DSCI 565: LINEAR NEURAL NETWORKS FOR CLASSIFICATION

Ke-Thia Yao

Lecture 5: 2025 September 10

More on Generalization

Error Measurements and the Test Set

- Given a dataset $\mathcal{D}$, empirical error $\epsilon_{\mathcal{D}}$ computes the fraction of wrong classification for model f:

indicator function

$$\epsilon_{\mathcal{D}}(f) = \frac{1}{n} \sum_{i=1}^{n} 1(f(\mathbf{x}^{(i)}) \neq y^{(i)})$$

- Population error ϵ is the expected error fraction with some underlying population distribution $P(X, Y)$ with probability density function $p(\boldsymbol{x}, y)$:

$$\epsilon(f) = E_{(\mathbf{x},y)\sim P} 1(f(\mathbf{x}) \neq y) = \int\int 1(f(\mathbf{x}) \neq y) p(\mathbf{x}, y)\, d\mathbf{x} dy$$

- We care about population error. Can approximate using empirical error, if
  - $\mathcal{D}$ is not used during training, i.e., $\mathcal{D}$ is the **test set**
  - $\mathcal{D}$ is sampled from the population distribution

Central Limit Theorem

- Central limit theorem
  - Given any distribution with mean μ and standard deviation σ, and n random samples from this distribution
  - As n approaches infinity the sample mean $\hat{\mu}$ approaches μ with sample standard deviation $\sigma/\sqrt{n}$
- Treat each instance in the test set as a sample
- With large enough n the sample mean approaches the population mean
- If we want to reduce the sample standard deviation by $1/2$, then we must increase n by a factor of 4

Test Set Size

- Probabilistically test error (0 or 1) of a random sample is just a Bernoulli random variable
- Variance of a Bernoulli random variable is $p(1-p)$ with maximum of 0.25, when $p = 0.5$
- Setting aside the fact central limit theorem requires n to approach ∞
  - If we want to be 68% confident (one std dev) sample mean is within ± 0.01 of population mean, then need 2500 samples $\left(\sqrt{0.25/2500} = 0.01\right)$
  - If we want to be 95% confident (two std dev), then need 10,000 samples $\left(\sqrt{0.25/10000} = 0.005\right)$

Test Set Reuse

- In the strictest sense, a test set should only be used **once**
- Adaptive overfitting
  - Suppose you develop what you thought was the best model f_1, but it did not perform well on the test set
  - You go back develop f_2
  - Then f_3, and so on
  - In a sense you have become part of machine learning algorithm, and you have seen the test set

Statistical Learning Theory

- Instead of a test set, can we derive theorical bounds on the generalization gap between empirical error and population error?
- Yes, *statistical learning theory* has developed such bounds
- These bound formulas on based on
  - The complexity (flexibility) of the model language, i.e., the VC dimension
  - Approximate correctness of the model α, i.e., the generalization gap
  - The probability δ of not finding an approximate correct model
  - The number of training examples n
- For example, we may want $\alpha = 0.1$ and $\delta = 0.2$, what should be the size training example n?

Bounding the Generalization Gap

$$\underbrace{P\left(\overbrace{R[p,f] - R_{\mathrm{emp}}[\mathbf{X},\mathbf{Y},f] < \alpha}^{\text{Approximately Correct Model}}\right) \geq 1-\delta}_{\text{Probably Approximately Correct Model}} \;\; \text{for} \;\; \alpha \geq c\sqrt{(\mathrm{VC} - \log\delta)/n}$$

