

Gradient and Stochastic Gradient Descent

David Rosenberg

New York University

October 29, 2016

Linear Least Squares Regression

Setup

- Input space $\mathcal{X} = \mathbf{R}^d$
- Output space $\mathcal{Y} = \mathbf{R}$
- Action space $\mathcal{Y} = \mathbf{R}$
- Loss: $\ell(\hat{y}, y) = \frac{1}{2} (y - \hat{y})^2$
- **Hypothesis space:** $\mathcal{F} = \{f : \mathbf{R}^d \rightarrow \mathbf{R} \mid f(x) = w^T x, w \in \mathbf{R}^d\}$

Linear Least Squares Regression

Setup

- Input space $\mathcal{X} = \mathbf{R}^d$
 - Output space $\mathcal{Y} = \mathbf{R}$
 - Action space $\mathcal{Y} = \mathbf{R}$
 - Loss: $\ell(\hat{y}, y) = \frac{1}{2} (y - \hat{y})^2$
 - **Hypothesis space:** $\mathcal{F} = \{f : \mathbf{R}^d \rightarrow \mathbf{R} \mid f(x) = w^T x, w \in \mathbf{R}^d\}$
-
- Given data set $\mathcal{D}_n = \{(x_1, y_1), \dots, (x_n, y_n)\}$,
 - Let's find the ERM $\hat{f} \in \mathcal{F}$.

Linear Least Squares Regression

Objective Function: Empirical Risk

The function we want to minimize is the empirical risk:

$$\hat{R}_n(w) = \frac{1}{n} \sum_{i=1}^n (w^T x_i - y_i)^2,$$

where $w \in \mathbf{R}^d$ parameterizes the hypothesis space \mathcal{F} .

Unconstrained Optimization

Setting

Objective function $f : \mathbf{R}^d \rightarrow \mathbf{R}$ is *differentiable*.

Want to find

$$x^* = \arg \min_{x \in \mathbf{R}^d} f(x)$$

The Gradient

Let $f : \mathbf{R}^d \rightarrow \mathbf{R}$ be differentiable at $x_0 \in \mathbf{R}^d$.

Definition

The **gradient** of f at the point x_0 , denoted $\nabla_x f(x_0)$, is the direction to move in for the **fastest increase** in $f(x)$, when starting from x_0 .

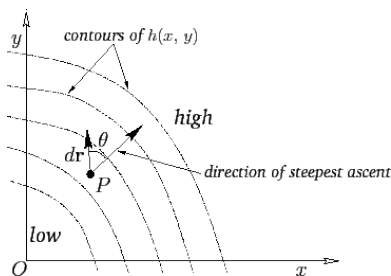


Figure A.111 from Newtonian Dynamics, by Richard Fitzpatrick.

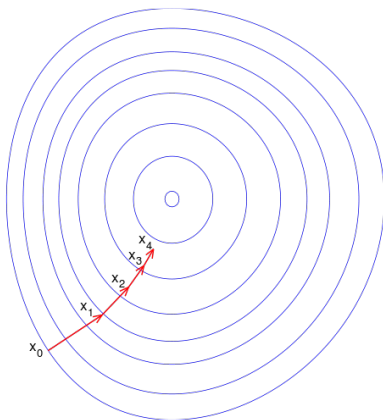
Gradient Descent

Gradient Descent

- Initialize $x = 0$
- repeat
 - $x \leftarrow x - \underbrace{\eta}_{\text{step size}} \nabla f(x)$
- until stopping criterion satisfied

Gradient Descent Path

Gradient Descent for a nice (convex) function



Gradient Descent - Details

Step Size

- Empirically $\eta = 0.1$ often works well
- **Better:** Optimize at every step (e.g. backtracking line search)

Gradient Descent - Details

Step Size

- Empirically $\eta = 0.1$ often works well
- **Better:** Optimize at every step (e.g. backtracking line search)

Stopping Rule

- Could use a maximum number of steps (e.g. 100)
- Wait until $\|\nabla f(x)\| \leq \varepsilon$.

