CS 760: Machine Learning
Regression: II

Fred Sala

University of Wisconsin-Madison

Sept. 30, 2021

Logistics

- **Announcements** :
  - HW 2 due tonight at midnight
  - Project info released. **Proposal due Oct. 14**
- Class roadmap:

Thursday Sept. 30	Regression II
Tuesday, Oct. 5	Naive Bayes
Thursday, Oct. 7	Neural Networks I
Tuesday, Oct. 12	Neural Networks II
Thursday, Oct. 14	Neural Networks III

Supervised Learning

Outline

- **Logistic Regression**
  - Maximum likelihood estimation, setup, comparisons
- **Logistic Regression: Multiclass**
  - Extending to multiclass, softmax, cross-entropy
- **Gradient Descent & SGD**
  - Convergence proof for GD, introduction to SGD

Outline

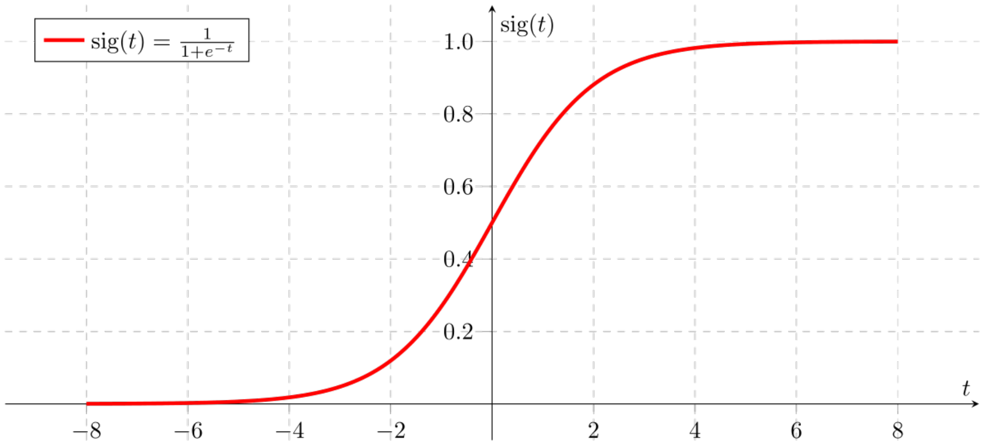
- **Logistic Regression**
  - Maximum likelihood estimation, setup, comparisons
- **Logistic Regression: Multiclass**
  - Extending to multiclass, softmax, cross-entropy
- **Gradient Descent & SGD**
  - Convergence proof for GD, introduction to SGD

Linear Classification: Attempt 2

- Let's think probabilistically. Learn $P_\theta(y|x)$ instead

- How?
  - Specify the conditional distribution $P_\theta(y|x)$
  - Use **MLE** to derive a loss
  - Run gradient descent (or related optimization algorithm)

  - Leads to logistic regression



Likelihood Function

- Captures the probability of seeing some data as a function of model parameters:

$$\mathcal{L}(\theta; X) = P_\theta(X)$$

  - If data is iid, we have $\mathcal{L}(\theta; X) = \prod_j p_\theta(x_j)$

- Often more convenient to work with the log likelihood
  - Log is a monotonic + strictly increasing function

Maximum Likelihood

- For some set of data, find the parameters that maximize the likelihood / log-likelihood

$$\hat{\theta} = \arg\max_{\theta} \mathcal{L}(\theta; X)$$

- Example: suppose we have n samples from a Bernoulli distribution

$$P_\theta(X = x) = \begin{cases} \theta & x = 1 \\ 1 - \theta & x = 0 \end{cases}$$

Then,

$$\mathcal{L}(\theta; X) = \prod_{i=1}^{n} P(X = x_i) = \theta^k (1-\theta)^{n-k}$$

Maximum Likelihood: Example

- Want to maximize likelihood w.r.t. Θ

$$\mathcal{L}(\theta; X) = \prod_{i=1}^{n} P(X = x_i) = \theta^k (1-\theta)^{n-k}$$

