Midterm Review

CSC411

Nitish Srivastava

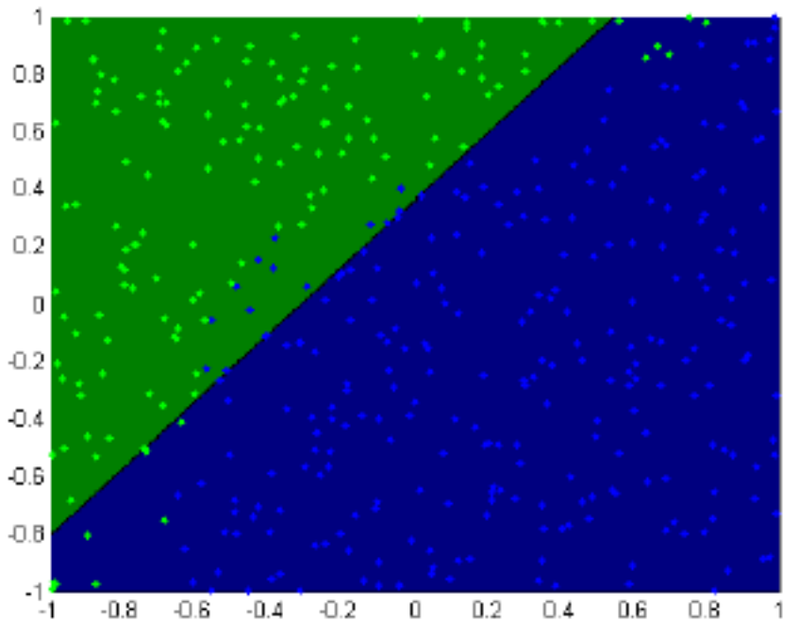
Oct 17 2014

Midterm Topics

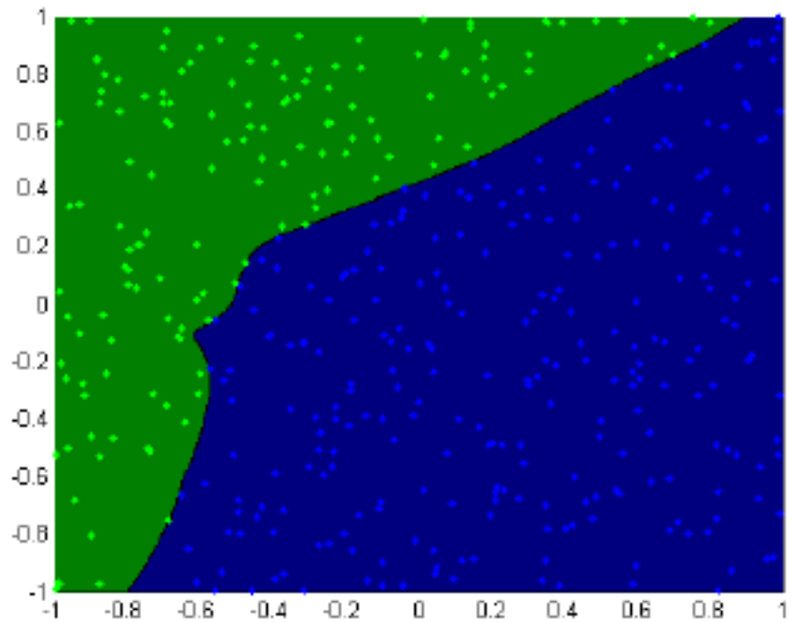
- Generative classifiers
- Logistic regression: understand loss function, with penalty, as in hw1, compare decision boundaries
- Naïve Bayes: basic assumption, decision rule
- K-nearest neighbors – when (in)appropriate
- Decision trees: basic alg, representation; entropy, information gain
- Neural networks: design simple ones for sample problems; when backprop feasible; methods of max. generalization
- Loss functions: why may want different functions
- Overfitting: regularizers, complexity penalties
- Bayes Rule; MAP and ML hypotheses
- Cross-validation

Sample Question 1

- One of the curves is from a logistic regression classifier and the other one is from a neural network.
- **Which one do you think is which, and why?**



(a)



(b)

Answer

a. is logistic regression, b. is neural network

- **Logistic regression** classifiers have **linear** decision boundaries because $p(y = 1|x) = \sigma(\mathbf{w}^T x + b)$.
- **Neural networks** have **nonlinear** decision boundaries due to nonlinearities in the hidden layers.

