Announcements

- Don’t Cheat
- Proposals due tonight

©Kevin Jamieson 2018

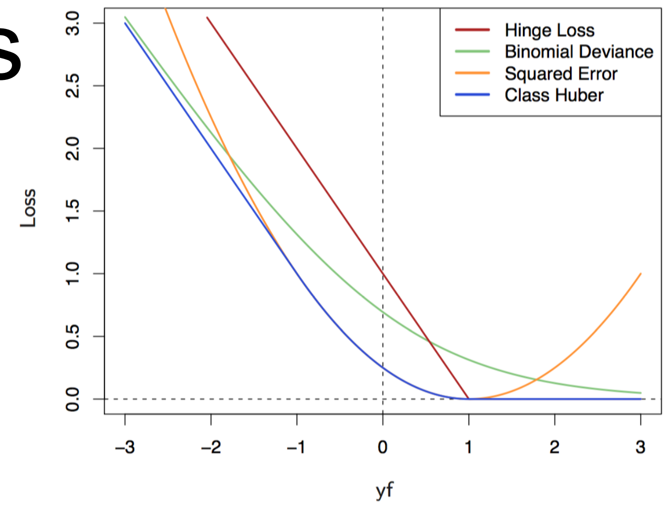
Logistic, SVM, and Perceptron

Machine Learning – CSE546
Kevin Jamieson
University of Washington

October 25, 2018

©2018 Kevin Jamieson

Machine Learning Problems



- Have a bunch of iid data of the form:

$$\{(x_i, y_i)\}_{i=1}^n \qquad x_i \in \mathbb{R}^d \qquad y_i \in \mathbb{R}$$

- Learning a model's parameters: $$\sum_{i=1}^n \ell_i(w)$$

  Each $\ell_i(w)$ is convex.

Hinge Loss: $\ell_i(w) = \max\{0, 1 - y_i x_i^T w\}$

Logistic Loss: $\ell_i(w) = \log(1 + \exp(-y_i\, x_i^T w))$

Squared error Loss: $\ell_i(w) = (y_i - x_i^T w)^2$

How do we solve for *w*? The last two lectures!

Perceptron is optimizing what?

Perceptron update rule:

$$\begin{bmatrix} w_{k+1} \\ b_{k+1} \end{bmatrix} = \begin{bmatrix} w_k \\ b_k \end{bmatrix} + y_k \begin{bmatrix} x_k \\ 1 \end{bmatrix} \mathbf{1}\{y_i(b + x_i^T w) < 0\}$$

SVM objective:

$$\boxed{\sum_{i=1}^n \max\{0, 1 - y_i(b + x_i^T w)\} + \lambda ||w||_2^2 \; = \sum_{i=1}^n \ell_i(w,b)}$$

$$\nabla_w \ell_i(w,b) = \begin{cases} -x_i y_i + \frac{2\lambda}{n} w & \text{if } y_i(b + x_i^T w) < 1 \\ \frac{2\lambda}{n} & \text{otherwise} \end{cases}$$

$$\nabla_b \ell_i(w,b) = \begin{cases} -y_i & \text{if } y_i(b + x_i^T w) < 1 \\ 0 & \text{otherwise} \end{cases}$$

Perceptron is almost SGD on SVM with $\lambda = 0$, $\eta = 1$!

SVMs vs logistic regression

- We often want probabilities/confidences, logistic wins here?

©2018 Kevin Jamieson

SVMs vs logistic regression

- We often want probabilities/confidences, logistic wins here?
- No! Perform isotonic regression or non-parametric bootstrap for probability calibration. Predictor gives some score, how do we transform that score to a probability?

©2018 Kevin Jamieson

Bootstrap

Machine Learning – CSE546
Kevin Jamieson
University of Washington

October 25, 2018

©2018 Kevin Jamieson

Limitations of CV

- An 80/20 split throws out a relatively large amount of data if only have, say, 20 examples.
- Test error is informative, but how accurate is this number? (e.g., 3/5 heads vs. 30/50)
- How do I get confidence intervals on statistics like the median or variance of a distribution?
- Instead of the error for the entire dataset, what if I want to study the error for a *particular example* x?