- Differentiate (use product rule) and set to 0. Get

$$\theta^{h-1}(1-\theta)^{n-h-1}(h - n\theta) = 0$$

- So: ML estimate is $\hat{\theta} = \frac{h}{n}$

h = |{x_i | x_i = 1}|

ML: Conditional Likelihood

- Similar idea, but now using conditional probabilities:

$$\mathcal{L}(\theta; Y, X) = p_\theta(Y|X)$$

  - If data is iid, we have

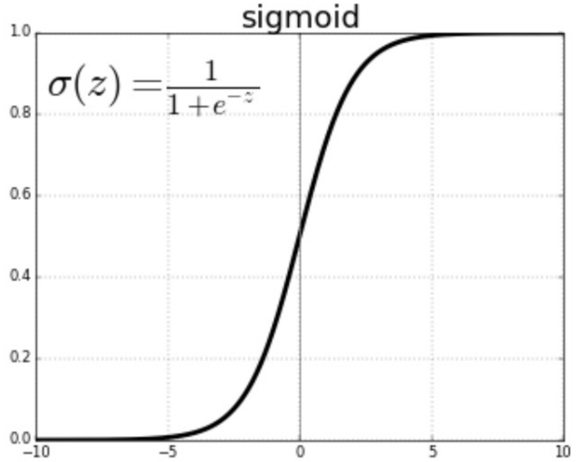
$$\mathcal{L}(\theta; Y, X) = \prod_j p_\theta(y_j|x_j)$$

  - Now we can apply this to linear classification: yields **logistics regression.**

Logistic Regression: Conditional Distribution

- Notation: $$\sigma(z) = \frac{1}{1 + \exp(-z)} = \frac{\exp(z)}{1 + \exp(z)}$$

Sigmoid



- **Conditional Distribution**:

$$P_\theta(y = 1|x) = \sigma(\theta^T x) = \frac{1}{1 + \exp(-\theta^T x)}$$

Logistic Regression: Loss

- Conditional MLE:

$$\log \text{likelihood}(w|x^{(i)}, y^{(i)}) = \log P_\theta(y^{(i)}|x^{(i)})$$

- So:

$$\min_\theta \ell(f_\theta) = \min_\theta -\frac{1}{n}\sum_{i=1}^{n} \log P_\theta(y^{(i)}|x^{(i)})$$

Or,

$$\min_\theta -\frac{1}{n}\sum_{y^{(i)}=1} \log \sigma(\theta^T x^{(i)}) - \frac{1}{n}\sum_{y^{(i)}=0} \log(1 - \sigma(\theta^T x^{(i)}))$$

Logistic Regression: Sigmoid Properties

- **Bounded**:

$$\sigma(z) = \frac{1}{1+\exp(-z)} \in (0,1)$$

- **Symmetric**:

$$1-\sigma(z) = \frac{\exp(-z)}{1+\exp(-z)} = \frac{1}{\exp(z)+1} = \sigma(-z)$$

- **Gradient**:

$$\sigma'(z) = \frac{\exp(-z)}{(1+\exp(-z))^2} = \sigma(z)(1-(\sigma(z))$$

Logistic regression: Summary

- **Logistic regression = sigmoid conditional distribution + MLE**

- More precisely:
  - Give training data iid from some distribution D,
  - **Train**:

$$\min_{\theta} \ell(f_\theta) = \min_{\theta} -\frac{1}{n}\sum_{i=1}^{n} \log P_\theta(y^{(i)}|x^{(i)})$$

  - **Test**: output label probabilities

$$P_\theta(y=1|x) = \sigma(\theta^T x) = \frac{1}{1+\exp(-\theta^T x)}$$

Logistic Regression: Comparisons

- Recall the first attempt:

$$\ell(f_\theta) = \frac{1}{m}\sum_{i=1}^{m} 1\{\text{step}(f_\theta(x^{(i)}) \neq y^{(i)})$$

- **Difficult to optimize!!**
- Another way: run least squares, ignore that y is 0 or 1:

$$\ell(f_\theta) = \frac{1}{n}\sum_{j=1}^{n}(f_\theta(x^{(j)}) - y^{(j)})^2$$

Logistic Regression: Comparisons

- Downside: not robust to “outliers”

