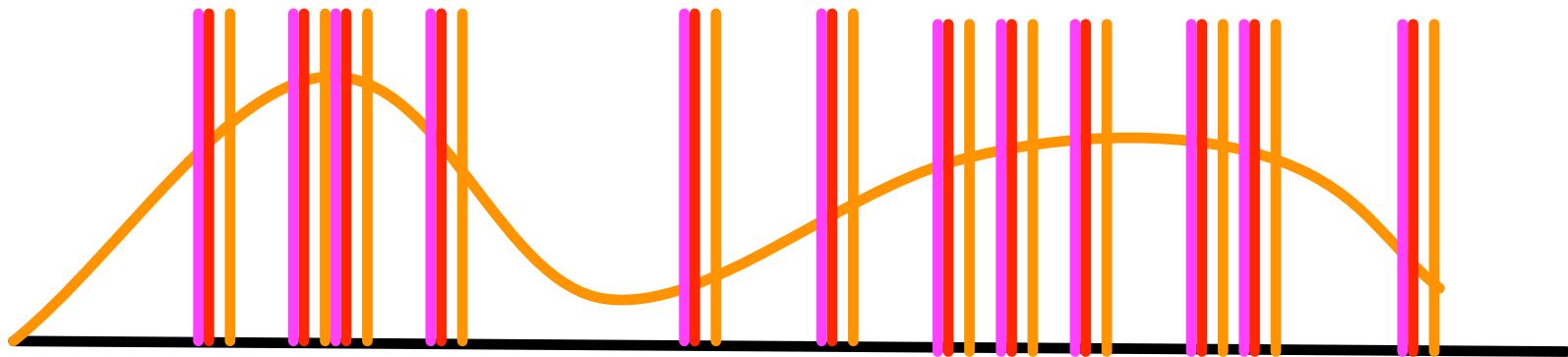
# Fixing it

## Data Distribution with Empirical Sample



# Fixing it

- Input noise (more on this later)
- Dropout (noise within the layers)
- Smoothing the function  $f$  (e.g. weight decay)



return

?

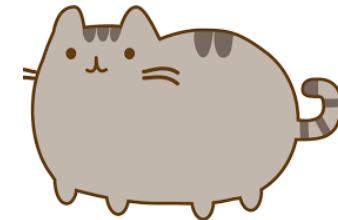
/

covariate shift

# Training set



# At test time



aws

# **Why would anyone do this?**

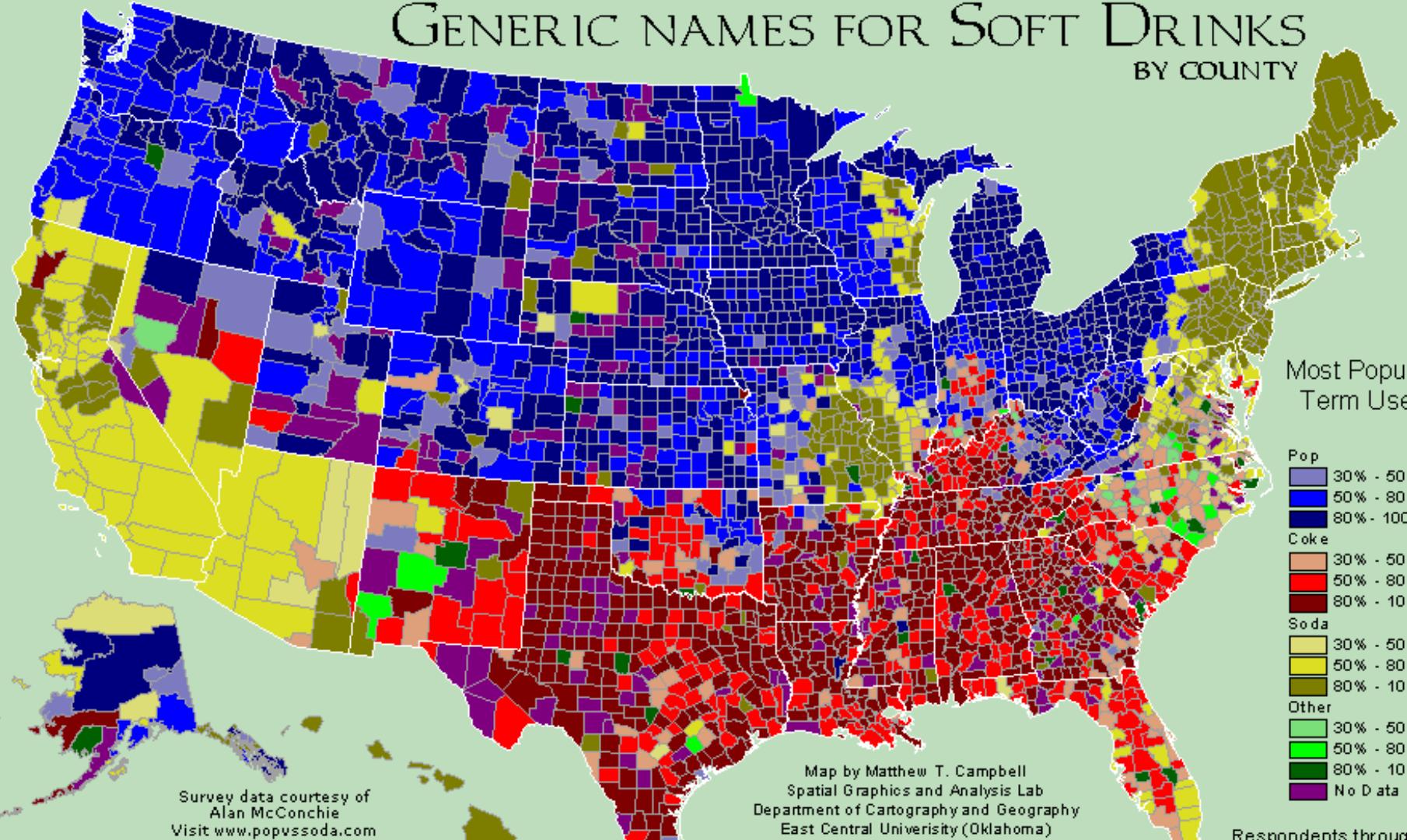


# Covariate Shift

- **Web search**
  - Training - page relevance data for the US market
  - Testing - recommend pages for Canada (UK, Australia)
- **Speech recognition**
  - Training - West coast accent
  - Testing - Southern drawl, Texan, non-native speaker
- **Language**
  - Training - ‘James, bring me a **soda**’
  - Testing - ‘John, bring me a ‘**pop**’ (or coke, etc.)