©2018 Kevin Jamieson

Limitations of CV

- An 80/20 split throws out a relatively large amount of data if only have, say, 20 examples.
- Test error is informative, but how accurate is this number? (e.g., 3/5 heads vs. 30/50)
- How do I get confidence intervals on statistics like the median or variance of a distribution?
- Instead of the error for the entire dataset, what if I want to study the error for a *particular example* x?

The Bootstrap: Developed by Efron in 1979.

"The most important innovation in statistics of the last 40 years"

— famous ML researcher and statistician, 2015

©2018 Kevin Jamieson

Bootstrap: basic idea

Given dataset drawn iid samples with CDF F_Z:

$$\mathcal{D} = \{z_1, \dots, z_n\} \overset{i.i.d.}{\sim} F_Z$$

We compute a *statistic* of the data to get: $\widehat{\theta} = t(\mathcal{D})$

Bootstrap: basic idea

Given dataset drawn iid samples with CDF F_Z:

$$\mathcal{D} = \{z_1, \dots, z_n\} \overset{i.i.d.}{\sim} F_Z$$

We compute a *statistic* of the data to get: $\widehat{\theta} = t(\mathcal{D})$

For b=1,…,B define the *b*th ***bootstrapped*** dataset as drawing *n* samples **with replacement** from *D*

$$\mathcal{D}^{*b} = \{z_1^{*b}, \dots, z_n^{*b}\} \overset{i.i.d.}{\sim} \widehat{F}_{Z,n}$$

and the *b*th bootstrapped statistic as: $\theta^{*b} = t(\mathcal{D}^{*b})$

©2018 Kevin Jamieson

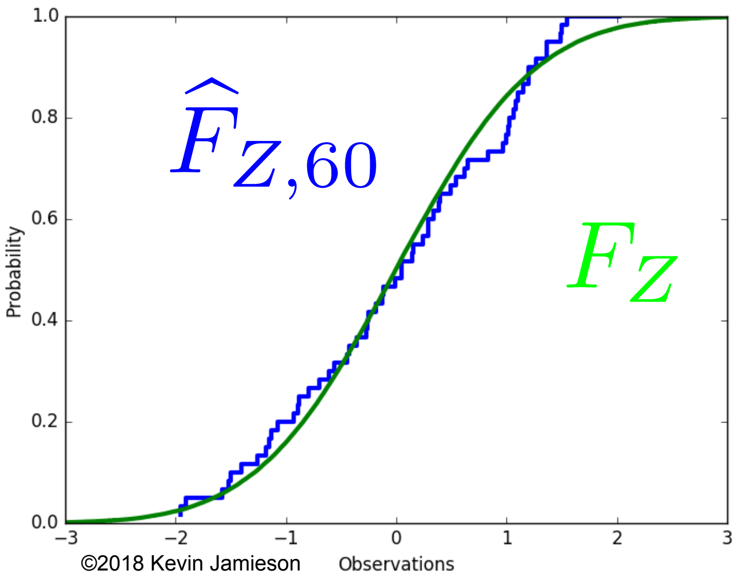
Bootstrap: basic idea

Given dataset drawn iid samples with CDF F_Z:

$$\mathcal{D} = \{z_1, \ldots, z_n\} \overset{i.i.d.}{\sim} F_Z \qquad \widehat{\theta} = t(\mathcal{D})$$

For b=1,…,B, samples sampled **with replacement** from *D*

$$\mathcal{D}^{*b} = \{z_1^{*b}, \ldots, z_n^{*b}\} \overset{i.i.d.}{\sim} \widehat{F}_{Z,n} \qquad \theta^{*b} = t(\mathcal{D}^{*b})$$