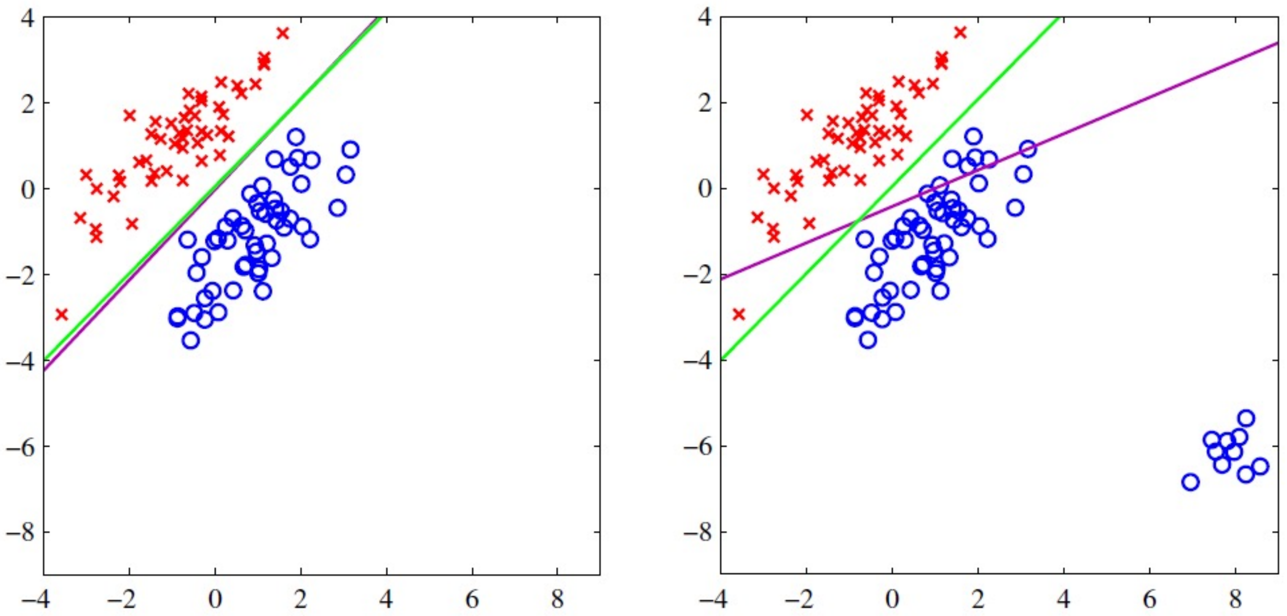


Figure 4.4 The left plot shows data from two classes, denoted by red crosses and blue circles, together with the decision boundary found by least squares (magenta curve) and also by the logistic regression model (green curve), which is discussed later in Section 4.3.2. The right-hand plot shows the corresponding results obtained when extra data points are added at the bottom left of the diagram, showing that least squares is highly sensitive to outliers, unlike logistic regression.

Figure: *Pattern Recognition and Machine Learning*, Bishop

Break & Quiz

Outline

- **Logistic Regression**
  - Maximum likelihood estimation, setup, comparisons
- **Logistic Regression: Multiclass**
  - Extending to multiclass, softmax, cross-entropy
- **Gradient Descent & SGD**
  - Convergence proof for GD, introduction to SGD

Logistic Regression: Beyond Binary

- We started with this conditional distribution:

$$P_\theta(y = 1|x) = \sigma(\theta^T x) = \frac{1}{1 + \exp(-\theta^T x)}$$

- Now let's try to extend it.
  - Can no longer just use one $\theta^T x$
  - But we can try multiple…

Logistic Regression: Beyond Binary

- Let's set, for y in 1,2,…,k

$$P_\theta(y = i|x) = \frac{\exp((\theta^i)^T x)}{\sum_{j=1}^k \exp((\theta^j)^T x)}$$

- Note: we have several weight vectors now (1 per class).
- To train, same as before (just more weight vectors).

$$\min_\theta -\frac{1}{n}\sum_{i=1}^n \log P_\theta(y^{(i)}|x^{(i)})$$

Cross-Entropy Loss

- Let's define $q^{(i)}$ as the one-hot vector for the ith datapoint.
- Next, let's let $p^{(i)} = P_\theta(y|x^{(i)})$ be our prediction

Note: only 1 term non-zero.

