CSC2515 Midterm Review

Haotian cui

Feb 23, 2021

Midterm Review

1. A brief overview

2. Some past midterm questions

- **Supervised learning and Unsupervised learning**

  Supervised learning: have a collection of training examples labeled with the correct outputs

  Unsupervised learning: have no labeled examples

- **Regression and Classification**

  Regression: predicting a scalar-valued target

  Classification: predicting a discrete-valued target

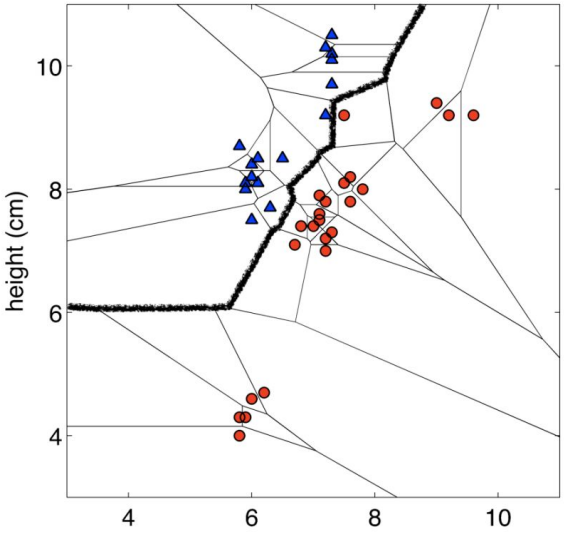
- **K-Nearest Neighbors**

Idea: Classify a new input **x** based on its k nearest neighbors in the training set

Decision boundary: the boundary between regions of input space assigned to different categories

Tradeoffs in choosing k: overfit / underfit

Pitfalls: curse of dimensionality, normalization, computational cost



Pitfalls: Computational Cost

- Number of computations at training time: 0
- Number of computations at test time, per query (naïve algorithm)
  - Calculuate D-dimensional Euclidean distances with N data points: $\mathcal{O}(ND)$
  - Sort the distances: $\mathcal{O}(N \log N)$
- This must be done for *each* query, which is very expensive by the standards of a learning algorithm!
- Need to store the entire dataset in memory!

- **Decision Trees**

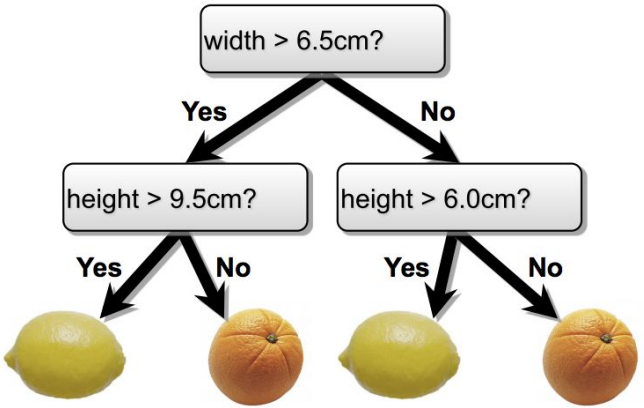
Model: make predictions by splitting on features according to a tree structure

Decision boundary: made up of axis-aligned planes

Entropy: uncertainty inherent in the variable's possible outcomes $H(Y) = -\sum_{y \in Y} p(y) \log_2 p(y)$

joint entropy; conditional entropy; properties

Information gain: $IG(Y|X) = H(Y) - H(Y|X)$ measures the informativeness of a variable; used to choose a good split