©2018 Kevin Jamieson

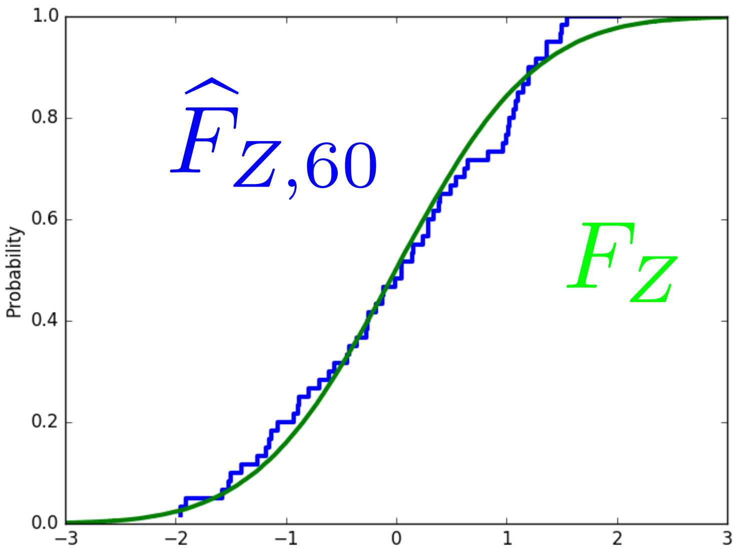
Bootstrap: basic idea

Given dataset drawn iid samples with CDF F_Z:

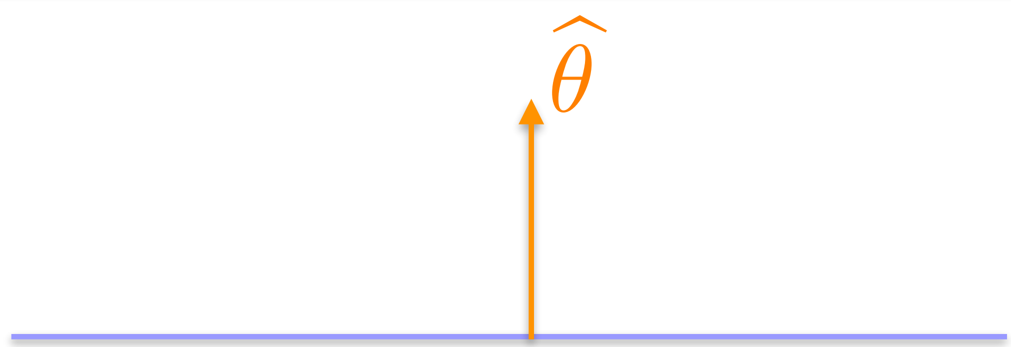
$$\mathcal{D} = \{z_1, \ldots, z_n\} \overset{i.i.d.}{\sim} F_Z \qquad \widehat{\theta} = t(\mathcal{D})$$

For b=1,…,B, samples sampled **with replacement** from *D*

$$\mathcal{D}^{*b} = \{z_1^{*b}, \ldots, z_n^{*b}\} \overset{i.i.d.}{\sim} \widehat{F}_{Z,n} \qquad \theta^{*b} = t(\mathcal{D}^{*b})$$