- Our loss terms can be written

$$-\log p(y^{(i)}|x^{(i)}) = -\sum_{j=1}^{k} \underbrace{q_j^{(i)} \log p(y = j|x^{(i)})}_{\text{Should look familiar...}}$$

- This is the "cross-entropy" $H(q^{(i)}, p^{(i)})$

Cross-Entropy Loss

- This is the "cross-entropy"

$$H(q^{(i)}, p^{(i)}) = \mathbb{E}_{q^{(i)}}[\log p^{(i)}]$$

- What are we doing when we minimize the cross-entropy?
- Recall KL divergence,

$$D(q^{(i)}||p^{(i)}) = \underbrace{\mathbb{E}_{q^{(i)}}[\log p^{(i)}]}_{\text{Cross-entropy}} - \underbrace{\mathbb{E}_{q^{(i)}}[\log q^{(i)}]}_{\substack{\text{Entropy } H(q^{(i)}) \\ \text{(fixed)}}}$$

- Matching distributions!

Softmax

- We wrote

$$P_\theta(y = i|x) = \frac{\exp((\theta^i)^T x)}{\sum_{j=1}^{k} \exp((\theta^j)^T x)}$$

- This operation is called softmax.
  - Converts a vector into a probability vector (note normalization).
  - If one component in the vector **a** is **large**, softmax(**a**) is close to one-hot vector

Break & Quiz

Outline

- **Logistic Regression**
  - Maximum likelihood estimation, setup, comparisons
- **Logistic Regression: Multiclass**
  - Extending to multiclass, softmax, cross-entropy
- **Gradient Descent & SGD**
  - Convergence proof for GD, introduction to SGD

Gradient Descent: Convergence Proof p. 1

- We'll use our **two ingredients**. Let's start with a Taylor expansion:

$$f(y) = f(x) + \nabla f(x)^T (y - x) + 1/2 (y - x)^T \nabla^2 f(z) (y - x)$$

- Next, our gradient Lipschitz condition means $\nabla^2 f(x) \prec LI$

$$\boxed{\implies \; f(y) \leq \underbrace{f(x) + \nabla f(x)^T (y - x)}_{\text{Linear Approximation}} + \underbrace{1/2 L \|y - x\|^2}_{\text{Remainder: at most a quadratic}}}$$

Gradient Descent: Convergence Proof p. 2

- Let's plug in our GD relationship $y \leftarrow x_{t+1} = x_t - \alpha \nabla f(x)$

$$\boxed{\implies f(y) \leq f(x) + \nabla f(x)^T (y - x) + 1/2L\|y - x\|^2}$$

- Start with some algebra

$$f(x_{t+1}) \leq f(x_t) + \nabla f(x_t)^T (x_{t+1} - x_t) + 1/2L\|x_{t+1} - x_t\|_2^2$$

$$= f(x_t) - \nabla f(x_t)^T \alpha \nabla f(x_t) + 1/2L\|\alpha \nabla f(x_t)\|_2^2$$

$$= f(x_t) - \alpha\|\nabla f(x_t)\|_2^2 + 1/2L\alpha^2\|\nabla f(x_t)\|_2^2$$

$$\boxed{= f(x_t) - \alpha(1 - 1/2L\alpha)\|\nabla f(x_t)\|_2^2}$$

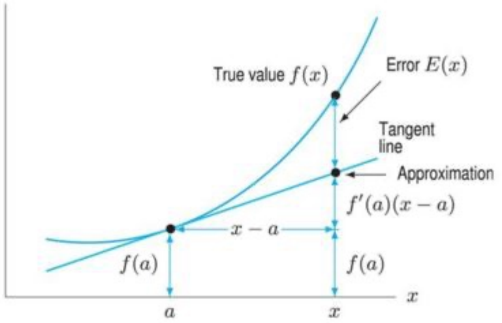
Gradient Descent: Convergence Proof p. 3

- So, we now have

$$\boxed{f(x_{t+1}) \leq f(x_t) - 1/2\alpha \underbrace{\|\nabla f(x_t)\|_2^2}_{\text{Positive except at minimum (where it's 0)}}}$$

