CSC311 Midterm Review

October 15, 2020

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

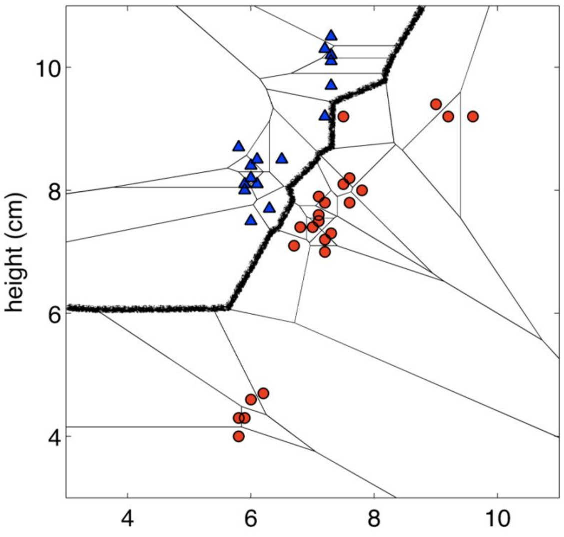
- **K-Nearest Neighbors**

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

<u>Decision boundary</u>: the boundary between regions of input space assigned to different categories

<u>Tradeoffs in choosing k</u>: overfit / underfit

<u>Pitfalls</u>: curse of dimensionality, normalization, computational cost



- **Linear Regression**

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

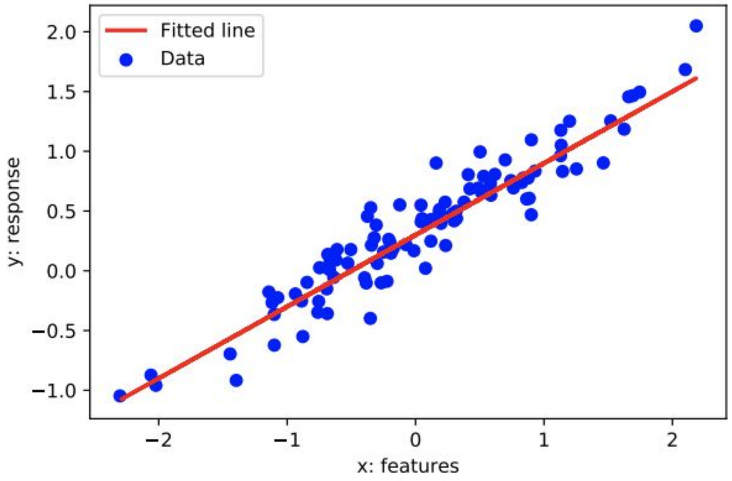
  Loss function: squared error loss $\mathcal{L}(y,t) = \frac{1}{2}(y-t)^2$

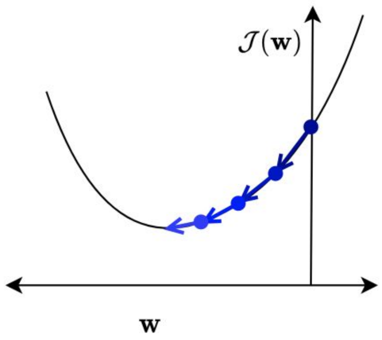
  Cost function: loss function averaged over all training examples

  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





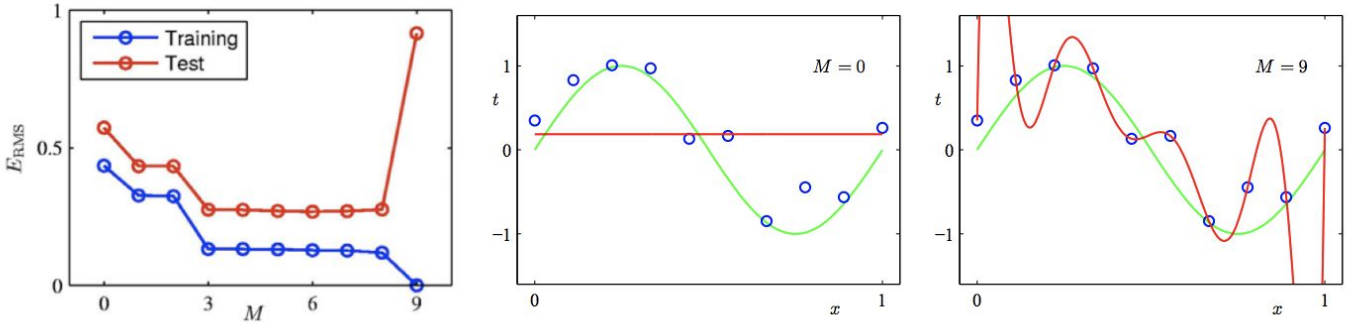
- **Model Complexity and Generalization**

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

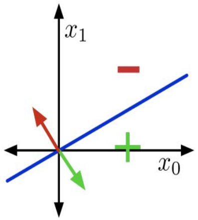
Model: $z = \mathbf{w}^\top \mathbf{x}$

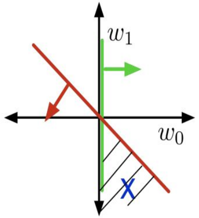
$$y = \begin{cases} 1 & \text{if } z \geq 0 \\ 0 & \text{if } z < 0 \end{cases}$$