- **Linear Regression**

Model: a linear function of the features $y = \mathbf{w}^\top \mathbf{x} + b$

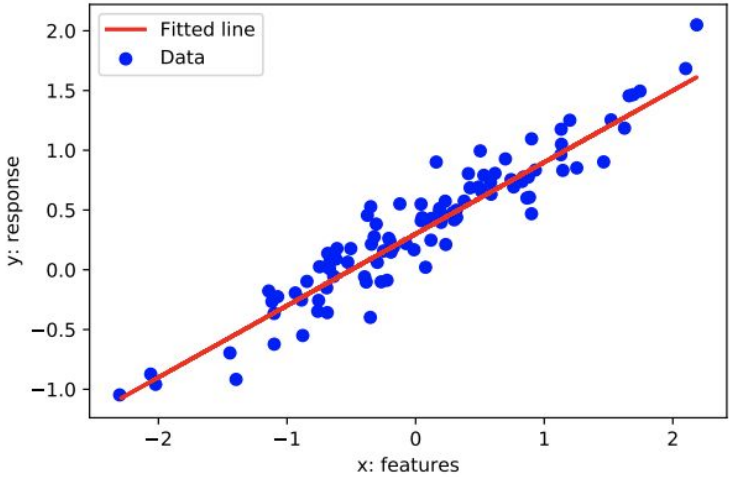
Loss function: squared error loss $\mathcal{L}(y, t) = \frac{1}{2}(y - t)^2$

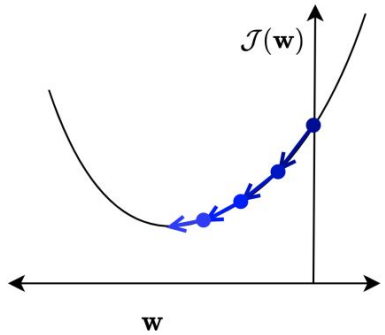
Cost function: loss function averaged over all training examples

Vectorization: advantages

Solving minimization problem: direct solution / gradient descent $\mathbf{w} \leftarrow \mathbf{w} - \alpha \frac{\partial \mathcal{J}}{\partial \mathbf{w}}$

Feature mapping: degree-M polynomial feature mapping





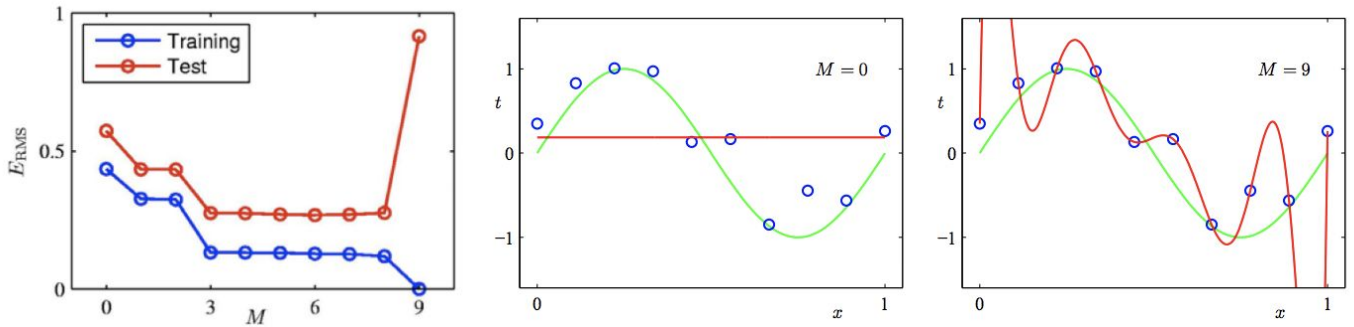
- **Model Complexity and Generalization**

Underfitting: too simplistic to describe the data

Overfitting: too complex, fit training examples perfectly, but fails to generalize to unseen data

Hyperparameter: can't include in the training procedure itself, tune it using a validation set

Regularization: $\mathcal{J}_{\text{reg}}(\mathbf{w}) = \mathcal{J}(\mathbf{w}) + \lambda \mathcal{R}(\mathbf{w})$, improve the generalization, L2 / L1 regularization



-Pattern Recognition and Machine Learning, Christopher Bishop.