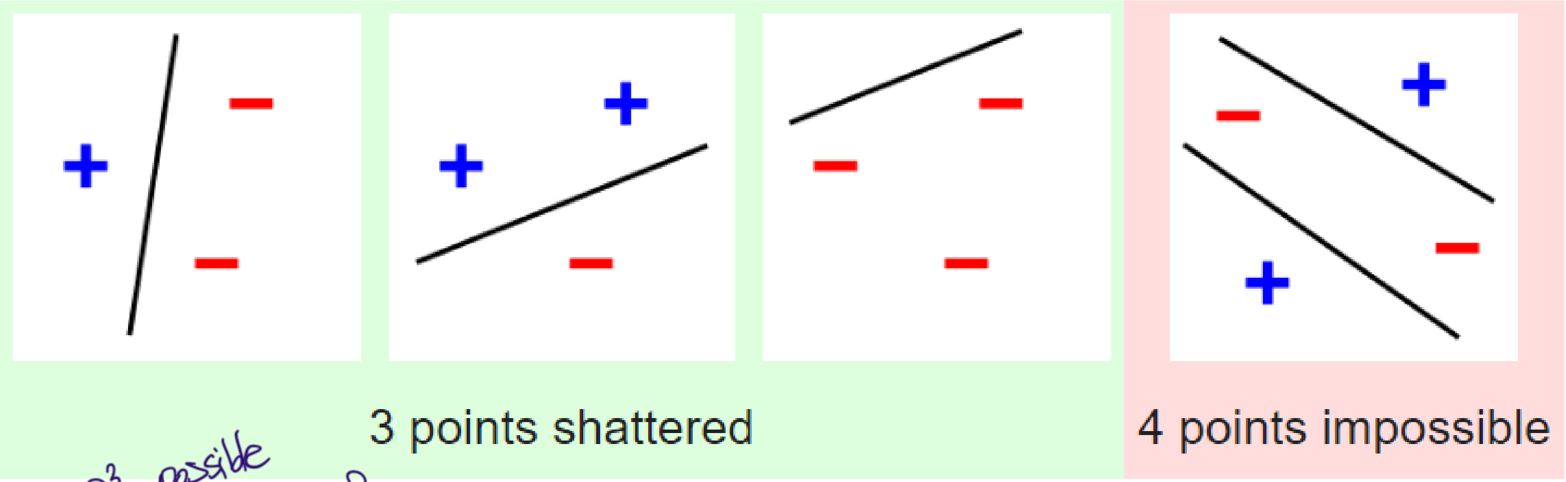
- For a given modeling language (VC) and lost function constant (c), to satisfy the inequality, we can
  - Get more training example, $\mathcal{O}(1/\sqrt{n})$
  - Allow less correct models, increase α
  - Lower probability of finding approximately correct model, increase δ

Vapnik Chervonenkis Dimension

- Vapnik-Chervonenkis (VC) dimension measures the representational power of the modeling language $\mathcal{H}$
- VC dimension of $\mathcal{H}$ is the number of points $\mathcal{H}$ can **shatter**
- Shattering n points
  - A set on n points can be labelled 2^n ways using binary labels $+, -$
  - If there exists a model $h \in \mathcal{H}$ for every possible 2^n labelling of **a set** of n points, then the VC dimension of $\mathcal{H}$ is at least n
- *Note: does not have to shatter all possible sets of n points, just have to shatter as least one set of n points*

VC Dimension of Linear Models

- For 2-dimensional inputs, VC dimension is 3



shatter 2^3 possible
linear model at least 3

https://en.wikipedia.org/wiki/Vapnik%E2%80%93Chervonenkis_dimension

- For d-dimensional inputs, VC dimension is d+1

for linear model

VC Dimension of Neural Networks

- Given a neural network with sigmoid activation function of $||V||$ nodes and $||E||$ edges
  - VC dimension is at least $\Omega(||E||^2)$
  - VC dimension is at most $O(||E||^2||V||^2)$