# GENERIC NAMES FOR SOFT DRINKS BY COUNTY



# Covariate Shift

- **Medical**

- Training - University students + old men with prostate cancer
  - Testing - Potentially sick old men

- **Reinforcement Learning**

- Training - Data gathered with current policy
  - Testing - Environment reacting to updated policy

- **Databases**

- Training - DB tuned to 2017 usage pattern
  - Testing - DB deployed on AWS in 2018



# What is happening? $q(x, y) = q(x)p(y|x)$

- Training Risk

Training data

$$\underset{w}{\text{minimize}} \int dx p(x) \int dy p(y|x) l(f(x, w), y)$$

or rather  $\underset{w}{\text{minimize}} \frac{1}{m} \sum_{I=1}^m l(f(x_i, w), y_i)$

- Test Risk is different

Test data

$$\int dx q(x) \int dy p(y|x) l(f(x, w), y)$$



# Fixing it (covariate shift correction)

- Basic algebra

$$\int dx q(x) f(x) = \int dx p(x) \underbrace{\frac{q(x)}{p(x)}}_{\alpha(x)} f(x) = \int dx p(x) \alpha(x) f(x)$$


- Need to find density ratio, but we don't have either one.
- Estimating p and q directly is really hard and requires specialized tools. Can we recycle classifiers?