- **Linear Classification**
  - **Binary Linear Classification**

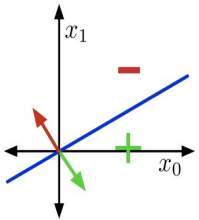
Model: $z = \mathbf{w}^\top \mathbf{x}$

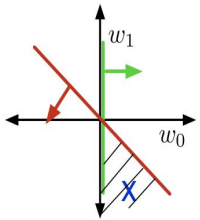
$$y = \begin{cases} 1 & \text{if } z \geq 0 \\ 0 & \text{if } z < 0 \end{cases}$$

Geometry: input space, weight space

Loss function: 0-1 loss $\mathcal{L}_{0-1}(y,t) = \begin{cases} 0 & \text{if } y = t \\ 1 & \text{if } y \neq t \end{cases}$

$= \mathbb{I}[y \neq t]$





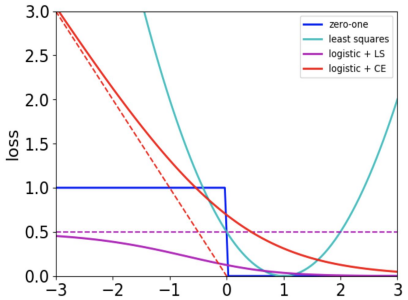
$w_0 \geq 0$

$w_0 + w_1 < 0$

  - **Logistic Regression**

Model: $z = \mathbf{w}^\top \mathbf{x}$

$y = \sigma(z)$



Loss function: 0-1 loss

→ squared error loss $\mathcal{L}_{\text{SE}}(z,t) = \frac{1}{2}(z-t)^2$

→ logistic + squared error loss $\mathcal{L}_{\text{SE}}(y,t) = \frac{1}{2}(y-t)^2.$

→ logistic + cross-entropy loss $\mathcal{L}_{\text{CE}} = -t \log y - (1-t)\log(1-y)$

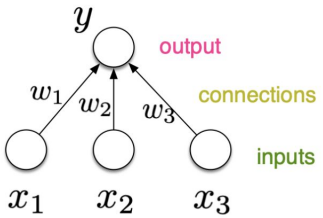
  - **Softmax Regression**

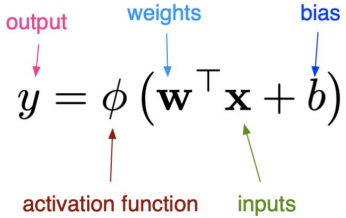
Multi-class classification

$$y_k = \text{softmax}(z_1, \ldots, z_K)_k = \frac{e^{z_k}}{\sum_{k'} e^{z_{k'}}}$$

$$\mathbf{z} = \mathbf{W}\mathbf{x}$$
$$\mathbf{y} = \text{softmax}(\mathbf{z})$$
$$\mathcal{L}_{\text{CE}} = -\mathbf{t}^\top (\log \mathbf{y})$$

- **Neural Networks**





Model: $\mathbf{y} = f^{(L)} \circ \cdots \circ f^{(1)}(\mathbf{x})$.

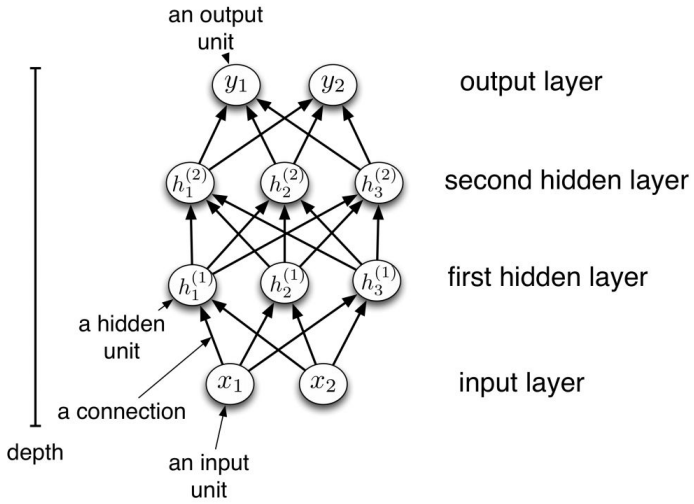
Unit, layer, weights, activation functions

Each first-layer hidden unit acts as a feature detector.