Sample Question 2

Why is it more difficult to learn the parameters of a neural network than a logistic regression classifier?

Answer

Optimization

- Learning the parameters for a **neural network** is a **non-convex** optimization problem. $\Rightarrow$ Easy to get stuck in local minima!
- Learning for **logistic regression** is a **convex** optimization problem $\Rightarrow$ We can always find the globally optimal solution.

Computation

- Computing gradients for a neural network is **more expensive**.
- We must first **forward propagate** the input to compute the error...
- then **back-propagate** the error to compute gradients for each parameter.

Sample Question 3

The "flexibility" of a neural network, it's ability to model different functions, is given by the number of hidden units.

If we wanted to, we could simply use millions (i.e. a lot) of hidden units in order to model any kind of function we wanted.

Why is this a bad idea in general? How could we avoid this problem?

Answer

There are two reasons why we don't want to use too many hidden units.

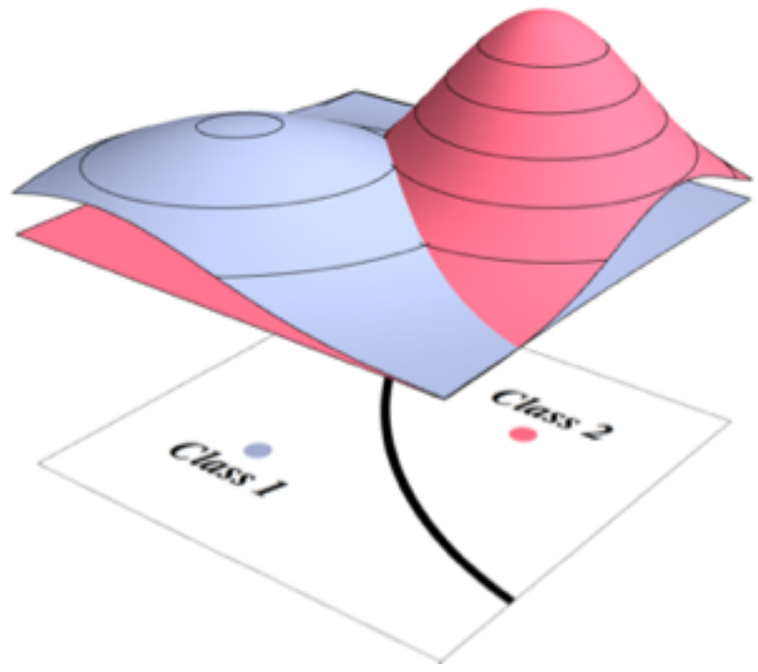
1. It will be very easy to **overfit** the training data.
2. Learning and prediction will be **expensive**.

Ways to mitigate these concerns:

- Limit the number of hidden units we use.
- Use **regularization** e.g. weight decay (L2 penalty on the weights)

Decision Boundaries

- Logistic Regression : Linear.
- Naive bayes: Depends on form of P(attr|class).
  Piece-wise linear or quadratic if Gaussian.