Gradient Descent - Details

Step Size

- Empirically $\eta = 0.1$ often works well
- **Better:** Optimize at every step (e.g. backtracking line search)

Stopping Rule

- Could use a maximum number of steps (e.g. 100)
- Wait until $\|\nabla f(x)\| \leq \varepsilon$.
- Wait until decreases in $f(x)$ become very slow.

Gradient Descent - Details

Step Size

- Empirically $\eta = 0.1$ often works well
- **Better:** Optimize at every step (e.g. backtracking line search)

Stopping Rule

- Could use a maximum number of steps (e.g. 100)
- Wait until $\|\nabla f(x)\| \leq \varepsilon$.
- Wait until decreases in $f(x)$ become very slow.
- Test performance on holdout data (in learning setting)

Gradient Descent for Linear Regression

Gradient of Objective Function:

The gradient of the objective is

$$\nabla_w \hat{R}_n(w) = \nabla_w \left[\frac{1}{n} \sum_{i=1}^n (w^T x_i - y_i)^2 \right]$$

Gradient Descent for Linear Regression

Gradient of Objective Function:

The gradient of the objective is

$$\begin{aligned}\nabla_w \hat{R}_n(w) &= \nabla_w \left[\frac{1}{n} \sum_{i=1}^n (w^T x_i - y_i)^2 \right] \\ &= \frac{2}{n} \sum_{i=1}^n \underbrace{(w^T x_i - y_i)}_{i\text{th residual}} x_i\end{aligned}$$

Gradient Descent: Does it scale?

- At every iteration, we compute the gradient at current w :

$$\nabla_w \hat{R}_n(w) = \frac{2}{n} \sum_{i=1}^n \underbrace{(w^T x_i - y_i)}_{i\text{th residual}} x_i$$

Gradient Descent: Does it scale?

- At every iteration, we compute the gradient at current w :

$$\nabla_w \hat{R}_n(w) = \frac{2}{n} \sum_{i=1}^n \underbrace{(w^T x_i - y_i)}_{i\text{th residual}} x_i$$

- We have to touch all n training points to take a single step. [$O(n)$]

Gradient Descent: Does it scale?

- At every iteration, we compute the gradient at current w :

$$\nabla_w \hat{R}_n(w) = \frac{2}{n} \sum_{i=1}^n \underbrace{(w^T x_i - y_i)}_{i\text{th residual}} x_i$$

- We have to touch all n training points to take a single step. $[O(n)]$
 - Will this scale to “big data”?

Gradient Descent: Does it scale?

- At every iteration, we compute the gradient at current w :

$$\nabla_w \hat{R}_n(w) = \frac{2}{n} \sum_{i=1}^n \underbrace{(w^T x_i - y_i)}_{i\text{th residual}} x_i$$

- We have to touch all n training points to take a single step. $[O(n)]$
 - Will this scale to “big data”?
- Can we make progress without looking at all the data?

Gradient Descent on the Risk

- 1 Real goal is to minimize the risk (expected loss)

$$R(f) = \mathbb{E}[\ell(f(X), Y)]$$

over a hypothesis space \mathcal{F} .

Gradient Descent on the Risk

- 1 Real goal is to minimize the risk (expected loss)

$$R(f) = \mathbb{E}[\ell(f(X), Y)]$$

over a hypothesis space \mathcal{F} .

- 2 Say hypothesis space \mathcal{F} is parameterized by $w \in \mathbf{R}^d$.

Gradient Descent on the Risk

- 1 Real goal is to minimize the risk (expected loss)

$$R(f) = \mathbb{E}[\ell(f(X), Y)]$$

over a hypothesis space \mathcal{F} .

- 2 Say hypothesis space \mathcal{F} is parameterized by $w \in \mathbb{R}^d$.
- 3 Can we do anything with

$$\nabla_w \mathbb{E}[\ell(f(X), Y)]?$$

Gradient Descent on the Risk

- We have

$$\text{Gradient}(\text{Risk}) = \nabla_w \mathbb{E}[\ell(f(X), Y)]$$

Gradient Descent on the Risk

- We have

$$\text{Gradient}(\text{Risk}) = \nabla_w \mathbb{E}[\ell(f(X), Y)]$$

- Switching ∇_w and \mathbb{E} we can write the gradient of risk as

$$\text{Gradient}(\text{Risk}) = \mathbb{E}[\nabla_w \ell(f(X), Y)]$$

Gradient Descent on the Risk

- We have

$$\text{Gradient}(\text{Risk}) = \nabla_w \mathbb{E}[\ell(f(X), Y)]$$

- Switching ∇_w and \mathbb{E} we can write the gradient of risk as

$$\text{Gradient}(\text{Risk}) = \mathbb{E}[\nabla_w \ell(f(X), Y)]$$

- Can we approximate this expectation?