Expressivity: universal function approximators (non-linear activation functions); Pros/Cons

Regularization: early stopping

Backpropagation: efficiently computing gradients in neural nets



Other topics to know

- Comparisons between different classifiers (KNN, logistic regression, decision trees, neural networks)
- Contrast the decision boundaries for different classifiers
- Draw computation graph and use backpropagation to compute the derivatives of a loss function

2018 Midterm Version A Q7

7. [**2pts**] Consider the classification problem with the following dataset:

x_1	x_2	x_3	t
0	0	0	1
0	1	0	0
0	1	1	1
1	1	1	0

Your job is to find a linear classifier with weights w_1, w_2, w_3, and b which correctly classifies all of these training examples. None of the examples should lie on the decision boundary.

(a) [**1pt**] Give the set of linear inequalities the weights and bias must satisfy.

(b) [**1pt**] Give a setting of the weights and bias that correctly classifies all the training examples. You don't need to show your work, but it might help you get partial credit.

Solution

x_1	x_2	x_3	t
0	0	0	1
0	1	0	0
0	1	1	1
1	1	1	0

$$t = 1, w_1x_1 + w_2x_2 + w_3x_3 + b > 0$$
$$t = 0, w_1x_1 + w_2x_2 + w_3x_3 + b < 0$$

Many answers are possible.
Here's one:

$$\begin{cases} w_1 \cdot 0 + w_2 \cdot 0 + w_3 \cdot 0 + b > 0 \\ w_1 \cdot 0 + w_2 \cdot 1 + w_3 \cdot 0 + b < 0 \\ w_1 \cdot 0 + w_2 \cdot 1 + w_3 \cdot 1 + b > 0 \\ w_1 \cdot 1 + w_2 \cdot 1 + w_3 \cdot 1 + b < 0 \end{cases} \Rightarrow \begin{cases} b > 0 & b = 1 \\ w_2 + b < 0 & w_1 = -2 \\ w_2 + w_3 + b > 0 & w_2 = -2 \\ w_1 + w_2 + w_3 + b < 0 & w_3 = 2 \end{cases}$$

2018 Midterm Version B Q7

7. **[2pts]** Suppose binary-valued random variables X and Y have the following joint distribution:

	$Y=0$	$Y=1$
$X=0$	$1/8$	$3/8$
$X=1$	$2/8$	$2/8$

Determine the information gain $IG(Y|X)$. You may write your answer as a sum of logarithms.

Conditional Entropy

	Cloudy	Not Cloudy
Raining	24/100	1/100
Not Raining	25/100	50/100

- The expected conditional entropy:

$$\begin{aligned} H(Y|X) &= \sum_{x \in X} p(x) H(Y|X=x) \\ &= -\sum_{x \in X} \sum_{y \in Y} p(x,y) \log_2 p(y|x) \end{aligned}$$

Solution

	$Y=0$	$Y=1$
$X=0$	$1/8$	$3/8$
$X=1$	$2/8$	$2/8$

$$p(Y=0) = p(X=0,Y=0) + p(X=1,Y=0) = \tfrac{3}{8}$$

$$p(Y=1) = p(X=0,Y=1) + p(X=1,Y=1) = \tfrac{5}{8}$$

$$p(X=0) = p(X=0,Y=0) + p(X=0,Y=1) = \tfrac{1}{2}$$

$$p(X=1) = p(X=1,Y=0) + p(X=1,Y=1) = \tfrac{1}{2}$$

$$\begin{aligned} p(Y=0|X=0) &= \frac{p(Y=0,X=0)}{p(X=0)} \\ &= \frac{p(Y=0,X=0)}{p(X=0,Y=0)+p(X=0,Y=1)} \\ &= \frac{1}{4} \end{aligned}$$