# **Fairness and Bias (covariate shift in action & news)**

# Training set



cat

dog



dog



dog

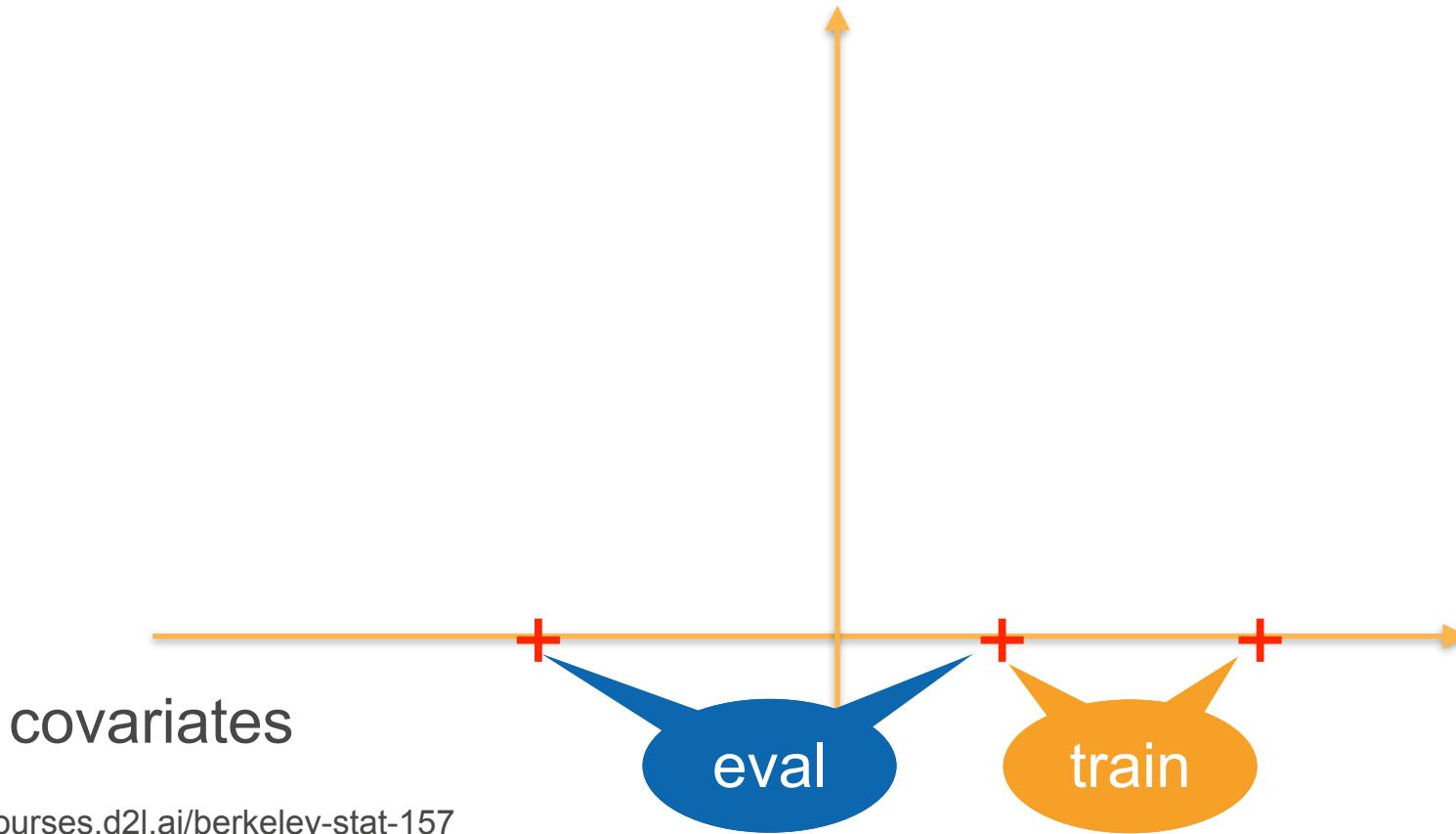
# Test set



**The classifier might perform a lot worse during test time**

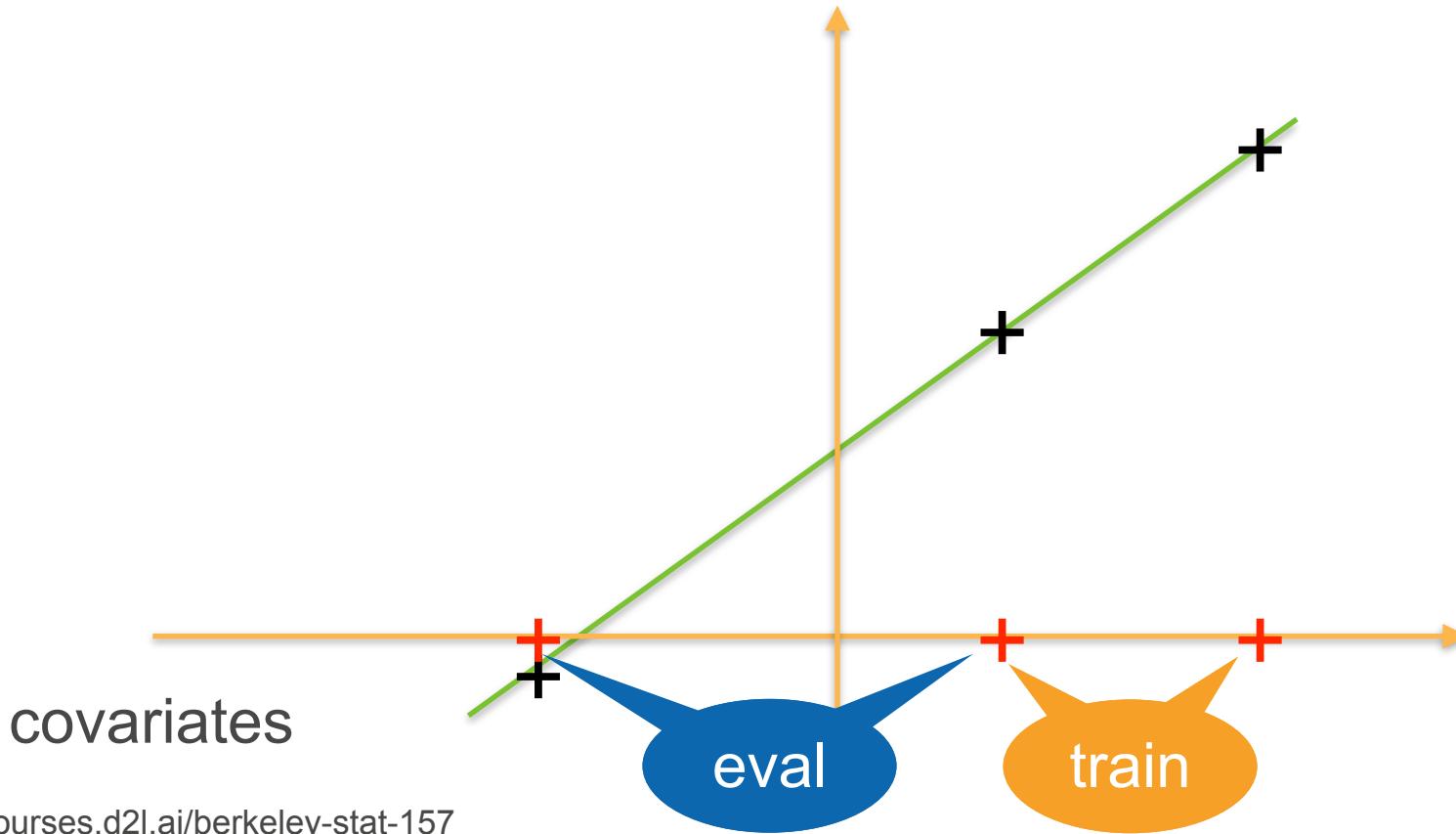
# Why?