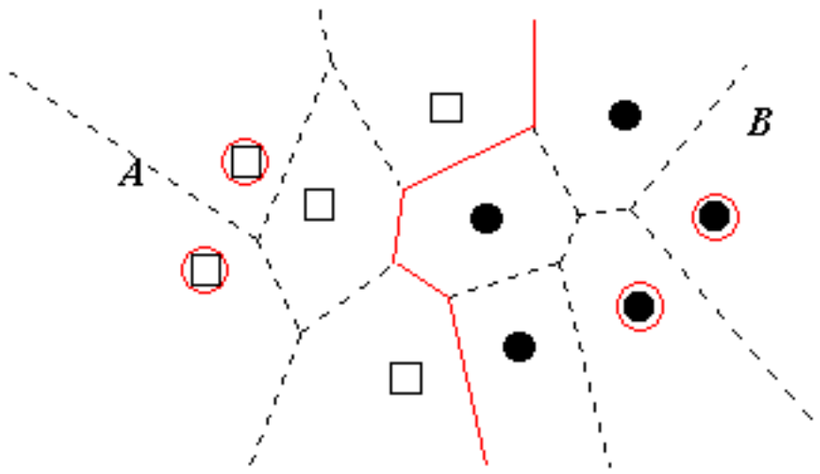


Decision Boundaries

- K-NN : Piece-wise linear. (If Euclidean distance is the measure of nearness).

- Neural Nets : Very flexible! (depends on number of hidden units and depth).

- Decision Trees : Axis-aligned, linear. Why ?



Sample Question 7

What kinds of data are expected to be (in)appropriate for a K-nearest neighbor classifier?