We used: $p(y|x) = \frac{p(x,y)}{p(x)}$ and $p(x) = \sum_y p(x,y)$

$$IG(Y|X) = H(Y) - H(Y|X)$$

$$\begin{aligned} H(Y) &= -\sum_y p(Y=y)\log_2 p(Y=y) \\ &= -p(Y=0)\log_2 p(Y=0) - p(Y=1)\log_2 p(Y=1) \\ &= -\frac{3}{8}\log_2\frac{3}{8} - \frac{5}{8}\log_2\frac{5}{8} \end{aligned}$$

$$\begin{aligned} H(Y|X) &= \sum_x p(X=x)H(Y|X=x) \\ &= p(X=0)H(Y|X=0) + p(X=1)H(Y|X=1) \\ &= \frac{1}{2}H(Y|X=0) + \frac{1}{2}H(Y|X=1) \end{aligned}$$

$$H(Y|X=x) = -\sum_y p(y|x)\log_2 p(y|x)$$

$$\begin{aligned} H(Y|X=0) &= -p(Y=0|X=0)\log_2 p(Y=0|X=0) \\ &\quad - p(Y=1|X=0)\log_2 p(Y=1|X=0) \\ &= -\frac{1}{4}\log_2\frac{1}{4} - \frac{3}{4}\log_2\frac{3}{4} \end{aligned}$$

$$H(Y|X=1) = -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2}$$

KNN (19 Fall midterm Q7)

- When we analyzed KNN, we assumed the training examples were sampled densely enough so that the true conditional probability $p(t \mid \mathbf{x})$ is approximately constant in the vicinity of a query point $\mathbf{x}_\star$. Suppose it is a binary classification task with targets $t \in \{0, 1\}$ and $p(t = 1 \mid \mathbf{x}_\star) = 0.6$.
- What is the asymptotic error rate at $\mathbf{x}_\star$ for a 1-nearest-neighbor classifier? (By asymptotic, I mean as the number of training examples $N \to \infty$.) Justify your answer.
  Let $t_\star$ denote the true target and t_N denote the target at the nearest neighbor. These are independent Bernoulli random variables with parameter 0.6. The classifier makes a mistake if $t_\star = 0$ and $t_N = 1$ or if $t_\star = 1$ and $t_N = 0$. Hence, the probability of a mistake, i.e. the error rate, is $0.4 \cdot 0.6 + 0.6 \cdot 0.4 = 0.48$.

KNN (19 Fall midterm Q7)

- When we analyzed KNN, we assumed the training examples were sampled densely enough so that the true conditional probability $p(t \mid \mathbf{x})$ is approximately constant in the vicinity of a query point $\mathbf{x}_\star$. Suppose it is a binary classification task with targets $t \in \{0, 1\}$ and $p(t = 1 \mid \mathbf{x}_\star) = 0.6$.
  For large K, the asymptotic KNN error rate is approximately the Bayes error rate. In this example, the Bayes classifier will predict $y = 1$. Hence, the error rate is 0.4.

Bias Variance (modified from CSC2515 19 midterm Q4)

- Carol and Dave are each trying to predict stock prices using neural networks. They formulate this as a regression problem using squared error loss. Carol trains a single logistic regression model on a certain training set and uses its predictions on the test set. Dave trains 5 different models (using exactly the same architecture, training data, etc. as Carol) starting with different random initializations, and averages their predictions on the test set.
  For each of the following questions, please briefly and informally justify your answer. You do not need to provide a mathematical proof.
- [4 points] Compared with Carol's approach, is the Bayes error for Dave's approach HIGHER, LOWER, or THE SAME?
  THE SAME. The Bayes error is a property of the data generating distribution, and doesn't depend on the algorithm that was used.