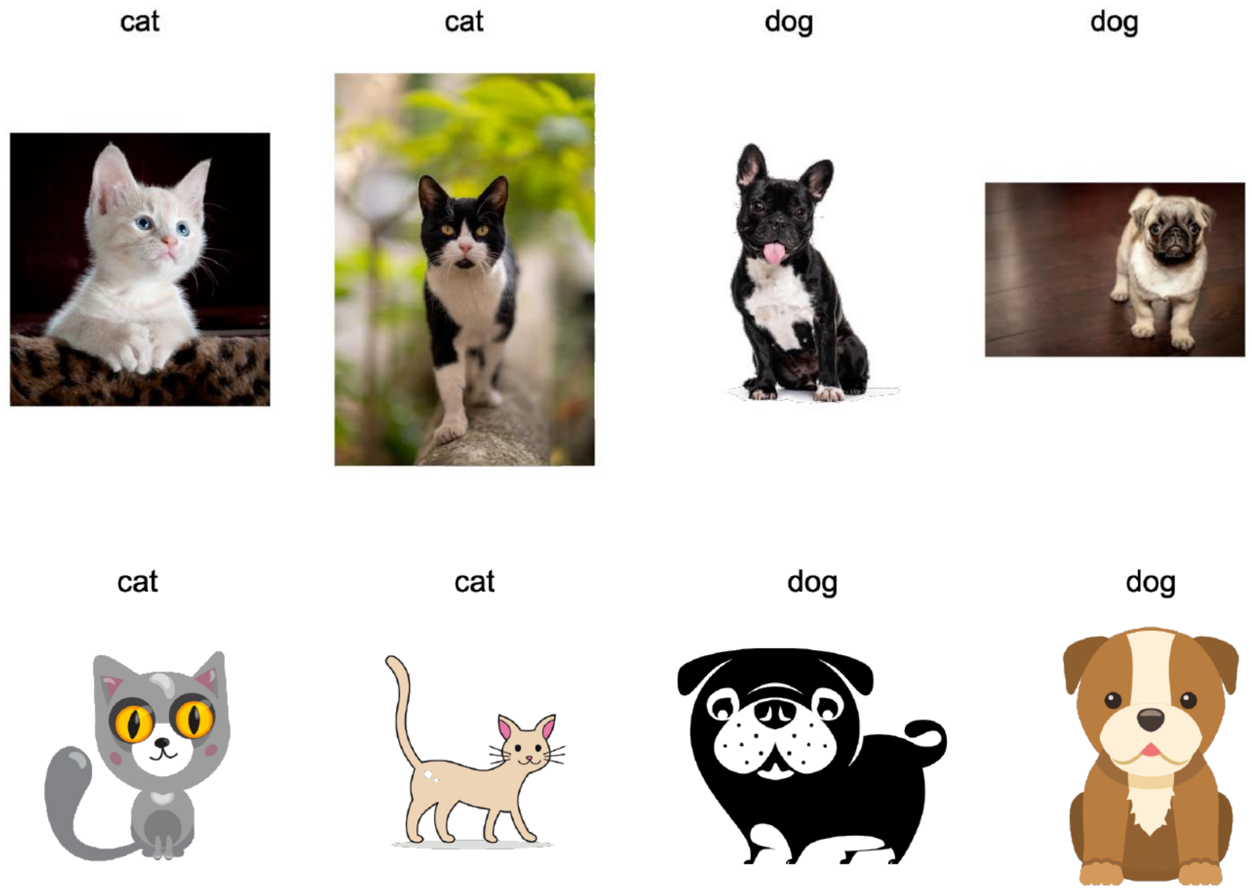
Deployment Environment

- Previous analysis assumes the training set is sampled from the population distribution
- But, if the actual environment in which we deploy the model may be different then the assumed population distribution
- Or perhaps the environment changes over time (patient population ages)
- Or the environment adapts to our model (spam detection)
- These are all causes of ***distribution shift***