$$\boxed{\sup_x |\widehat{F}_n(x) - F(x)| \to 0 \quad \text{as } n \to \infty}$$



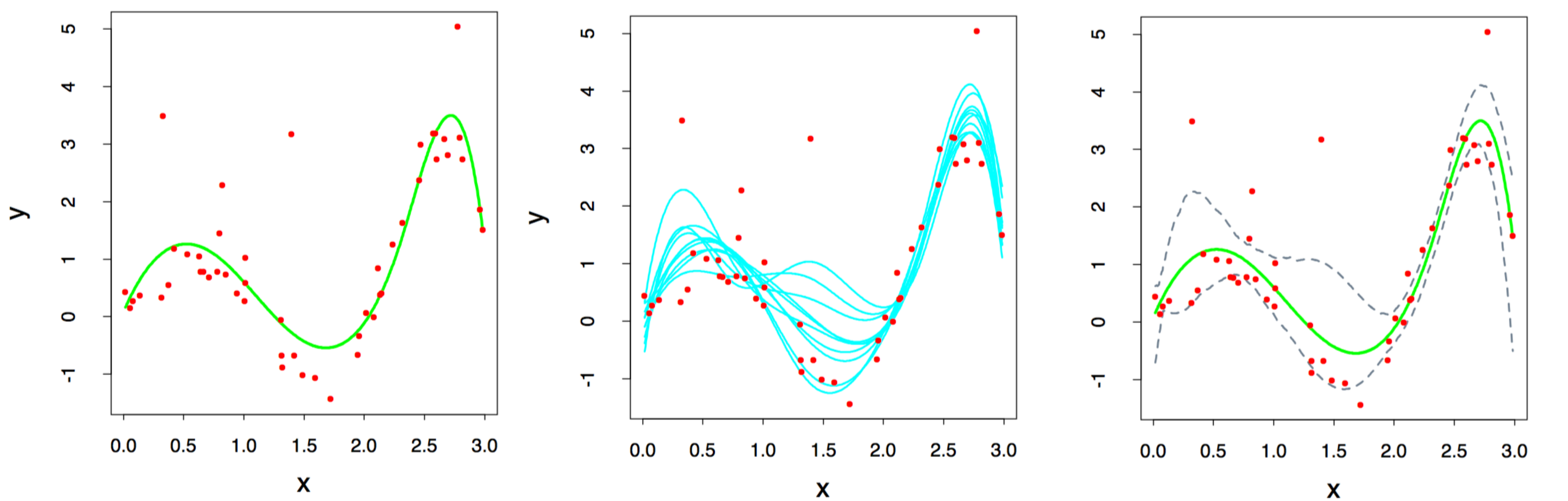
©2018 Kevin Jamieson

Applications

Common applications of the bootstrap:

- Estimate parameters that escape simple analysis like the variance or median of an estimate
- Confidence intervals
- Estimates of error for a particular example:

$\mathcal{D}$ $\widehat{\theta}$ θ^{*b} for $b = 1, \ldots, 10$ 95% confidence interval



Figures from Hastie et al

©2018 Kevin Jamieson

Takeaways

Advantages:

- Bootstrap is **very** generally applicable. Build a confidence interval around ***anything***
- **Very** simple to use
- Appears to give meaningful results even when the amount of data is very small
- Very strong **asymptotic theory** (as num. examples goes to infinity)

©2018 Kevin Jamieson

Takeaways

Advantages:

- Bootstrap is **very** generally applicable. Build a confidence interval around ***anything***
- **Very** simple to use
- Appears to give meaningful results even when the amount of data is very small
- Very strong **asymptotic theory** (as num. examples goes to infinity)

Disadvantages

- Very few meaningful finite-sample guarantees
- Potentially **computationally intensive**
- Reliability relies on test statistic and rate of convergence of empirical CDF to true CDF, which is unknown
- Poor performance on “extreme statistics” (e.g., the max)

Not perfect, but better than nothing.

©2018 Kevin Jamieson

Warm up: risk prediction with logistic regression

- Boss gives you a bunch of data on loans defaulting or not:

$$\{(x_i, y_i)\}_{i=1}^n \quad x_i \in \mathbb{R}^d, \quad y_i \in \{-1, 1\}$$

- You model the data as: $$P(Y = y|x, w) = \frac{1}{1 + \exp(-y\, w^T x)}$$
- And compute the maximum likelihood estimator:

$$\widehat{w}_{MLE} = \arg\max_w \prod_{i=1}^n P(y_i|x_i, w)$$

For a new loan application x, boss recommends to give loan if your model says they will repay it with probability at least .95 (i.e. low risk):

Give loan to x if $$\frac{1}{1 + \exp(-\widehat{w}_{MLE}^T x)} \geq .95$$

- One year later only half of loans are paid back and the bank folds. What might have happened?