- Promising! Our estimates are getting better.
  - Still need how big these gradient magnitudes are

Gradient Descent: Convergence Proof p. 4



- Haven't used convexity yet, so let's:

$$\boxed{f(x_t) \leq f(x^*) + \nabla f(x)^T (x_t - x^*)}$$

- Combine with $\boxed{f(x_{t+1}) \leq f(x_t) - 1/2\alpha \|\nabla f(x_t)\|_2^2}$

$$f(x_{t+1}) \leq f(x^*) + \nabla f(x_t)^T (x_t - x^*) - \alpha/2 \|\nabla f(x_t)\|_2^2$$

$$f(x_{t+1}) - f(x^*) \leq \frac{1}{2\alpha}(2\alpha \nabla f(x_t)^T (x_t - x^*) - \alpha^2 \|\nabla f(x_t)\|_2^2)$$

$$\boxed{f(x_{t+1}) - f(x^*) \leq \frac{1}{2\alpha}(\|x_t - x^*\|_2^2 - \|x_t - \alpha \nabla f(x_t) - x^*\|_2^2)}$$

Gradient Descent: Convergence Proof p. 5

- Now, simplify

$$\boxed{f(x_{t+1}) - f(x^*) \leq \frac{1}{2\alpha}(\|x_t - x^*\|_2^2 - \|\underbrace{x_t - \alpha\nabla f(x_t)}_{\text{This part is just } x_{t+1}} - x^*\|_2^2)}$$

$$\boxed{f(x_{t+1}) - f(x^*) \leq \frac{1}{2\alpha}(\|x_t - x^*\|_2^2 - \|x_{t+1} - x^*\|_2^2)}$$

Gradient Descent: Convergence Proof p. 6

- So, we have something familiar…

$$\boxed{f(x_{t+1}) - f(x^*) \leq \frac{1}{2\alpha}\left(\underbrace{\|x_t - x^*\|_2^2 - \|x_{t+1} - x^*\|_2^2}_{\text{Can telescope!}}\right)}$$

$$\sum_{t=0}^{T-1} f(x_{t+1}) - f(x^*) \leq \sum_{t=0}^{T-1} \frac{1}{2\alpha}\left(\|x_t - x^*\|_2^2 - \|x_{t+1} - x^*\|_2^2\right)$$

$$\boxed{\sum_{t=0}^{T-1} f(x_{t+1}) - f(x^*) \leq \frac{1}{2\alpha}\left(\|x_0 - x^*\|_2^2 - \|x_T - x^*\|_2^2\right)}$$

Gradient Descent: Convergence Proof p. 7

- Now we have

$$\sum_{t=0}^{T-1} f(x_{t+1}) - f(x^*) \leq \frac{1}{2\alpha}(\|x_0 - x^*\|_2^2 - \|x_T - x^*\|_2^2)$$

- Can ignore the rightmost term (we're just making the RHS same or bigger)

$$\sum_{t=0}^{T-1} \underbrace{f(x_{t+1}) - f(x^*)}_{\text{Value gap for all steps}} \leq \frac{1}{2\alpha}\underbrace{(\|x_0 - x^*\|_2^2)}_{\text{Initial guess gap to minimizer}}$$

Gradient Descent: Convergence Proof p. 7

- Continue,

$$\boxed{\sum_{t=0}^{T-1} f(x_{t+1}) - f(x^*) \leq \frac{1}{2\alpha}(\|x_0 - x^*\|_2^2)}$$

- But, recall that each iterate has a smaller value, ie,

$$\boxed{f(x_{t+1}) \leq f(x_t) - 1/2\alpha\|\nabla f(x_t)\|_2^2}$$

- So,

$$\sum_{t=0}^{T-1} f(x_T) \leq \sum_{t=0}^{T-1} f(x_{t+1})$$

Gradient Descent: Convergence Proof p. 8

- Almost there! We have $$\sum_{t=0}^{T-1} f(x_T) \leq \sum_{t=0}^{T-1} f(x_{t+1})$$
- Divide by T,

$$f(x_T) - f(x^*) \leq \frac{1}{T} \sum_{i=0}^{T-1} f(x_t) - f(x^*)$$

- Combine with $$\sum_{t=0}^{T-1} f(x_{t+1}) - f(x^*) \leq \frac{1}{2\alpha}(\|x_0 - x^*\|_2^2)$$