Geometry: input space, weight space

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

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

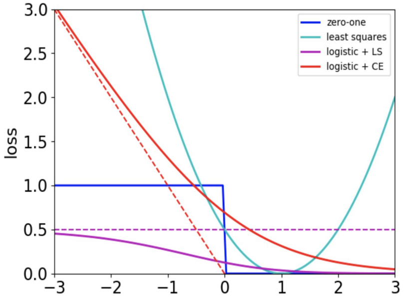




$w_0 \geq 0$

$w_0 + w_1 < 0$

- ## Logistic Regression



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

$y = \sigma(z)$

Loss function: 0-1 loss

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

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

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

- ## Softmax Regression

Multi-class classification

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

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

- **Neural Networks**

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

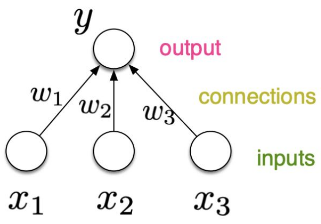
  Unit, layer, weights, activation functions

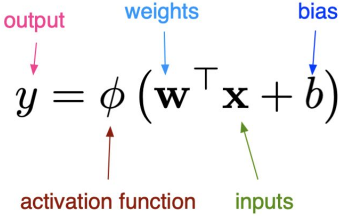
  Each first-layer hidden unit acts as a feature detector.

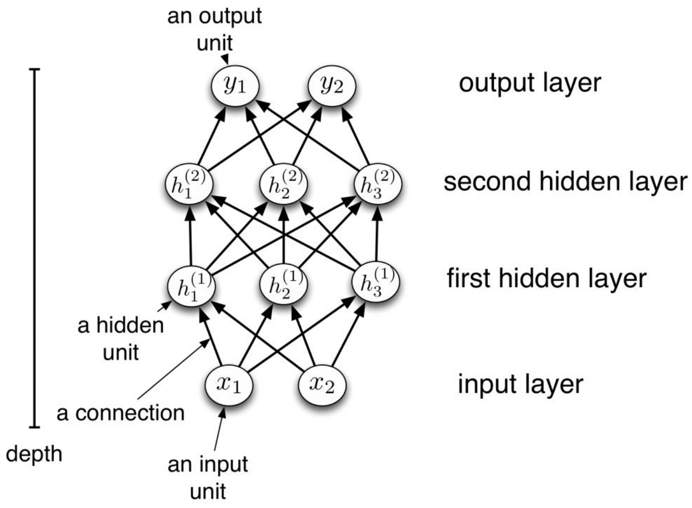
  Expressivity: universal function approximators (non-linear activation functions); Pros/Cons

  Regularization: early stopping

  Backpropagation: efficiently computing gradients in neural nets







- **Decision Trees**

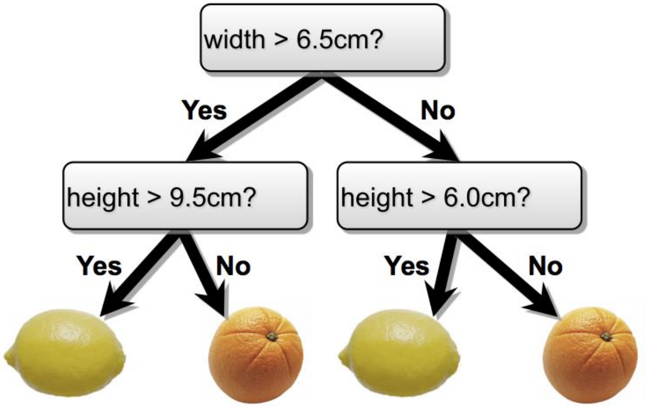
<u>Model</u>: make predictions by splitting on features according to a tree structure

<u>Decision boundary</u>: made up of axis-aligned planes

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

<u>joint entropy; conditional entropy; properties</u>

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



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

$$t = 1, w_1x_1 + w_2x_2 + w_3x_3 + b \boxed{>} 0$$
$$t = 0, w_1x_1 + w_2x_2 + w_3x_3 + b < 0$$

Many answers are possible.
Here's one:

$$\begin{cases} w_1 \cdot 0 + w_2 \cdot 0 + w_3 \cdot 0 + b > 0 \\ w_1 \cdot 0 + w_2 \cdot 1 + w_3 \cdot 0 + b < 0 \\ w_1 \cdot 0 + w_2 \cdot 1 + w_3 \cdot 1 + b > 0 \\ w_1 \cdot 1 + w_2 \cdot 1 + w_3 \cdot 1 + b < 0 \end{cases} \Longrightarrow \begin{cases} b > 0 \\ w_2 + b < 0 \\ w_2 + w_3 + b > 0 \\ w_1 + w_2 + w_3 + b < 0 \end{cases}$$

$$b = 1$$
$$w_1 = -2$$
$$w_2 = -2$$
$$w_3 = 2$$

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.

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 $p(x)=\sum_y p(x,y)$

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

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

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

$$H(Y|X=x)=\boxed{-}\textstyle\sum_y p(y|x)\log_2 p(y|x)$$

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

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