Types of Distribution Shift: Covariate Shift

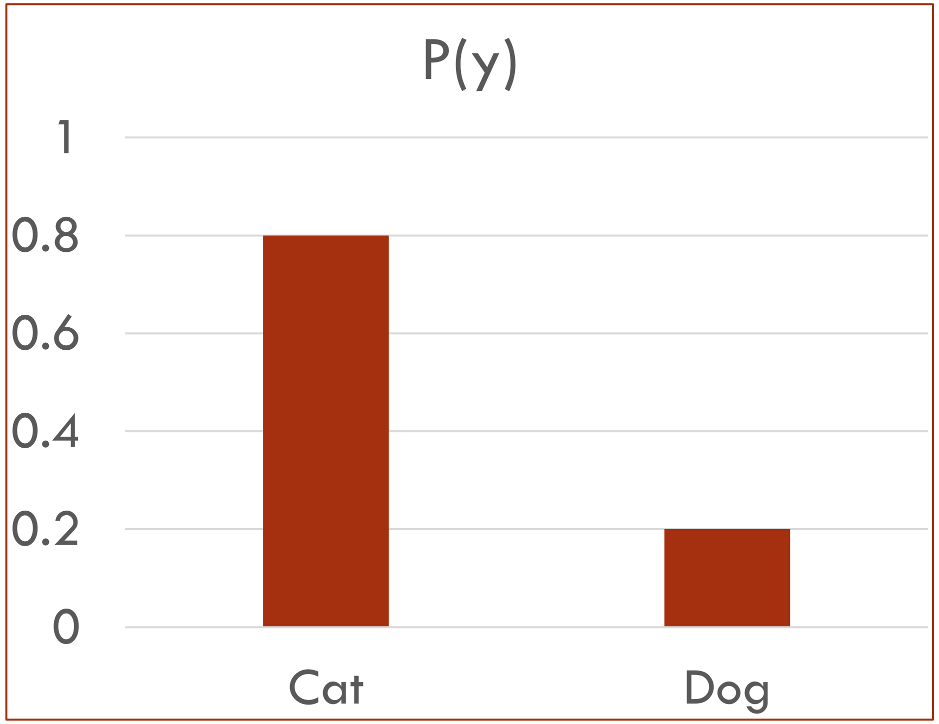
- Assume our training data is sampled from $P_S(\boldsymbol{x}, y)$, but the test data is sampled from $P_T(\boldsymbol{x}, y)$

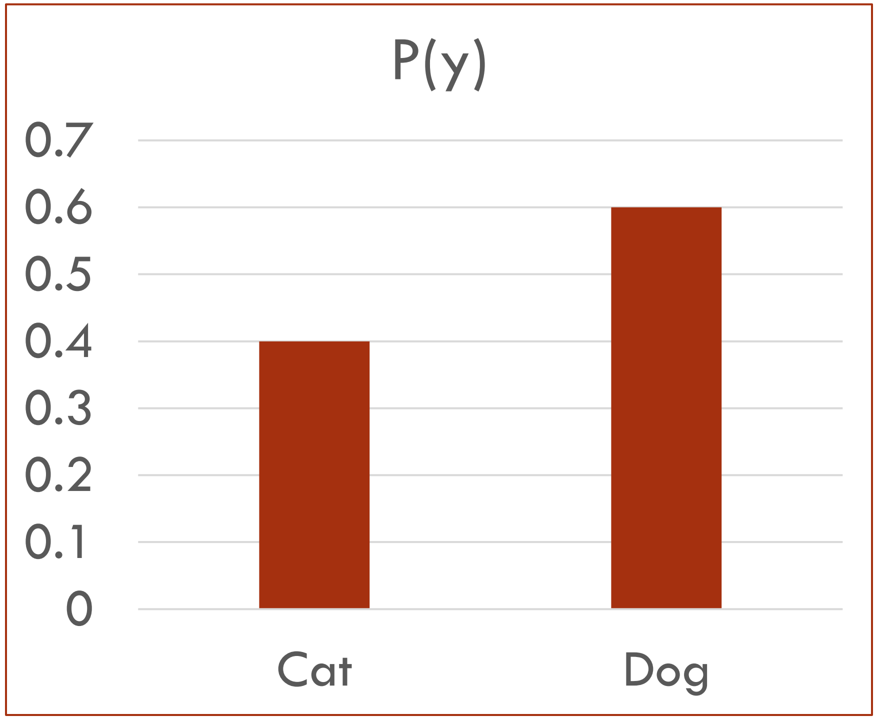
- Covariate Shift assumes $P(\boldsymbol{x})$ changes, but $P(y|\boldsymbol{x})$ does not change