How would you use the bootstrap to do this differently?

©Kevin Jamieson 2018

Decision Theory

Machine Learning – CSE546
Kevin Jamieson
University of Washington

October 25, 2018

©2018 Kevin Jamieson

Binary Classification

- **Learn**: f:**X —>**Y
  - **X** – features
  - Y – target classes

    $Y \in \{-1, 1\}$

- **Loss function:**

  $\ell(f(x), y) = \mathbf{1}\{f(x) \neq y\}$

- **Expected loss of f:**

$$\mathbb{E}_{XY}[\mathbf{1}\{f(X) \neq Y\}] = \mathbb{E}_X[\mathbb{E}_{Y|X}[\mathbf{1}\{f(x) \neq Y\}|X = x]]$$

$$\mathbb{E}_{Y|X}[\mathbf{1}\{f(x) \neq Y\}|X = x] = 1 - P(Y = f(x)|X = x)$$

- **Bayes optimal classifier:**

$$f(x) = \arg\max_y \mathbb{P}(Y = y|X = x)$$

- **Model of logistic regression:**

$$P(Y = y|x, w) = \frac{1}{1 + \exp(-y\, w^T x)}$$

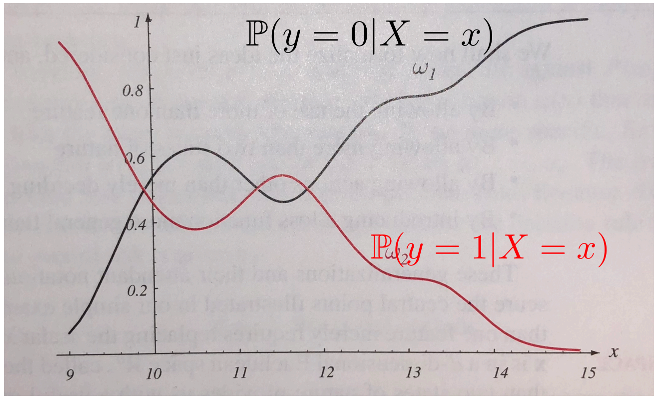
Binary Classification

- **Learn**: f:**X —>**Y
  - **X** – features
  - Y – target classes

$$Y \in \{-1, 1\}$$

- **Bayes optimal classifier:**

$$\boxed{f(x) = \arg\max_y \mathbb{P}(Y = y | X = x)}$$



Binary Classification

- **Learn**: f:**X —>**Y
  - **X** – features
  - Y – target classes

$$Y \in \{-1, 1\}$$

- **Bayes optimal classifier:**

$$f(x) = \arg\max_y \mathbb{P}(Y = y|X = x)$$

$$f(x) = \arg\max_y \mathbb{P}(X = x|Y = y)\mathbb{P}(Y = y)$$

Bayes rule: $$\mathbb{P}(Y = y|X = x) = \frac{\mathbb{P}(X = x|Y = y)\mathbb{P}(Y = y)}{P(X = x)}$$

©Kevin Jamieson 2018

Binary Classification

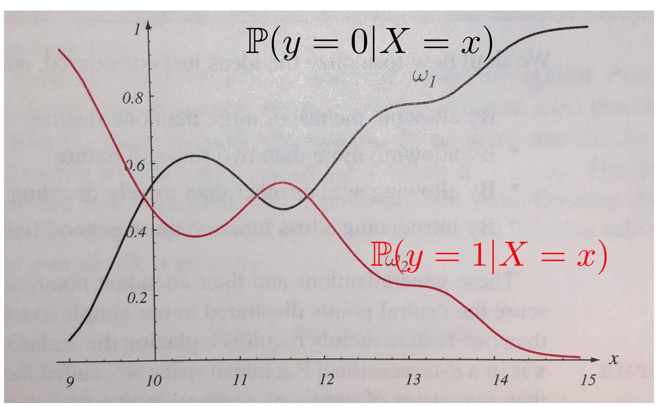
- **Learn**: f:**X** —>Y
  - **X** – features
  - Y – target classes