Bias Variance (modified from CSC2515 19 midterm Q4)

- Carol and Dave are each trying to predict stock prices using neural networks. They formulate this as a regression problem using squared error loss. Carol trains a single logistic regression model on a certain training set and uses its predictions on the test set. Dave trains 5 different models (using exactly the same architecture, training data, etc. as Carol) starting with different random initializations, and averages their predictions on the test set.
  For each of the following questions, please briefly and informally justify your answer. You do not need to provide a mathematical proof.
- Compared with Carol's approach, is the bias for Dave's approach HIGHER, LOWER, or THE SAME?
  THE SAME. Sampling multiple hypotheses from the same distribution and averaging their predictions doesn't change the expected predictions due to linearity of expectation. Hence it doesn't change the bias.

Bias Variance (modified from CSC2515 19 midterm Q4)

- Compared with Carol's approach, is the variance for Dave's approach HIGHER, LOWER, or THE SAME?
  LOWER. Averaging over multiple samples reduces the variance of the predictions, even if those samples are not fully independent. (In this case, they're not fully independent as they share the same training set.)

1. **Bias, Variance, and Bayes Error.** The purpose of this exercise is to show a simple example where you can compute the bias, variance, and Bayes error of a predictor. For this question, we assume we have N scalar-valued observations $\{x^{(i)}\}_{i=1}^{N}$ sampled independently from a Gaussian distribution $\mathcal{N}(x; \mu, \sigma^2)$ with known variance σ^2 and unknown mean μ. We'd like to estimate the mean parameter μ, or equivalently, choose a $\hat{\mu}$ which minimizes the squared error risk $\mathbb{E}[(x - \hat{\mu})^2]$.

   We'll introduce the Gaussian distribution properly in a later lecture, but hopefully you've seen it before in a probability course. It is a bell-shaped distribution whose density is:

$$p(x; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right).$$

   The details of the Gaussian distribution (such as the density) aren't important for this exercise. The important facts are that $\mathbb{E}[x] = \mu$ and $\mathrm{Var}(x) = \sigma^2$).

   We will estimate the unknown mean paramter μ by taking the empirical mean, or average, of the observations:

$$\hat{\mu} = \frac{1}{N} \sum_{i=1}^{N} x^{(i)}.$$

Q1: Decomposition

- Decompose the mean squared error (MSE) of sample mean.

$$\mathbb{E}[(x - \hat{\mu})^2]$$

- Take expectation w.r.t. $x \sim N(x; \mu, \sigma^2)$

$$\begin{aligned}
\mathbb{E}_x[(x - \hat{\mu})^2] &= \mathbb{E}[x^2 - 2x\hat{\mu} + \hat{\mu}^2] \\
&= \mathbb{E}[x^2] - 2\hat{\mu}\mathbb{E}[x] + \hat{\mu}^2 \\
&= \mathrm{Var}[x] + \ \mathbb{E}[x]^2 - 2\hat{\mu}\mathbb{E}[x] + \hat{\mu}^2 \\
&= (\mathbb{E}[x] - \hat{\mu})^2 + \mathrm{Var}[x] \\
&= (\mu - \hat{\mu})^2 + \mathrm{Var}[x]
\end{aligned}$$

Q1: Decomposition

- Take expectation w.r.t estimator $\hat{\mu}$
  - Estimator is a random variable since the training data its generated from is randomly drawn from the true distribution

$$
\begin{aligned}
\mathbb{E}_{\hat{\mu}}[\mathbb{E}_x[(x-\hat{\mu})^2]] &= \mathbb{E}\left[(\mu-\hat{\mu})^2 + \mathrm{Var}[x]\right] \\
&= \mathbb{E}[(\mu-\hat{\mu})^2] + \mathrm{Var}[x] \\
&= \mathbb{E}[(\mu^2 - 2\mu\hat{\mu} + \hat{\mu}^2)] + \mathrm{Var}[x] \\
&= \mu^2 - 2\mu\mathbb{E}[\hat{\mu}] + \mathbb{E}[\hat{\mu}^2] + \mathrm{Var}[x] \\
&= \mu^2 - 2\mu\mathbb{E}[\hat{\mu}] + \mathbb{E}[\hat{\mu}]^2 + \mathrm{Var}[\hat{\mu}] + \mathrm{Var}[x] \\
&= (\mu - \mathbb{E}[\hat{\mu}])^2 + \mathrm{Var}[\hat{\mu}] + \mathrm{Var}[x]
\end{aligned}
$$