Types of Distribution Shift: Label Shift

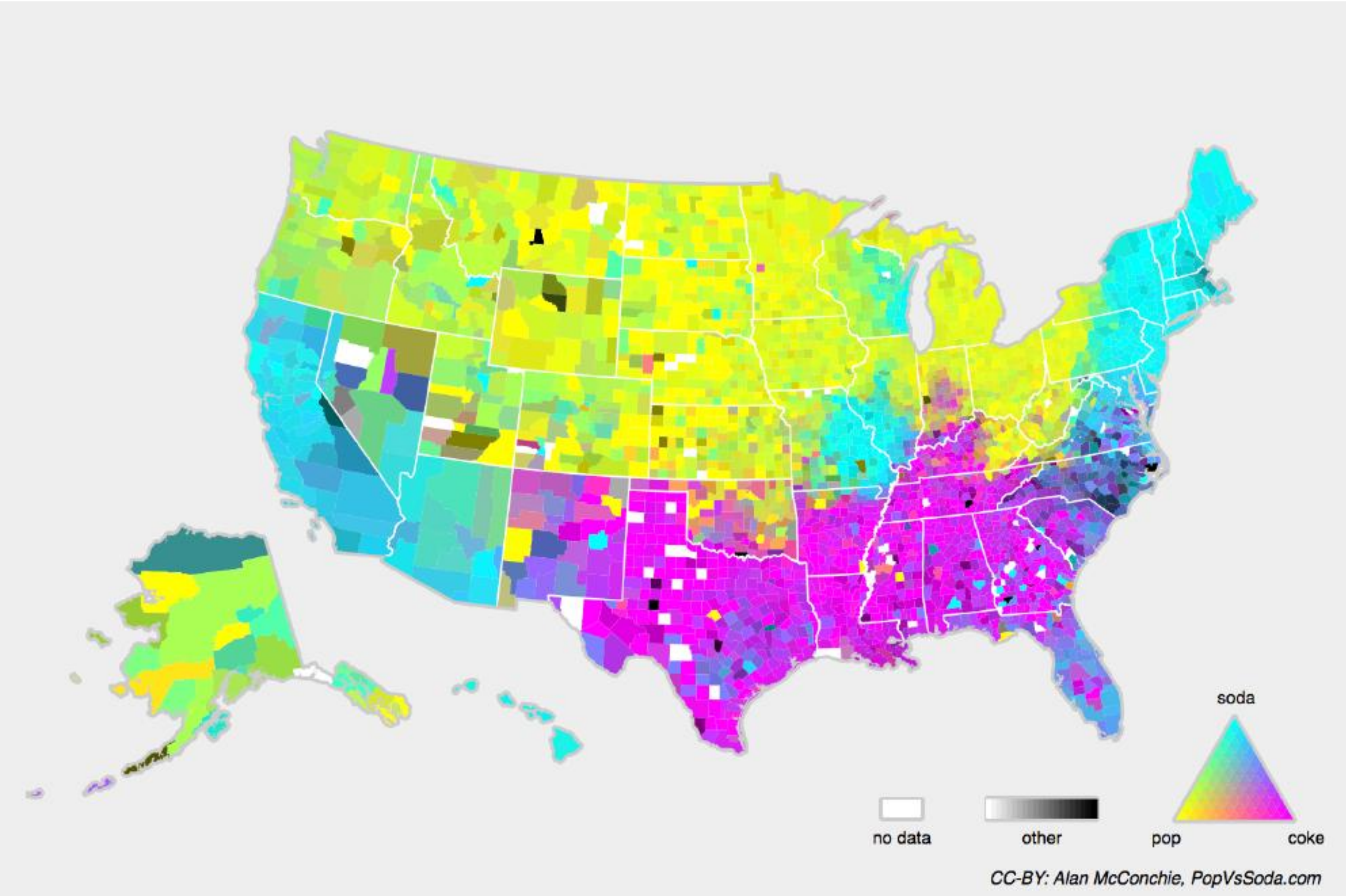
- Label shift assumes $P(y)$ changes, but $P(\boldsymbol{x}|y)$ does not change
- Label shift is sometimes called prior probability shift





Types of Distribution Shift: Concept Shift

- The definition of the label changes, i.e., $P(y|\boldsymbol{x})$
- For example, label for soft drink differs across US