# Simple regression problem

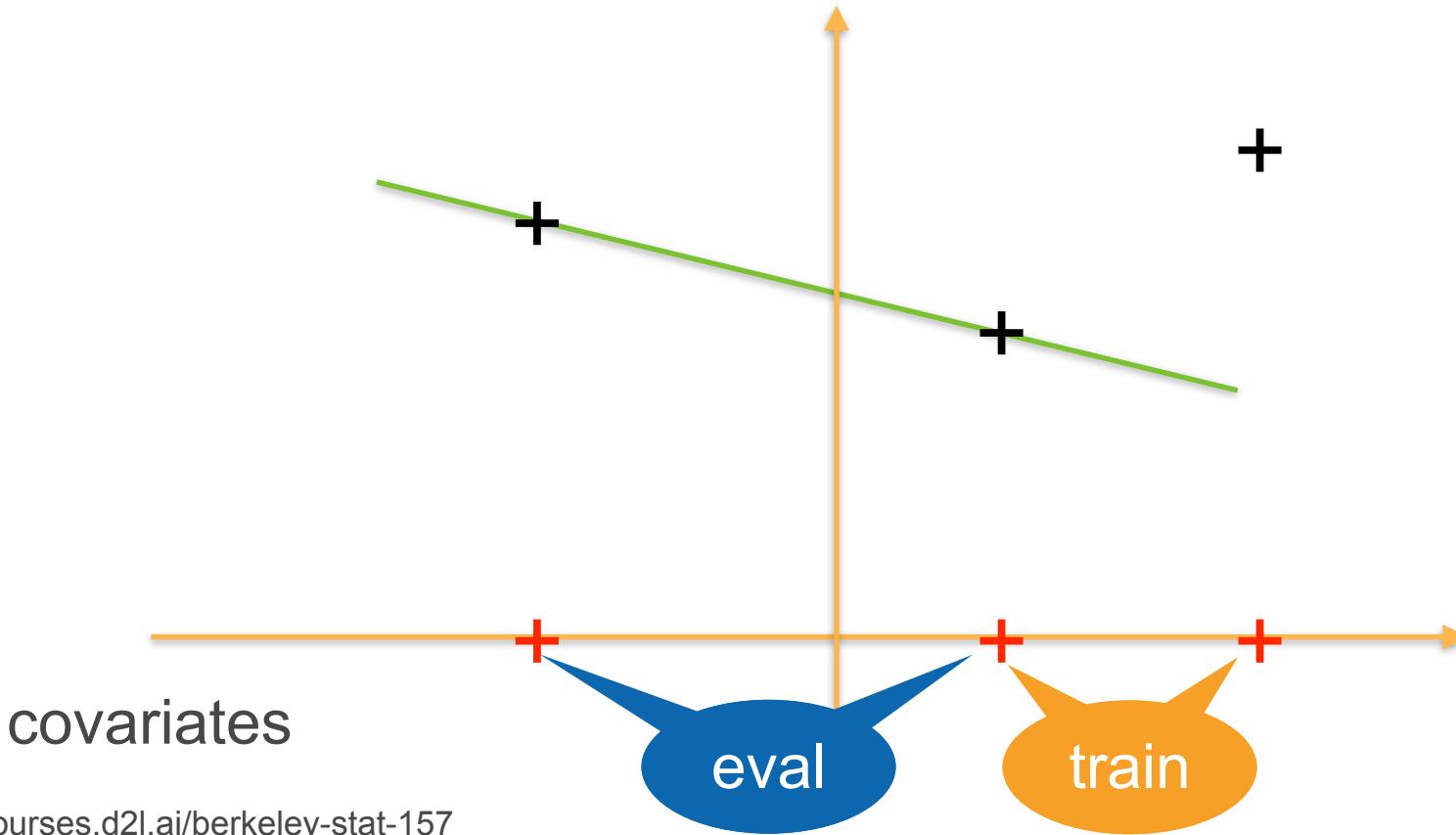


covariates

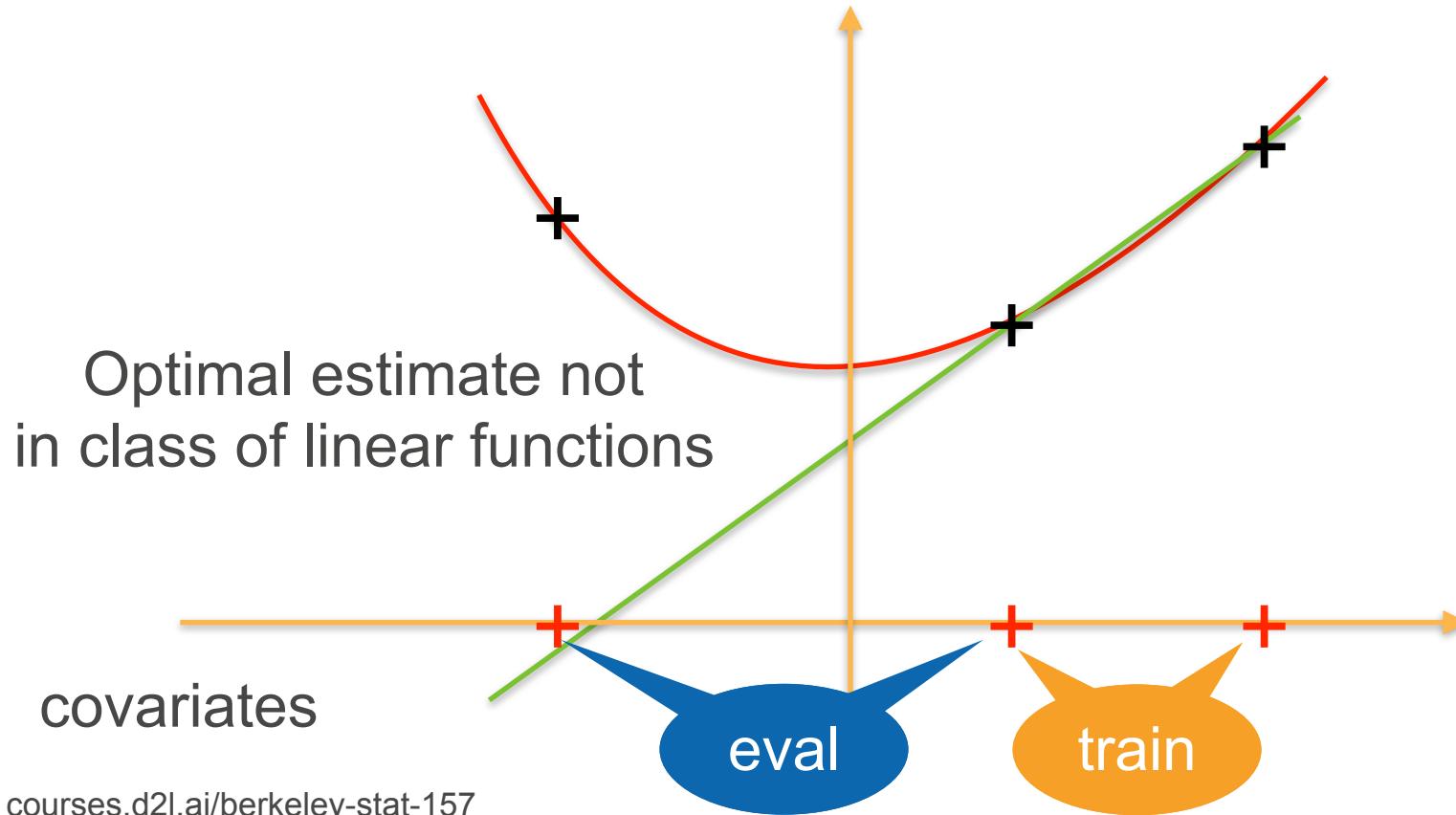
# Simple regression problem



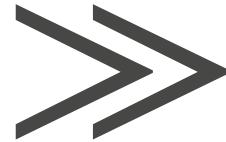
# Simple regression problem



# Simple regression problem



# Training error may be misleading (e.g. faces)



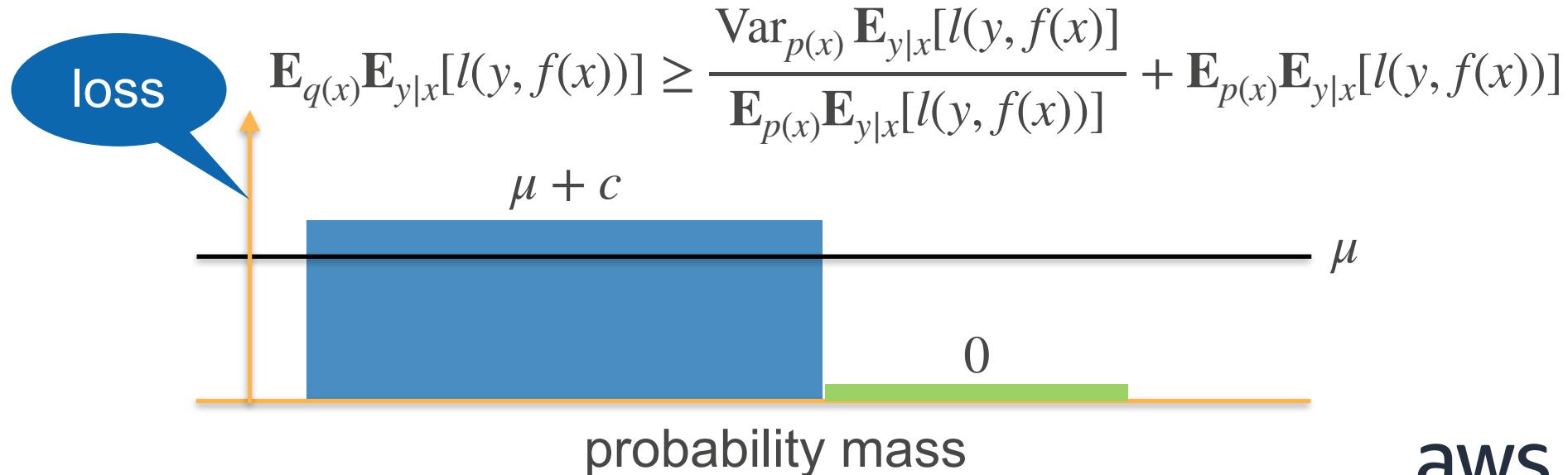
Train on IMDB



Test on weird prof

# No Protection against Bias Theorem

- Estimator performs better (or worse) on some data
- We can always find a distribution  $q$  that is much worse



**TL;DR Testing where we have insufficient amounts of training data *may* yield strange results.**



# Recall - Multiclass Classification

## Calibrated Scale

- Output matches probabilities (nonnegative, sums to 1)

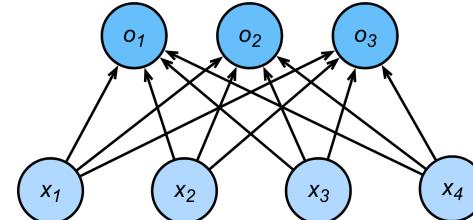
$$\begin{aligned} p(y|o) &= \text{softmax}(o) \\ &= \frac{\exp(o_y)}{\sum_i \exp(o_i)} \end{aligned}$$

- Negative log-likelihood

$$-\log p(y|o) = \log \sum_i \exp(o_i) - o_y$$

## Classification

- Multiple classes, typically multiple outputs
- Score *should* reflect confidence ...