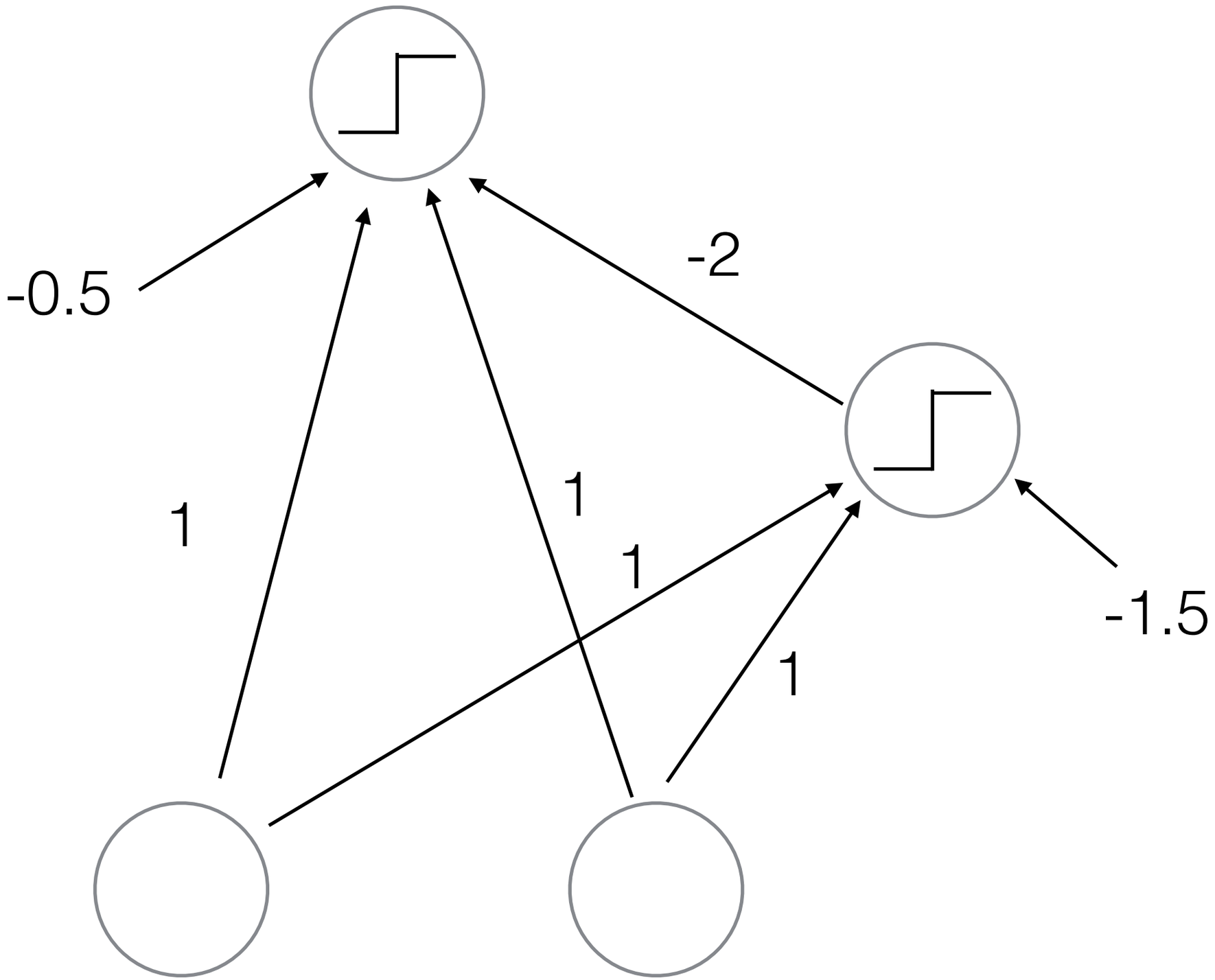
Answer

Considerations

- KNN handles non-linearly separable classes much better than logistic regression.
- Notion of distance becomes important.
    - Features with larger ranges $\rightarrow$ normalize scale.
    - Irrelevant or correlated features $\rightarrow$ may have to eliminate or weight.
    - Distances become larger for higher dimensions.
- Must store all training cases $\rightarrow$ becomes an issue for large training set size
- Sensitive to class noise

Construct a neural net

- Design a network to implement the XOR function.



Also try OR, AND, NOT, combinations of them

Last years midterm

- Q1 : Bayes Rules, Naive Bayes.
- Q2 : Decision Trees, Information Gain.
- Q3 : Neural nets.
- Q4 : Overfitting, cross validation.

CSC 411 MID-TERM

Name (1 point): Student Number (1 point):

Please check that your exam has 5 pages, including this one. Use the back of the page if you need more space on a question. The answer to each lettered question needs no more than two sentences.

1. Imagine that you want to decide if the Leafs are going to win or lose their next hockey game. You are going to base this decision on a dataset that contains the following data for each of their last 1000 games: opponent, day of the week, location, and outcome. You have the same information for the next game, except the outcome.

   (A). (6 points) Write down the equation that you would use to to compute the maximum a posteriori (MAP) outcome for the game (recall Bayes Rule: $P(A|B) = P(B|A)P(A)/P(B)$). Be sure to define which variables correspond to which quantities in the database.

   need to know prob of every combo of var = (opponent, day,location,outcome)

   choose argmax(C=win,lose) P(var—C)

   [3 pts] variables [3 pts] eqn

   (B). (6 points) Write down the decision rule equation for the a Naive Bayes classifier for this problem. Define the key simplifying assumption in the Naive Bayes method, and explain why you think it is or is not applicable here.

   prob of each var (day, location, opponent) cond indy given outcome

   $P(d, l, o|c) = P(d|c)P(l|c)P(o|c)$

   unlikely true (location depends on opponent regardless of outcome) but may be good enough to get good predictor

   [2 pts for cond indep assumption, 2 for multiplication idea, 2 for reasonable applicable]

   (C). (6 points) How well would you expect a nearest-neighbor classifier to do on this problem? Explain your answer.

2. Suppose you want to build a decision tree for a problem. In the dataset, there are two classes, with 150 examples in the $+$ class and 50 examples in the $-$ class.

   Recall the definitions of information gain and entropy:

$$\begin{aligned} Entropy(C) &\equiv H(C) = \sum_c -P(C=c)\log_2 P(C=c) \\ Gain(C,A) &= H(C) - \sum_{v \in Values(A)} P(A=v)H(C|A=v) \end{aligned}$$

   (A). (6 points) What is the entropy of the class variable (you can leave this in terms of $\log$s)?

$$
\begin{aligned}
H(C) &= -\frac{150}{150+50}\log\frac{150}{150+50} - \frac{50}{150+50}\log\frac{50}{150+50} \\
&= -\frac{3}{4}\log\frac{3}{4} - \frac{1}{4}\log\frac{1}{4}
\end{aligned}
$$