Q1: Problem Statement

- Find exact bias, variance, Bayes error of sample mean MSE
  - Bias: $(\mu - \mathbb{E}[\hat{\mu}])^2$
  - Variance: $\mathrm{Var}[\hat{\mu}]$
  - Bayes Error: $\mathbb{E}(x - \mu)^2$

- Use properties of expectation / variance
- Remember that $\mathbb{E}[x] = \mu, \mathrm{Var}[x] = \sigma^2$
- Also remember $\hat{\mu}$ is our sample mean estimator, meaning its defined by the equation in the handout

Q1: Bias Solution

$$(\mu - \mathbb{E}[\hat{\mu}])^2$$

Looks like we need $\mathbb{E}[\hat{\mu}]$

$$\mathbb{E}[\hat{\mu}] = \mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N} x_i\right] = \frac{1}{N}\sum_{i=1}^{N} \mathbb{E}[x_i] = \frac{1}{N}\sum_{i=1}^{N} \mu = \frac{1}{N}(N\mu) = \mu$$

Substituting back in

$$(\mu - \mathbb{E}[\hat{\mu}])^2 = (\mu - \mu)^2 = 0$$

Q1: Bias Solution

- Since $(\mu - \mathbb{E}[\hat{\mu}])^2 = 0$, it is an unbiased estimator

- Estimators which have bias $= 0$ are unbiased, and vice versa
  - Example of biased estimator: Trying to estimate an unknown variance via

$$S^2 = \frac{1}{n}\sum_i (x_i - \hat{\mu})^2$$

Q1: Variance Solution

$$\mathrm{Var}[\hat{\mu}] = \mathrm{Var}\left[\frac{1}{N}\sum_{i=1}^{N} x_i\right] = \frac{1}{N^2}\mathrm{Var}\left[\sum_{i=1}^{N} x_i\right] = \frac{1}{N^2}\sum_{i=1}^{N}\mathrm{Var}[x_i]$$

$$= \frac{1}{N^2}\sum_{i=1}^{N}\sigma^2 = \frac{1}{N^2}(\mathrm{N}\,\sigma^2)$$

- Aside: This can be converted into the standard error formula by square rooting both sides. Pretty cool connection!

Q1: Bayes Error Solution

- Note that we already obtained Bayes error of $\mathrm{Var}[x] = \sigma^2$ in decomposition. Starting from handout equation…

$$
\begin{aligned}
\mathbb{E}(x-\mu)^2 &= \mathbb{E}[x^2 - 2x\mu + \mu^2] \\
&= \mathbb{E}[x^2] - 2\mu\mathbb{E}[x] + \mathbb{E}[\mu^2] \\
&= \mathbb{E}[x]^2 + \mathrm{Var}[x] - 2\mu\mathbb{E}[x] + \mathbb{E}[\mu^2] \\
&= \mu^2 + \sigma^2 - 2\mu\mu + \mu^2 \\
&= 2\mu^2 - 2\mu^2 + \sigma^2 \\
&= \sigma^2
\end{aligned}
$$

Q2: Entropy Properties Part (a)