# Two Classes

- Classes 1 and -1

$$p(y = 1 | o) = \text{softmax}(o) = \frac{\exp(o_1)}{\exp(o_{-1}) + \exp(o_1)}$$

- Shift invariance  $o_i \leftarrow o_i + c$

$$p(y = 1 | o) = \frac{\exp(o_1 + c)}{\exp(o_{-1} + c) + \exp(o_1 + c)} = \frac{\exp(o_1)}{\exp(o_{-1}) + \exp(o_1)}$$

# Two Classes

- Choose  $o_{-1} = 0$

$$p(y = 1 | o) = \frac{\exp(o_1)}{\exp(0) + \exp(o_1)} = \frac{1}{1 + \exp(-o_1)}$$

- Negative log-likelihood

$$-\log p(y | o) = \log(1 + \exp(-yo_1))$$

The graph illustrates the behavior of the logistic loss function as  $x \rightarrow \infty$  and  $x \rightarrow -\infty$ . Three curves are shown: a green curve that decreases from 5 at  $x = -5$  towards 0; a blue curve that decreases from 4 at  $x = -5$  towards 0; and an orange curve that increases from -1 at  $x = -5$  towards 0. All three curves approach 0 as  $|x|$  increases.

$$\lim_{x \rightarrow \infty} \log(1 + \exp(-x)) = \log 1 = 0$$
$$\lim_{x \rightarrow -\infty} \log(1 + \exp(-x)) + x = \lim_{x \rightarrow -\infty} \log(1 + \exp(x)) = 0$$

## Logistic loss function

-4      -2      0      2      4



# Logistic Regression Summary

- **Data**  $(x_i, y_i)$  where  $x_i \in \mathcal{X}$  and  $y_i \in \{\pm 1\}$
- **Objective**
$$\underset{w}{\text{minimize}} - \sum_{i=1}^m \log(1 + \exp(-y_i f(x_i, w))) + \text{penalty}(w)$$

- **Conditional Probability Estimate**

$$\log p(y = 1 | o) = \frac{1}{1 + \exp(-o)}$$

return

covariate shift correction

# Covariate shift correction

- Propensity scoring

$$\int dx q(x) f(x) = \int dx p(x) \underbrace{\frac{q(x)}{p(x)}}_{\alpha(x)} f(x) = \int dx p(x) \alpha(x) f(x)$$


- Need to find density ratio, but we don't have either one.
- Key idea: train a classifier between p and q

$$r(x, y) = \frac{1}{2} [p(x)\delta(y, 1) + q(x)\delta(y, -1)]$$

# Covariate shift correction

- Conditional class probability

$$r(y = 1|x) = \frac{p(x)}{p(x) + q(x)} \text{ and hence } \alpha = \frac{q(x)}{p(x)} = \frac{r(y = -1|x)}{r(y = 1|x)}$$

- Logistic regression

$$r(y = 1 | x) = \frac{1}{1 + \exp(-f(x))}$$

$$\implies \alpha(x) = \frac{r(y = -1 | x)}{r(y = 1 | x)} = \exp(f(x))$$



# Covariate Shift Correction Redux

- Training and test data
- Split **as if it were a binary classification problem** (labels -1 and 1 for training and test respectively)
- Train with **logistic regression** to get  $f$
- Use binary classifier output to reweight data
- Solve original problem but weighted

$$\sum_i l(x_i, y_i, g(x_i, w)) \longrightarrow \sum_i \exp(f(x_i)) \cdot l(x_i, y_i, g(x_i, w))$$

# Label shift

# Training set



# Test set



A vertical orange line separates the cat images from the dog images.



# **Why would anyone do this?**



# Label Shift

- **Medical diagnosis**
  - Train on data with few sick patients
  - Test on data during flu season where  $q(\text{flu}) > p(\text{flu})$  while flu symptoms  $p(\text{symptoms}|\text{flu})$  are still the same
- **Speech recognition**
  - Train on newscast data before election
  - Test on newscast after election (new topics, names, discussions, but still same language)

# Label Shift

$$q(x, y) = q(y)p(x|y)$$

- Data generating process  $p(x|y)$  is unchanged
- Labels change since the underlying cause changed
- Need to reweight according to  $\beta(y) = \frac{q(y)}{p(y)}$  to get

$$\int q(y) dy \int p(x|y) dx l(f(x), y) = \int p(y) \frac{q(y)}{p(y)} dy \int p(x|y) dx l(f(x), y)$$



We don't have samples from  $q(y)!$



# Label Shift

$$q(x, y) = q(y)p(x|y)$$

- **Key Idea - measure the estimates on test set**
  - $p(x|y)$  is the same for training and test
  - Distribution of predictions  $|x, y$  has to be the same
- **Simple ‘spectral’ algorithm** (Lipton, Wang, Smola, 2018)
  - Confusion matrix  $C[y'|y] = \Pr(\hat{y}(x) = y'|y)$  on hold out
  - Predicted label vector on test set  $\mu[y'] = \Pr(\hat{y}(x) = y')$
  - Obtain  $q(y)$  via matrix inversion since