Examples of Distribution Shift

- Medical diagnostics: cancer detector works wonderfully on train/test set, but fails miserably on deployment
  - Oncologist did not provide enough negative instances. Additionally negatives instances gathered from university student volunteers.
- Object detection: tank detector failed to tanks in in forest
  - Images without tanks were taken in morning, and images with tanks were taken later in the day
- Recommendation system: continuous to recommend Santa hats long after Christmas
- …

Covariate Shift Correction

- The loss over the true population $p(\boldsymbol{x}, y)$ is

$$E_{p(\mathbf{x},y)}[l(f(\mathbf{x}), y)] = \int\int l(f(\mathbf{x}), y)p(\mathbf{x}, y)\, d\mathbf{x}dy.$$

- But training distribution is drawn from $q(\boldsymbol{x}, y)$ with $p(y|\boldsymbol{x}) = q(y|\boldsymbol{x})$
- We write population loss as

$$\int\int l(f(\mathbf{x}), y)p(y \mid \mathbf{x})p(\mathbf{x})\, d\mathbf{x}dy = \int\int l(f(\mathbf{x}), y)\underbrace{q(y \mid \mathbf{x})q(\mathbf{x})}^{q(x,y)}\boxed{\frac{p(\mathbf{x})}{q(\mathbf{x})}}\, d\mathbf{x}dy$$

Adjust weight of each instance

Covariate Shift Correction

- If we know the ratio:

$$\beta_i \stackrel{\text{def}}{=} \frac{p(\mathbf{x}_i)}{q(\mathbf{x}_i)}$$

- Then we train model to minimize weighted empirical risk:

$$\underset{f}{\text{minimize}} \frac{1}{n} \sum_{i=1}^{n} \beta_i l(f(\mathbf{x}_i), y_i)$$

- In practice we do not know β. But if we can sample from $p(\boldsymbol{x})$ then we can learn β.

Learning β Correction Using Logistic Regression

- Suppose points drawn from p is labelled $z = 1$, and from q is labelled $z = -1$ (our training data)
- Then

$$P(z=1 \mid \mathrm{x}) = \frac{p(\mathrm{x})}{p(\mathrm{x}) + q(\mathrm{x})} \text{ and hence } \frac{P(z=1 \mid \mathrm{x})}{P(z=-1 \mid \mathrm{x})} = \frac{p(\mathrm{x})}{q(\mathrm{x})}$$

- Using logistic regression:

$$P(z=1 \mid \mathrm{x}) = \frac{1}{1+\exp(-h(\mathrm{x}))}$$

- Then:

$$\beta_i = \frac{1/(1+\exp(-h(\mathrm{x}_i)))}{\exp(-h(\mathrm{x}_i))/(1+\exp(-h(\mathrm{x}_i)))} = \exp(h(\mathrm{x}_i))$$

A Taxonomy of Learning Problems

- Batch Learning
  - Learn using entire labelled training data.
  - No distributional shift. Learn once, deploy, and never have to change.
- Online Learning
  - Learning one sample at a time $(\boldsymbol{x}_i, y_i)$
  - Continuously adjust model according to success of predicting y_i
- Control
  - Learn actions to control the environment (e.g., control car air conditioner)
  - Adjust model based on the response environment (hot parked car → max AC)
- Reinforcement Learning
  - Learn how environment behaves and learn policies on how to act
  - Environment responds in complex ways, e.g., adversarial for competitive games (chess, Go) or cooperative (allowing autonomous car to change lanes)

Fairness, Accountability and Transparency in Machine Learning

- Machine learning model are being used to guild decision making in the real world
- Before deploying a model
  - Analyze the impact of using model
  - Make sure its decisions are appropriate for various subpopulations
  - Setup a governance structure for monitoring and management
- COMPAS: a deployed ML system for criminal risk assessment
  - Systematically gave black defendants higher risk,
  - Of all defendants receiving higher risk black defendants have lower percentage of recidivism
- Runaway feedback
  - After reading a few webpages on a conspiracy theory, systems recommends more webpages about conspiracy theory
  - Predictive policing system repeatedly sends patrol to the same neighborhoods