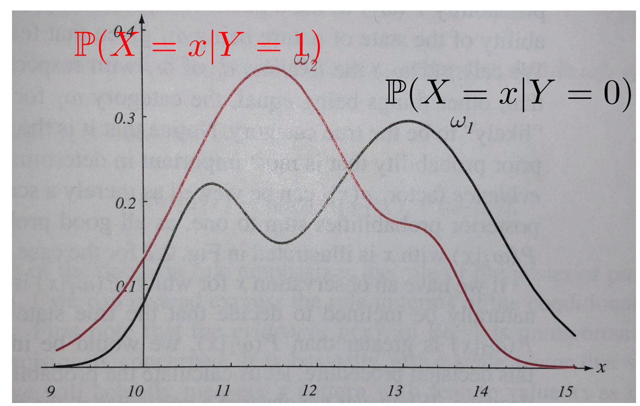
$$Y \in \{-1, 1\}$$

- **Bayes optimal classifier:**

$$\boxed{f(x) = \arg\max_y \mathbb{P}(Y = y | X = x)}$$

$$\boxed{f(x) = \arg\max_y \mathbb{P}(X = x | Y = y)\mathbb{P}(Y = y)}$$





$\mathbb{P}(y = 1) = 1/3$ $\mathbb{P}(y = 0) = 2/3$

©Kevin Jamieson 2018

Binary classification: Gaussians

Let
$$\mathbb{P}(X=x) = \mathbb{P}(X=x|Y=0)\mathbb{P}(Y=0) + \mathbb{P}(X=x|Y=1)\mathbb{P}(Y=1)$$
$$=: (1-\pi)P_0(x) + \pi P_1(x)$$

Suppose $P_0(x) = \mathcal{N}(x; \mu_0, \sigma^2)$ $\quad P_1(x) = \mathcal{N}(x; \mu_1, \sigma^2)$

$$\begin{aligned} f(x) &= \arg\max_y \mathbb{P}(Y=y|X=x) \\ &= \arg\max_y \mathbb{P}(X=x|Y=y)\mathbb{P}(Y=y) \end{aligned}$$

$$f(x) = 1 \text{ if } \frac{P_1(x)\pi}{P_0(x)(1-\pi)} \geq 1$$

©2018 Kevin Jamieson

Binary classification: Gaussians

Let
$$\mathbb{P}(X=x) = \mathbb{P}(X=x|Y=0)\mathbb{P}(Y=0) + \mathbb{P}(X=x|Y=1)\mathbb{P}(Y=1)$$
$$=: (1-\pi)P_0(x) + \pi P_1(x)$$

Suppose $P_0(x) = \mathcal{N}(x;\mu_0,\sigma^2)$ $\quad P_1(x) = \mathcal{N}(x;\mu_1,\sigma^2)$

$$\boxed{f(x) = 1 \text{ if } \frac{P_1(x)\pi}{P_0(x)(1-\pi)} \geq 1}$$

$$f(x) = 1 \text{ if } \tfrac{\mu_1-\mu_0}{\sigma^2}\left(x - \tfrac{\mu_1+\mu_0}{2}\right) \geq -\log\left(\tfrac{\pi}{1-\pi}\right)$$

$$f(x) = 1 \text{ if } x \geq \tfrac{\mu_1+\mu_0}{2} - \frac{\sigma^2}{\mu_1-\mu_0}\log\left(\tfrac{\pi}{1-\pi}\right)$$