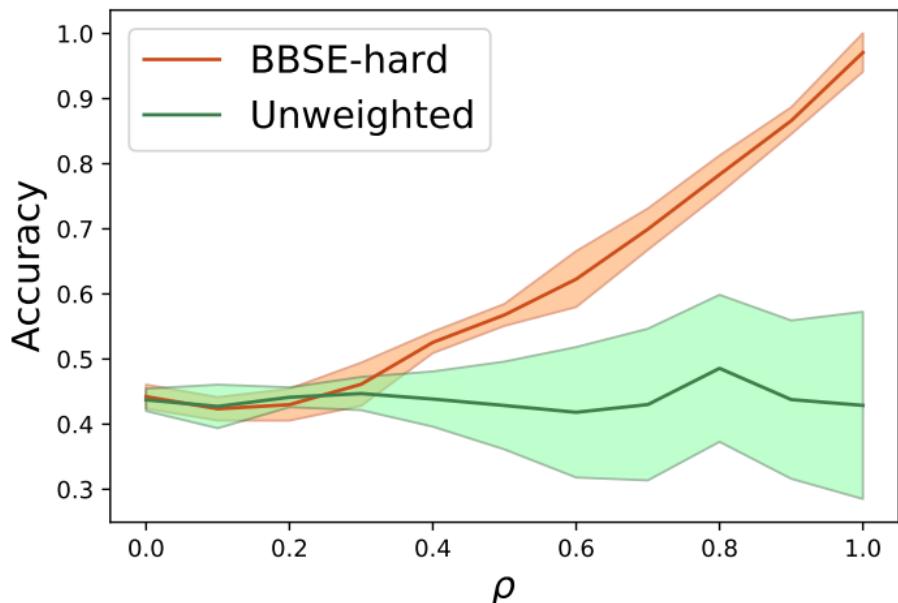
$$\mu[y'] = \sum_y C[y'|y]q(y)$$



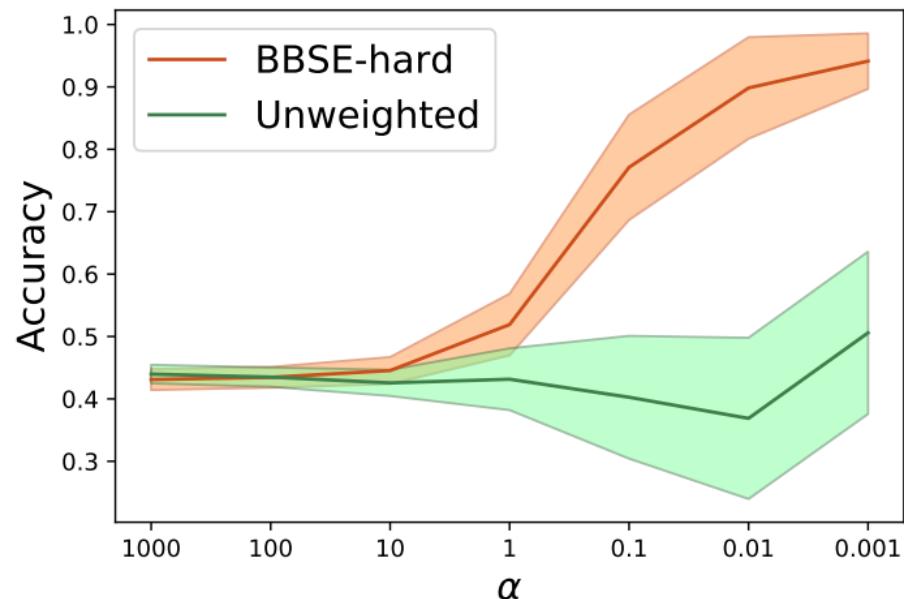
# Guarantees

- **Robust under misspecification**
  - Even if the estimates  $y(x)$  are wrong, calibration is OK:  
(same errors on hold-out and test set)
  - Confusion matrix and label vector are concentrated:  
(use matrix Bernstein inequality)
- **Simple algorithm**
  - Cubic in number of classes, linear in sample size

# Black Box Shift Correction on CIFAR10



Tweaking one class probability

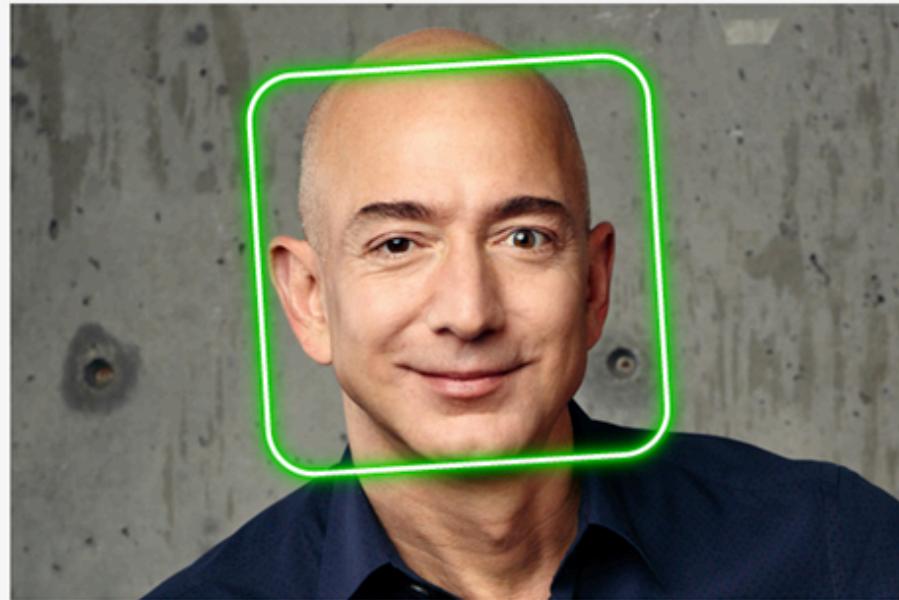


Dirichlet prior over shifts

# Adversarial data

# Celebrity recognition

Rekognition automatically recognizes celebrities in images and provides confidence scores (Your images aren't stored.)



Choose a sample Image



Use your own image

 Upload

or drag and drop

Use image URL

Go

Done with the demo?

[Download SDKs](#)