Gradient Descent on the Risk

- Let's approximate Gradient(Risk)

$$\nabla_w R(f) = \mathbb{E}[\nabla_w \ell(f(X), Y)]$$

with an average over the data:

$$\widehat{\nabla_w R(f)} = \frac{1}{n} \sum_{i=1}^n [\nabla_w \ell(f_w(x_i), y_i)]$$

Gradient Descent on the Risk

- Let's approximate Gradient(Risk)

$$\nabla_w R(f) = \mathbb{E}[\nabla_w \ell(f(X), Y)]$$

with an average over the data:

$$\widehat{\nabla_w R(f)} = \frac{1}{n} \sum_{i=1}^n [\nabla_w \ell(f_w(x_i), y_i)]$$

Three things to note about $\widehat{\nabla_w R(f)}$ as an estimator of $\nabla_w R(f)$:

Gradient Descent on the Risk

- Let's approximate Gradient(Risk)

$$\nabla_w R(f) = \mathbb{E}[\nabla_w \ell(f(X), Y)]$$

with an average over the data:

$$\widehat{\nabla_w R(f)} = \frac{1}{n} \sum_{i=1}^n [\nabla_w \ell(f_w(x_i), y_i)]$$

Three things to note about $\widehat{\nabla_w R(f)}$ as an estimator of $\nabla_w R(f)$:

- 1 **Unbiased:** $\mathbb{E} \widehat{\nabla_w R(f)} = \nabla_w R(f)$.

Gradient Descent on the Risk

- Let's approximate Gradient(Risk)

$$\nabla_w R(f) = \mathbb{E}[\nabla_w \ell(f(X), Y)]$$

with an average over the data:

$$\widehat{\nabla_w R(f)} = \frac{1}{n} \sum_{i=1}^n [\nabla_w \ell(f_w(x_i), y_i)]$$

Three things to note about $\widehat{\nabla_w R(f)}$ as an estimator of $\nabla_w R(f)$:

- 1 **Unbiased:** $\mathbb{E} \widehat{\nabla_w R(f)} = \nabla_w R(f)$.
- 2 **Consistent:** $\lim_{n \rightarrow \infty} \widehat{\nabla_w R(f)} = \nabla_w R(f)$. (Law of large numbers.)

Gradient Descent on the Risk

- Let's approximate Gradient(Risk)

$$\nabla_w R(f) = \mathbb{E}[\nabla_w \ell(f(X), Y)]$$

with an average over the data:

$$\widehat{\nabla_w R(f)} = \frac{1}{n} \sum_{i=1}^n [\nabla_w \ell(f_w(x_i), y_i)]$$

Three things to note about $\widehat{\nabla_w R(f)}$ as an estimator of $\nabla_w R(f)$:

- 1 **Unbiased:** $\mathbb{E} \widehat{\nabla_w R(f)} = \nabla_w R(f)$.
- 2 **Consistent:** $\lim_{n \rightarrow \infty} \widehat{\nabla_w R(f)} = \nabla_w R(f)$. (Law of large numbers.)
- 3 It's exactly the gradient of the empirical risk $\nabla \hat{R}(f)$.

Gradient Descent on the Risk

- We want $\text{Gradient}(\text{Risk})$
- Estimate it using sample of size n .
 - (Our standard procedure when we see an expectation.)

Gradient Descent on the Risk

- We want $\text{Gradient}(\text{Risk})$
- Estimate it using sample of size n .
 - (Our standard procedure when we see an expectation.)
- Bigger $n \implies$ Better estimate

Gradient Descent on the Risk

- We want $\text{Gradient}(\text{Risk})$
- Estimate it using sample of size n .
 - (Our standard procedure when we see an expectation.)
- Bigger $n \implies$ Better estimate
- Bigger $n \implies$ Touching more data (slower!)