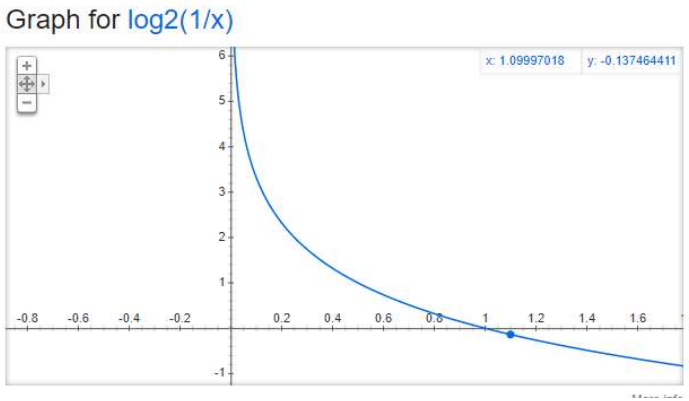
- Prove entropy $H(X)$ is non-negative

$$H(X) = \sum_{x} p(x) \log_2 \left(\frac{1}{p(x)} \right)$$

- X is a discrete random variable. Thus:
  - $p(x_i) \geq 0$
  - $\sum_{x \in X} p(x) = 1$
- The two conditions also imply $p(x_i) \leq 1$

Q2: Entropy Properties Part (a)

- Since $0 \leq p(x_i) \leq 1, \log_2\left(\frac{1}{p(x)}\right) \geq 0$



- We are basically done.
- $H(X) = \sum_x p(x) \log_2\left(\frac{1}{p(x)}\right)$

Non-negative Non-negative

Sums of non-negative values will remain non-negative

Q2: Entropy Properties Part (b)

Prove $H(X,Y) = H(X\,|Y) + H(Y)$

$$H(X,Y) = \sum_x \sum_y p(x,y) \log_2 \left(\frac{1}{p(x,y)}\right)$$

$$= -\sum_x \sum_y p(x,y) \log_2 p(x,y)$$

$$= -\sum_x \sum_y p(x,y) \log(p(y|x)p(x))$$

$$= -\sum_x \sum_y p(x,y)(\log p(y|x) + \log p(x)) \qquad \text{Log product identity}$$

$$= -\sum_x \sum_y p(x,y) \log p(y|x) \; - \sum_x \sum_y p(x,y) \log p(x) \qquad \text{By commutativity and associativity of summation}$$

Q2: Entropy Properties Part (b)

$$H(X,Y) = -\sum_x \sum_y p(x,y) \log p(y|x) \; - \sum_x \sum_y p(x,y) \log p(x)$$

$$= -\sum_x \sum_y p(x,y) \log p(y|x) - \sum_x \log p(x) \sum_y p(x,y)$$

Since $\log p(x)$is not dependent on y

$$= -\sum_x \sum_y p(x,y) \log p(y|x) - \sum_x \log p(x) \left(p(x)\right)$$

Marginalizing out y

$$= -\sum_x \sum_y p(x,y) \log p(y|x) + H(X)$$

By definition of $H(X)$

Q2: Entropy Properties Part (b)

$$H(X,Y) = -\sum_x \sum_y p(x,y) \log p(y|x) + H(X)$$

$$= -\sum_x \sum_y p(y|x)p(x) \log p(y|x) + H(X)$$

$$= -\sum_x p(x) \sum_y p(y|x) \log p(y|x) + H(X)$$

Since $p(x)$is not dependent on y

$$= -\sum_x p(x) \left(-H(Y|X=x)\right) + H(X)$$

By definition of $H(Y|X=x)$

To show the other way around, we can do equivalent proof, but note $H(Y|X) \neq H(X|Y)$ in general.

Q2: Entropy Properties Part (c)

- Prove $H(X,Y) \geq H(X)$

- We know that $H(X) \geq 0$, and $H(X,Y) = H(Y|X) + H(X)$

- Non rigorous demonstration
  - If $H(Y|X) = 0$, then $H(X,Y) = H(X)$
  - If $H(Y|X) > 0$, then $H(X,Y) \geq H(X,Y) - H(Y|X) = H(X)$
  - $H(Y|X)$ cannot be less than 0 [proof similar to part (a)]