▼ Results



**Jeff Bezos**  
[Learn More](#)

Match confidence

100%

► Request

► Response

aws

# Adversarial Image Generation (e.g. Sharif et al. 2017)

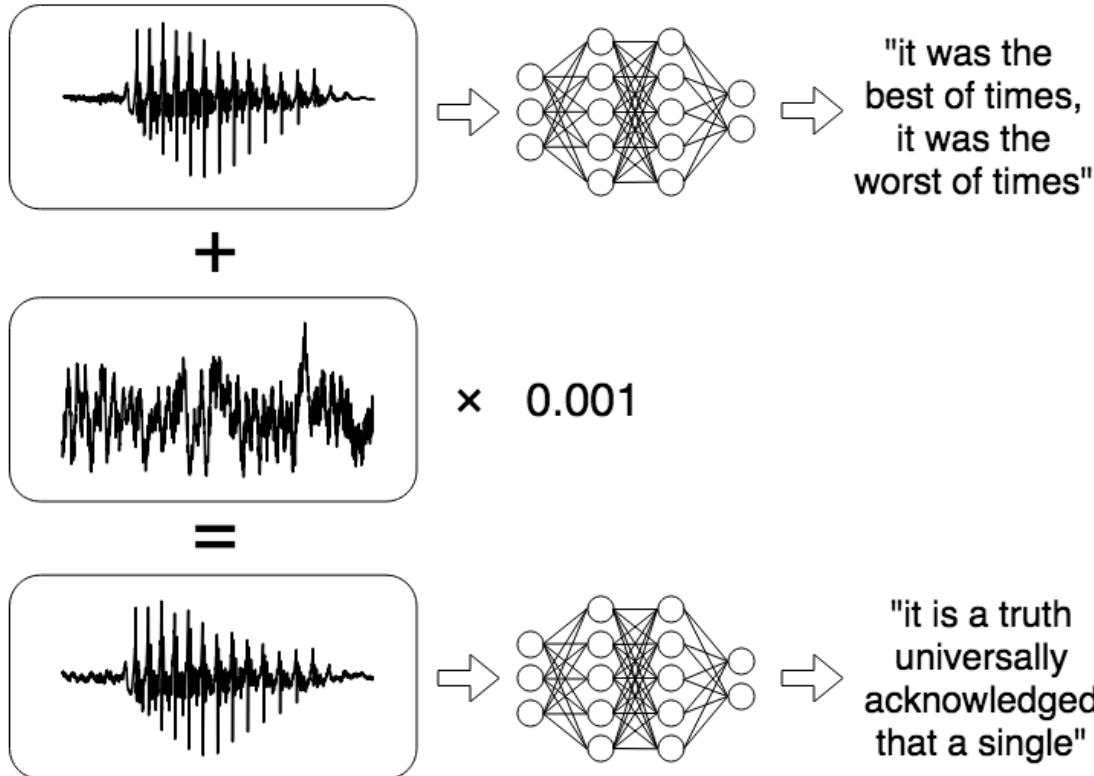


Digital manipulation  
to dodge recognition



In real life - via 3D  
printed glasses

# Adversarial Audio Generation (e.g. Carlini & Wagner, 2018)



- Modify data slightly such as to obtain wrong class

$$\underset{\delta}{\text{maximize}} \ l(f(x + \delta), y)$$

$$\text{subject to } \|\delta\| \leq \epsilon$$

Different norms  
Different datasets  
Different papers ...

# Why does this work?



# 'Unnatural' data



- Training and 'natural' test data live in small subset
- Adversarial data is slightly off that support
- Function behavior undefined away from where data occurs

**Wow. Breathtaking. Is this new?**

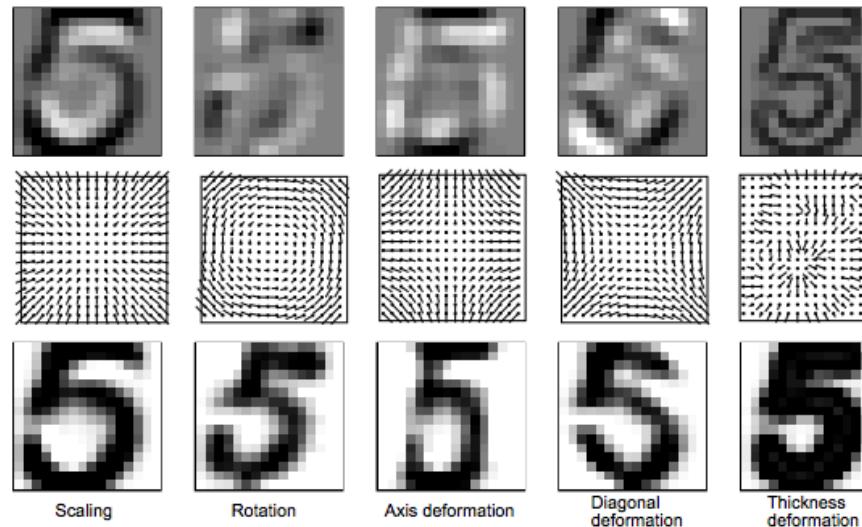


# Spam defenses

- **While TRUE**
  - Mail host extends dataset and trains new classifier
  - Spammer's e-mails are rejected
  - Spammer finds a modification that succeeds
- **Examples**
  - Add highly scoring words (or sentences) to email
  - Add highly scoring sentences (and vary them)
  - Change or forge header ('Dear Alex, ...')

# Invariances

- Tangent Distance (Simard et al., 1995)
  - Invariance transforms don't change the label
  - Explore data and their neighborhood



# Invariances

- **Virtual Support Vectors** (Schoelkopf, 1997)  
Only change the data at the boundary (not enough RAM)
- **Data augmentation for training**
  - **Imagenet** (pretty much every paper)  
Cropping, scaling, change mean, per channel, ...
  - **Speech Recognition**  
Background noise, scenes, ...
  - **Document Analysis**  
Random substrings, word removal, insertion



# Invariant and robust loss

- **Convex loss** (Teo et al, 2005)
  - Family of transformations  $\delta \in \Delta$
  - Penalty for extreme transformations  $1 \geq \eta(\delta) \geq 0$
  - Find the ‘worst’ possible example at each step

Adversarially Robust  
Networks

$$L(x, y, f) = \sup_{\delta \in \Delta} \eta(\delta) l(f(x + \delta), y)$$

e.g. adversarial  
example generator  
Finds worst possible

Reduced penalty for  
extreme distortions

# Nonstationary Environments

# Interaction with Environment

- **Batch** (download a book)  
Observe training data  $(x_1, y_1) \dots (x_l, y_l)$  then deploy
- **Online** (follow the class)  
Observe  $x$ , predict  $f(x)$ , observe  $y$  (stock market, homework)
- **Active learning** (ask questions in class)  
Query  $y$  for  $x$ , improve model, pick new  $x$
- **Bandits** (do well at homework)  
Pick arm, get reward, pick new arm (also with context)
- **Reinforcement Learning** (play chess, drive a car)  
Take action, environment responds, take new action