©2018 Kevin Jamieson

Binary classification: Gaussians

Let
$$\mathbb{P}(X=x) = \mathbb{P}(X=x|Y=0)\mathbb{P}(Y=0) + \mathbb{P}(X=x|Y=1)\mathbb{P}(Y=1)$$
$$=: (1-\pi)P_0(x) + \pi P_1(x)$$

Suppose $P_0(x) = \mathcal{N}(x; \mu_0, \sigma^2)$ $\quad P_1(x) = \mathcal{N}(x; \mu_1, \sigma^2)$

$$f(x) = 1 \text{ if } \frac{P_1(x)\pi}{P_0(x)(1-\pi)} \geq 1$$

$$f(x) = 1 \text{ if } x \geq \tfrac{\mu_1+\mu_0}{2} - \tfrac{\sigma^2}{\mu_1-\mu_0}\log(\tfrac{\pi}{1-\pi})$$

$\pi = 1/2$ $\quad \pi \in (1/2, 1)$ $\quad \pi \in (0, 1/2)$

$\frac{\mu_1+\mu_0}{2}$ $\quad \frac{\mu_1+\mu_0}{2}$ $\quad \frac{\mu_1+\mu_0}{2}$

©2018 Kevin Jamieson

Binary classification: Gaussians

Same ideas extend to higher dimensions:

$$P_1(x) = \mathcal{N}(x; \mu_1, \Sigma_1) \qquad P_0(x) = \mathcal{N}(x; \mu_0, \Sigma_0)$$

$$\boxed{f(x) = 1 \text{ if } \frac{P_1(x)\pi}{P_0(x)(1-\pi)} \geq 1}$$

Cases:

$$\Sigma_0 = \Sigma_1 :$$

$$\Sigma_0 \neq \Sigma_1 :$$

©2018 Kevin Jamieson

Binary classification: Gaussians

Same ideas extend to higher dimensions:

$$P_1(x) = \mathcal{N}(x; \mu_1, \Sigma_1) \qquad P_0(x) = \mathcal{N}(x; \mu_0, \Sigma_0)$$

$$\boxed{f(x) = 1 \text{ if } \frac{P_1(x)\pi}{P_0(x)(1-\pi)} \geq 1}$$

In practice we observe $\{(x_i, y_i)\}_{i=1}^n$

$$\widehat{\mu}_k = \frac{1}{|\{i : y_i = k\}|} \sum_{i:y_i=k} x_i$$

$$\widehat{\Sigma}_k = \frac{1}{|\{i : y_i = k\}| - 1} \sum_{i:y_i=k} (x_i - \widehat{\mu}_k)(x_i - \widehat{\mu}_k)^T$$

©2018 Kevin Jamieson

Binary Classification

- **Learn**: f:**X —>**Y
  - **X** – features
  - Y – target classes

$$Y \in \{-1, 1\}$$

- **Bayes optimal classifier:**

$$\boxed{f(x) = \arg\max_y \mathbb{P}(Y = y | X = x)}$$

$$\boxed{f(x) = \arg\max_y \mathbb{P}(X = x | Y = y)\mathbb{P}(Y = y)}$$

Discriminative learning directly models $\mathbb{P}(Y = y | X = x)$

Example:

Generative learning models $\mathbb{P}(X = x, Y = y) = \mathbb{P}(X = x | Y = y)\mathbb{P}(Y = y)$

Example:

©Kevin Jamieson 2018

Hypothesis testing

Machine Learning – CSE546

Kevin Jamieson
University of Washington

October 25, 2018

©2018 Kevin Jamieson

Anomaly detection

You are Amazon and wish to detect transactions with stolen credit cards.