(B). (6 points) For this data, suppose the $Color$ attribute takes on one of 3 values (red, green, and blue), and the split into the two classes across $red/green/blue$ is $+ : (120/10/20)$ and $- : (0/10/40)$. Write down an expression for the class entropy in the subset containing all $green$ examples. Is this entropy greater or less than the entropy in the previous question?

$$
\begin{aligned}
H(C|Color = green) &= -\frac{10}{10+10}\log\frac{20}{10+10} - \frac{10}{10+10}\log\frac{10}{10+10} \\
&= -\frac{1}{2}\log\frac{1}{2} - \frac{1}{2}\log\frac{1}{2} = 1
\end{aligned}
$$

[3 for either eqn]

Greater entropy [3 pts]

(C). (6 points) Is $Color$ a good attribute to add to the tree? Explain your answer.

Yes – the other two attribute values have almost no entropy

(D). (6 points) What is the information gain for a particular attribute if every value of the attribute has the same ratio between the number of $+$ examples and the total number of examples?

conditional entropies all the same, so must be the same as $H(C)$, so gain is 0

constant [3pts] zero [3 more]

3. Neural Networks

Consider the following learning rule:

$$
w_{ji}^{\texttt{new}} = w_{ji}^{\texttt{old}} - \eta \sum_n (o_j^{(n)} - t_j^{(n)}) x_i^{(n)}
$$

(A). (6 points) Define each of the five terms on the right-hand side of the learning rule.

[1 each (+ 1)]

- old weight
- learn rate
- output of unit j on example n
- target of unit j on example n
- value of input i on example n

(B). (6 points) Imagine that another term is added, producing this new learning rule:

$$w_{ji}^{\texttt{new}} = w_{ji}^{\texttt{old}} - \eta \sum_n (o_j^n - t_j^n) x_i^n - 2\alpha w_{ji}^{\texttt{old}}$$

What is the main aim of such a term? What effect does this term have on the network weights?

weight decay. used to control generalization – make wts smaller.

momentum (-6)

(C). (6 points) Explain whether or not a linear classifier can be used to learn the following (Input (3 features);Output) association from these four examples: { (- + - ; +) (+ - + ; -) (+ + + ; -) (- - - ; +) }

Yes - Output equals opposite of last Input.

4. Suppose you have a dataset of labeled examples for training a machine learning system.

   (A). (4 points) Define a *validation set*.

   pseudo test set [3], examples taken out of training set, not used for training but to eval generalization [3]

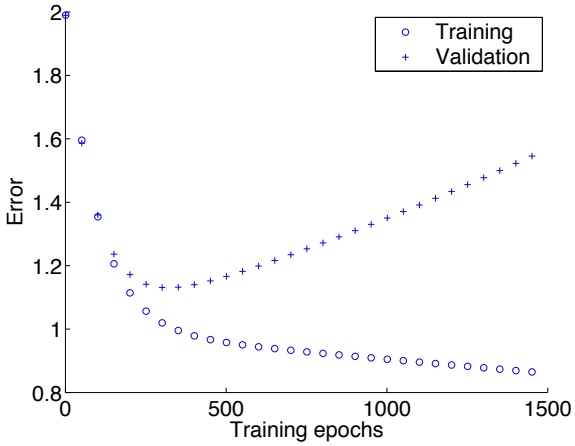
   (B). (4 points) Describe the trade-offs involved in assigning examples to the validation set versus the training set (for example, compare a 50/50 training/validation split of the data to a 75/25 split).

   [3 for each, 1 for extra]

   - more training instances: more representative of true distn; less overfitting
   - more val. instances: better evaluation of generalization ability (more accurate – more representative of true distn)

   extras - redundancy could lead to slower training in big train set

   (C). (4 points) Imagine that a neural net is trained using batch gradient descent on a 50/50 training/validation split, and the following error curves show how the training and validation set errors vary with the number of training epochs.



Suppose that you were to retrain the network using the same algorithm and split, but with ten times as much available data. Would you expect the training curve to be different? If so draw what you would expect. You only need to give a qualitative sketch.