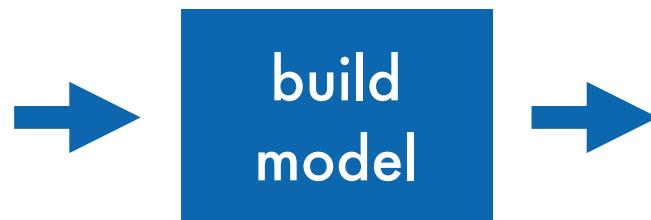
# Batch

training data

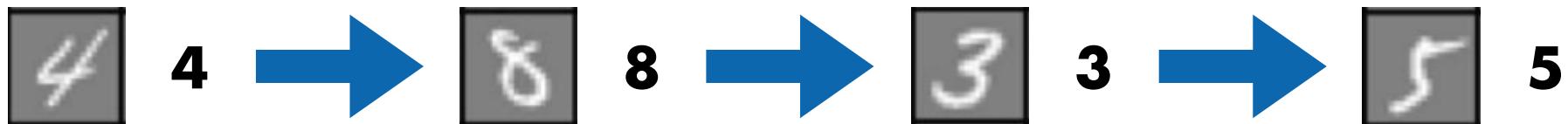
6	5	5	4	1	0
7	4	0	8	4	3
3	4	2	8	1	0
0	0	1	6	5	5
1	1	1	6	7	1
8	6	4	5	3	8
1	7	2	8	4	7
5	2	8	0	4	8
3	3	7	0	5	3
4	8	9	4	0	4

test data

4	9	1	7
6	4	5	6
7	5	9	7
1	1	5	9
4	1	3	1
7	2	9	1
6	8	9	3
3	7	+	6
1	1	0	3
5	0	5	0



# Online



System improves as we see more data

# Bandits

- Choose an arm (action)
- See what happens (get reward)
- Update model
- Choose next arm (action)

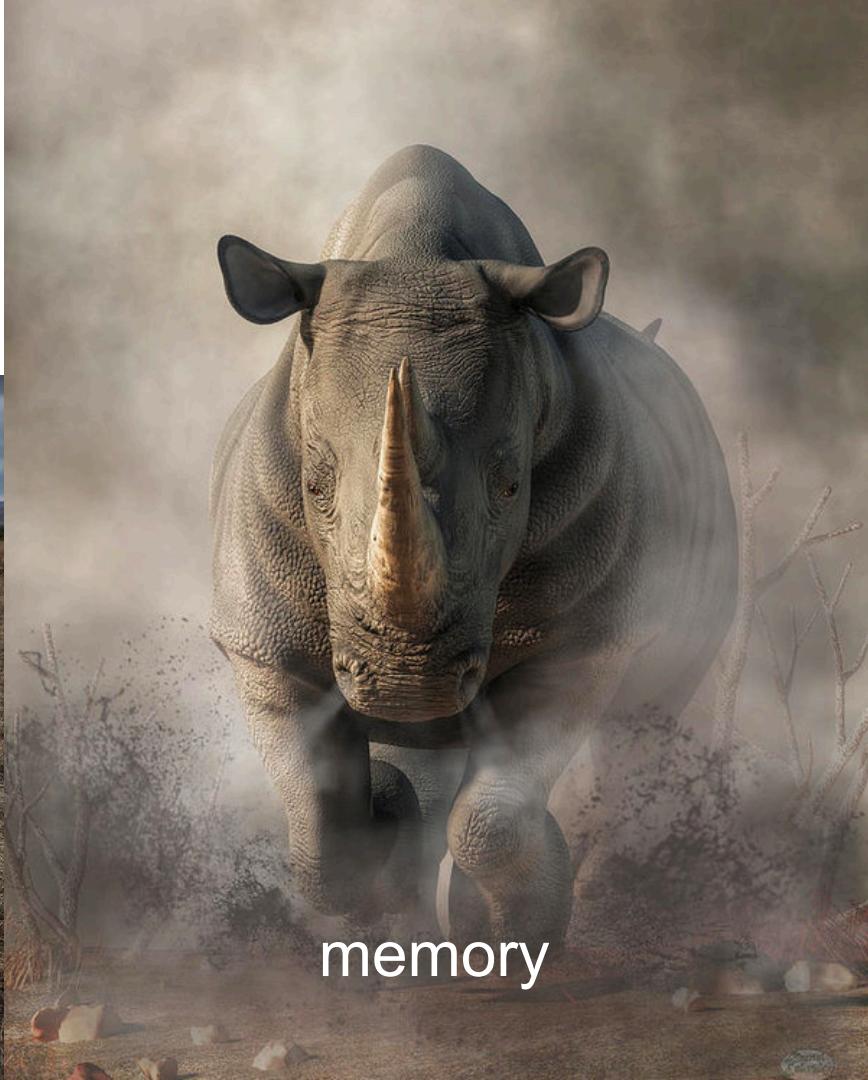
The bandit **doesn't remember** what you did last summer.



# Stateful Systems



no memory



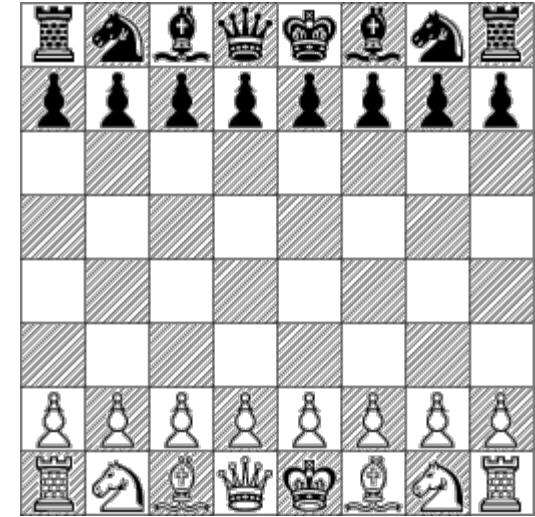
memory

# Reinforcement Learning & Control

- Take action
- Environment reacts
- Observe stuff
- Update model

Repeat

- environment (cooperative, adversary, doesn't care)
- memory (goldfish, elephant)
- state space (tic tac toe, chess, car)
- past observations (server log, generated during training)



# Reinforcement Learning & Control

- **Games**

- Chess, Go, Backgammon (fully observed)
- Poker, Starcraft, ATARI (partially observed, random)

- **Parallelism**

- Computation advertising, recommender systems (multiple agents & independent parallel games)
- Load balancing & scheduling (multiple agents)

- **Actions**

- Continuous decisions (driving, flying, robots in general, HVAC)
- Discrete (elevator, work allocation)

- **Simulations**

- MuJoCo style
- Only reality (server center)

# Training ≠ Testing

- **Generalization performance**  
(the empirical distribution lies)
- **Covariate shift**  
(the covariate distribution lies)
- **Logistic regression**  
(tools to fix shift)
- **Covariate shift correction**
- **Label shift**  
(the label distribution lies)
- **Nonstationary Environments**

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

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

$$\log(1 + \exp(-yf(x)))$$

$$\frac{1}{2} (p(x)\delta(1, y) + q(x)\delta(-1, y))$$

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