For each transaction we observe a **feature vector X**:
{ email-address, age of account, anonymous PO box, price of items, copies of purchased item, etc. }
and the transaction is either **real (Y=0)** or **fraudulent (Y=1)**

Hypothesis testing:

H0: $X \sim P_0$

H1: $X \sim P_1$

$$P_k = \mathbb{P}(X = x | Y = k)$$

Your job is to build a (possibly randomized) decision function $\delta(x) \in \{0, 1\}$

$$\mathbb{P}(X = x) = \pi \mathbb{P}_1(x) + (1 - \pi) \mathbb{P}_0(x)$$

©Kevin Jamieson 2018

Anomaly detection

You are Amazon and wish to detect transactions with stolen credit cards.

For each transaction we observe a **feature vector X**:
{ email-address, age of account, anonymous PO box, price of items, copies of purchased item, etc. }
and the transaction is either **real (Y=0)** or **fraudulent (Y=1)**

Hypothesis testing:

H0: $X \sim P_0$

H1: $X \sim P_1$

$$P_k = \mathbb{P}(X = x | Y = k)$$

Your job is to build a (possibly randomized) decision function $\delta(x) \in \{0, 1\}$

Bayesian Hypothesis Testing:

Assume $\mathbb{P}(Y = 1) = \pi$

$$\mathbb{P}(X = x) = \pi P_1(x) + (1 - \pi) P_0(x)$$

$$\arg\min_{\delta} \mathbb{P}_{XY}(Y \neq \delta(X))$$

Anomaly detection

You are Amazon and wish to detect transactions with stolen credit cards.

For each transaction we observe a **feature vector X**:
{ email-address, age of account, anonymous PO box, price of items, copies of purchased item, etc. }
and the transaction is either **real (Y=0)** or **fraudulent (Y=1)**

Hypothesis testing:

H0: $X \sim P_0$

H1: $X \sim P_1$

$$P_k = \mathbb{P}(X = x | Y = k)$$

Your job is to build a (possibly randomized) decision function $\delta(x) \in \{0, 1\}$

Minimax Hypothesis Testing:

$$\arg\min_{\delta} \max\{\mathbb{P}(\delta(X) = 0 | Y = 1), \mathbb{P}(\delta(X) = 1 | Y = 0)\}$$

©Kevin Jamieson 2018

Anomaly detection

You are Amazon and wish to detect transactions with stolen credit cards.

For each transaction we observe a **feature vector X**:
{ email-address, age of account, anonymous PO box, price of items, copies of purchased item, etc. }
and the transaction is either **real (Y=0)** or **fraudulent (Y=1)**

Hypothesis testing:

H0: $X \sim P_0$

H1: $X \sim P_1$

$$P_k = \mathbb{P}(X = x | Y = k)$$

Your job is to build a (possibly randomized) decision function $\delta(x) \in \{0, 1\}$

Neyman-Pearson Hypothesis Testing:

$$\arg\max_{\delta} \mathbb{P}(\delta(X) = 1 | Y = 1), \text{ subject to } \mathbb{P}(\delta(X) = 1 | Y = 0) \leq \alpha\}$$

©Kevin Jamieson 2018

Neyman-Pearson Testing

Hypothesis testing:

H0: $X \sim P_0$

H1: $X \sim P_1$

$$P_k = \mathbb{P}(X = x | Y = k)$$

Neyman-Pearson Hypothesis Testing:

$$\arg\max_{\delta} \mathbb{P}(\delta(X) = 1 | Y = 1), \text{ subject to } \mathbb{P}(\delta(X) = 1 | Y = 0) \leq \alpha\}$$

Theorem: The optimal test δ^* has the form

$$\mathbb{P}(\delta^*(X) = 1) = \begin{cases} 1 & \text{if } \frac{P_1(x)}{P_0(x)} > \eta \\ \gamma & \text{if } \frac{P_1(x)}{P_0(x)} = \eta \\ 0 & \text{if } \frac{P_1(x)}{P_0(x)} < \eta \end{cases}$$

and satisfies $\mathbb{P}(\delta^*(X) = 1 | Y = 0) = \alpha$

©Kevin Jamieson 2018