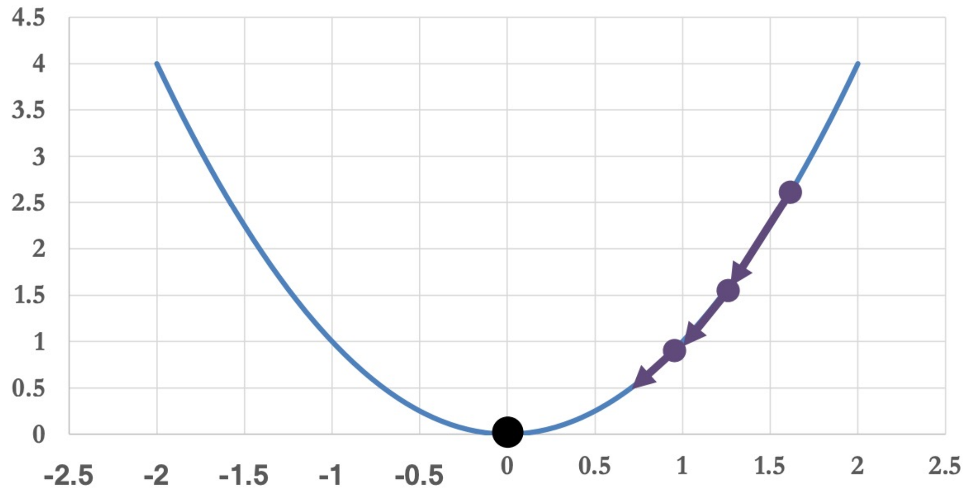
$$\boxed{\implies f(x_T) - f(x^*) \leq \frac{\|x_0 - x^*\|_2^2}{2T\alpha}}$$

Done!

Gradient Descent: Convergence Proof Recap

- **Note**: used all conditions in one or more places in the proof.
  - If you don't use an assumption, either your result is stronger than you thought or (more likely) you are making a mistake

- Proof credit: Ryan Tibshirani.
- Other assumptions that lead to varying proofs/rates:
  - **Strong convexity**
  - **Non-convexity**
  - **Non-differentiability**



GD: Downside

- Why would we use anything but GD?
- Let's go back to ERM. $\arg\min_{h \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} \ell(h(x^{(i)}, y^{(i)})$
- For GD, need to compute $\nabla \ell(h(x^{(i)}, y^{(i)})$
  - Each step: n gradient computations
  - ImageNet: 10^6 samples… so for 100 iterations, **10^8 gradients**

Solution: **Stochastic** Gradient Descent

- Simple modification to GD.
- Let's use some notation: ERM:

$$\arg\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} \ell(f(\theta; x^{(i)}), y^{(i)})$$

Note: this is what we're optimizing over! x's are fixed samples.

- GD: $$\theta_{t+1} = \theta_t - \frac{\alpha}{n} \sum_{i=1}^{n} \nabla \ell(f(\theta_t; x^{(i)}), y^{(i)})$$

Solution: **Stochastic** Gradient Descent

- Simple modification to GD:

$$\theta_{t+1} = \theta_t - \frac{\alpha}{n} \sum_{i=1}^{n} \nabla \ell(f(\theta_t; x^{(i)}), y^{(i)})$$

- SGD: $$\theta_{t+1} = \theta_t - \alpha \nabla \ell(f(\theta_t; x^{(a)}), y^{(a)})$$

  - Here, a is selected uniformly from 1,…,n (“**stochastic**” bit)
  - Note: **no sum**!
  - In expectation, same as GD.

Thanks Everyone!

Some of the slides in these lectures have been adapted/borrowed from materials developed by Mark Craven, David Page, Jude Shavlik, Tom Mitchell, Nina Balcan, Elad Hazan, Tom Dietterich, Pedro Domingos, Jerry Zhu, Yingyu Liang, Volodymyr Kuleshov