CSC321 Lecture 3
Binary linear classification

Roger Grosse and Nitish Srivastava

January 14, 2015

Overview

Last lecture was about regression (predicting a real value)
This week is about classification (predicting a discrete value).

- Lecture 1: How to think about binary classification (regardless of the algorithm) .
- Lecture 2: The perceptron learning algorithm.

Overview

Linear regression vs. perceptron

Linear regression	**Perceptron**
predict real values	predict binary value
optimization problem	constraint satisfaction problem
closed-form solution	iterative procedure

Overview

Linear regression vs. perceptron

Linear regression	**Perceptron**
predict real values	predict binary value
optimization problem	constraint satisfaction problem
closed-form solution	iterative procedure

Some commonalities

- Limitations to what we can predict with linear models.
- But we can make them more powerful using basis functions.

Your questions from the quiz

Course organization

- Focus on video lectures or in-class?
  - Class meetings typically expand on the material in the video lectures.
  - Also about 6 traditional lectures, which cover different material and which you are responsible for.

Your questions from the quiz

Course organization

- Focus on video lectures or in-class?
  - Class meetings typically expand on the material in the video lectures.
  - Also about 6 traditional lectures, which cover different material and which you are responsible for.
- Are all video lectures going to be as long as this week?
  - We had two weeks worth of videos due this week – future weeks will have 45 minutes to 1 hour of video.

Your questions from the quiz

Course organization

- Focus on video lectures or in-class?
  - Class meetings typically expand on the material in the video lectures.
  - Also about 6 traditional lectures, which cover different material and which you are responsible for.
- Are all video lectures going to be as long as this week?
  - We had two weeks worth of videos due this week – future weeks will have 45 minutes to 1 hour of video.
- How important are proofs in the course?
  - What's important is that you can justify things informally – we're not going to focus on rigorous mathematical proofs.

Your questions from the quiz

Common themes (we'll cover these this week)

- threshold and bias, and how they're related ($\times 4$)
- examples of perceptron training ($\times 6$)
- weight space visualizations ($\times 5$)
- more details on convergence/limitations proofs ($\times 6$)
- how perceptrons and basis functions relate to neural nets ($\times 3$)

We'll say more about feed-forward vs. recurrent nets later in the course

Your questions from the quiz

- Examples of how you would use perceptrons (or other binary classifiers)
  - binary classification tasks, e.g. "Does this patient have disease X?"
  - multi-way classification tasks, e.g. handwritten digit classification: train 10 perceptrons, each one classifies one digit class vs. all the others

Your questions from the quiz

- Examples of how you would use perceptrons (or other binary classifiers)
  - binary classification tasks, e.g. "Does this patient have disease X?"
  - multi-way classification tasks, e.g. handwritten digit classification: train 10 perceptrons, each one classifies one digit class vs. all the others
- How does the threshold relate to the bias?

$$\mathbf{w}^T\mathbf{x} + b > 0 \Longleftrightarrow \mathbf{w}^T\mathbf{x} > -b$$

Question 1: Input space vs. weight space

Understanding weight space and input space for binary linear classification.

- Draw the following points in input space.
  - $(x_1 = 1, x_2 = 1)$ with target 1
  - $(x_1 = -1, x_2 = 1)$ with target 1
  - $(x_1 = 0, x_2 = 2)$ with target 0
- Assume our linear classifier has a threshold at zero and **no bias term**.
  - Write down the inequalities that must be satisfied in order to classify each point correctly.
  - Represent these inequalities in weight space. Indicate “good” and “bad” regions for each.
  - Based on this figure, is this dataset linearly separable?

Question 1: Input space vs. weight space

Understanding weight space and input space for binary linear classification.

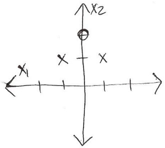
- Draw the following points in input space.
  - $(x_1 = 1, x_2 = 1)$ with target 1
  - $(x_1 = -1, x_2 = 1)$ with target 1
  - $(x_1 = 0, x_2 = 2)$ with target 0
- Assume our linear classifier has a threshold at zero and **no bias term**.
  - Write down the inequalities that must be satisfied in order to classify each point correctly.
  - Represent these inequalities in weight space. Indicate "good" and "bad" regions for each.
  - Based on this figure, is this dataset linearly separable?
- Now suppose the classifier has a bias term whose value is fixed at 1. Again, write down the inequalities and represent them in weight space. Now is it linearly separable?

Question 1: Input space vs. weight space

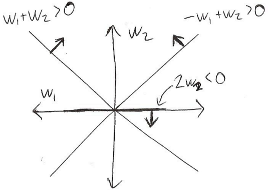
Understanding weight space and input space for binary linear classification.

- Draw the following points in input space.
  - $(x_1 = 1, x_2 = 1)$ with target 1
  - $(x_1 = -1, x_2 = 1)$ with target 1
  - $(x_1 = 0, x_2 = 2)$ with target 0
- Assume our linear classifier has a threshold at zero and **no bias term**.
  - Write down the inequalities that must be satisfied in order to classify each point correctly.
  - Represent these inequalities in weight space. Indicate "good" and "bad" regions for each.
  - Based on this figure, is this dataset linearly separable?
- Now suppose the classifier has a bias term whose value is fixed at 1. Again, write down the inequalities and represent them in weight space. Now is it linearly separable?

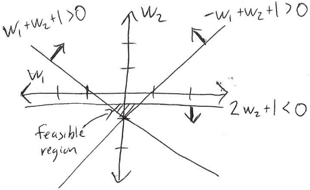
Question 1: Input space vs. weight space



(a) Input space



(b) Weight space (no bias)



(c) Weight space (bias = 1)

Question 1: Input space vs. weight space

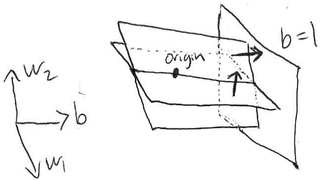
Summary: representations in input space and weight space

		Input space	Weight space
(no bias)	**Data points**	points	half-spaces through origin
	Classifiers	half-spaces through origin	points
(with bias)	**Data points**	points	half-spaces through origin
	Classifiers	general half-spaces	points

Note: with N inputs, the weight space is $N+1$ dimensional if we include the bias. We fixed the bias so that we could visualize the 2-D slice of the weight space where $b = 1$. But in practice, we almost always use a bias parameter and almost always learn it.

Question 1: Input space vs. weight space

Here's a lame attempt to visualize the relationship between the 3-D weight space and the $b = 1$ slice we visualized earlier. Note that the constraints are half-spaces whose decision boundary passes through the origin, and the decision boundaries are planes. (The constraints in this figure aren't meant to correspond to the ones we used in this question.)



Question 2: Feasible region

Geometry of the solutions: Why does the feasible region look like a cone (assuming no bias parameter)?

Mathematically, a set S is a cone if:

- The line segment connecting any two points in S is contained in S. (This requirement is called *convexity*)
- The ray from 0 through any $x \in S$ is contained in S

Why does this hold for the feasible region?

Question 2: Feasible region

Geometry of the solutions: Why does the feasible region look like a cone (assuming no bias parameter)?

Mathematically, a set S is a cone if:

- The line segment connecting any two points in S is contained in S. (This requirement is called *convexity*)
- The ray from 0 through any $x \in S$ is contained in S

Why does this hold for the feasible region?

- Take any two points $\mathbf{w}_1, \mathbf{w}_2$ in the feasible region.
- Represent the line segment between them as $\lambda \mathbf{w}_1 + (1 - \lambda)\mathbf{w}_2$, $0 \leq \lambda \leq 1$.
- Are points on this line feasible ?
- Represent the ray from the origin to $\mathbf{w}_1$ as $\alpha \mathbf{w}_1$, $\alpha > 0$.
- Are points on this ray feasible ?

Question 2: Feasible region

Geometry of the solutions: Why does the feasible region look like a cone (assuming no bias parameter)?

Mathematically, a set S is a cone if:

- The line segment connecting any two points in S is contained in S.
  - Suppose $\mathbf{w}_1^T \mathbf{x} > 0$ and $\mathbf{w}_2^T \mathbf{x} > 0$. Then for $0 \leq \lambda \leq 1$,

$$(\lambda \mathbf{w}_1 + (1 - \lambda)\mathbf{w}_2)^T \mathbf{x} = \lambda \mathbf{w}_1^T \mathbf{x} + (1 - \lambda)\mathbf{w}_2^T \mathbf{x} > 0.$$

  Therefore, if $\mathbf{w}_1$ and $\mathbf{w}_2$ classify all examples correctly, then so does $\lambda \mathbf{w}_1 + (1 - \lambda)\mathbf{w}_2$.

Question 2: Feasible region

Geometry of the solutions: Why does the feasible region look like a cone (assuming no bias parameter)?

Mathematically, a set S is a cone if:

- The line segment connecting any two points in S is contained in S.
  - Suppose $\mathbf{w}_1^T \mathbf{x} > 0$ and $\mathbf{w}_2^T \mathbf{x} > 0$. Then for $0 \leq \lambda \leq 1$,

    $$(\lambda \mathbf{w}_1 + (1 - \lambda)\mathbf{w}_2)^T \mathbf{x} = \lambda \mathbf{w}_1^T \mathbf{x} + (1 - \lambda)\mathbf{w}_2^T \mathbf{x} > 0.$$

    Therefore, if $\mathbf{w}_1$ and $\mathbf{w}_2$ classify all examples correctly, then so does $\lambda \mathbf{w}_1 + (1 - \lambda)\mathbf{w}_2$.
- The ray from 0 through any $x \in S$ is contained in S
  - Rescaling by $\alpha > 0$ doesn't change the predictions, since $\alpha \mathbf{w}^T \mathbf{x} > 0$ if and only if $\mathbf{w}^T \mathbf{x} > 0$.