Gradient Descent on the Risk

- We want $\text{Gradient}(\text{Risk})$
- Estimate it using sample of size n .
 - (Our standard procedure when we see an expectation.)
- Bigger $n \implies$ Better estimate
- Bigger $n \implies$ Touching more data (slower!)
- But how big an n do we need?

Gradient Descent on the Risk [approximately]

- Gradient descent takes a bunch of steps whether we use
 - the perfect step direction $\nabla R(w)$,

Gradient Descent on the Risk [approximately]

- Gradient descent takes a bunch of steps whether we use
 - the perfect step direction $\nabla R(w)$,
 - an empirical estimate using all training data $\nabla \hat{R}_n(w)$, or

Gradient Descent on the Risk [approximately]

- Gradient descent takes a bunch of steps whether we use
 - the perfect step direction $\nabla R(w)$,
 - an empirical estimate using all training data $\nabla \hat{R}_n(w)$, or
 - an empirical estimate using a random subset of data $\nabla \hat{R}_m(w)$ ($m \ll n$)

Gradient Descent on the Risk [approximately]

- Gradient descent takes a bunch of steps whether we use
 - the perfect step direction $\nabla R(w)$,
 - an empirical estimate using all training data $\nabla \hat{R}_n(w)$, or
 - an empirical estimate using a random subset of data $\nabla \hat{R}_m(w)$ ($m \ll n$)
- What about $m = 1$?

Gradient Descent on the Risk [approximately]

- Gradient descent takes a bunch of steps whether we use
 - the perfect step direction $\nabla R(w)$,
 - an empirical estimate using all training data $\nabla \hat{R}_n(w)$, or
 - an empirical estimate using a random subset of data $\nabla \hat{R}_m(w)$ ($m \ll n$)
- What about $m = 1$?
- Even with a sample of size 1, the estimate

$$\nabla_w \ell(f_w(x_i), y_i)$$

is still **unbiased for Gradient(Risk)**.

Terminology for Gradient Descent Risk Minimization

- **Gradient descent** or “**batch**” **gradient descent**
 - Use full data set of size n to determine step direction

¹See Yoshua Bengio's “Practical recommendations for gradient-based training of deep architectures” <http://arxiv.org/abs/1206.5533>.

Terminology for Gradient Descent Risk Minimization

- **Gradient descent** or “**batch**” **gradient descent**
 - Use full data set of size n to determine step direction
- **Minibatch gradient descent**
 - Use a random subset of size m to determine step direction

¹See Yoshua Bengio's “Practical recommendations for gradient-based training of deep architectures” <http://arxiv.org/abs/1206.5533>.

Terminology for Gradient Descent Risk Minimization

- **Gradient descent** or “batch” gradient descent
 - Use full data set of size n to determine step direction
- **Minibatch gradient descent**
 - Use a random subset of size m to determine step direction
 - Yoshua Bengio says¹:
 - m is typically between 1 and few hundred
 - $m = 32$ is a good default value
 - With $m \geq 10$ we get computational speedup (per datum touched)
- **Stochastic gradient descent**
 - Minibatch with $m = 1$.
 - Use a single randomly chosen point to determine step direction.

¹See Yoshua Bengio's “Practical recommendations for gradient-based training of deep architectures” <http://arxiv.org/abs/1206.5533>.

Minibatch Gradient Descent

Minibatch Gradient Descent (minibatch size m)

- initialize $w = 0$
- repeat
 - randomly choose m points $\{(x_i, y_i)\}_{i=1}^m \subset \mathcal{D}_n$
 - $w \leftarrow w - \eta \left[\frac{1}{m} \sum_{i=1}^m \nabla_w \ell(f_w(x_i), y_i) \right]$
- until stopping criteria met

Stochastic Gradient Descent (SGD)

Stochastic Gradient Descent

- initialize $w = 0$
- repeat
 - randomly choose training point $(x_i, y_i) \in \mathcal{D}_n$
 - $w \leftarrow w - \eta \underbrace{\nabla_w \ell(f_w(x_i), y_i)}_{\text{Grad(Loss on i'th example)}}$
- until stopping criteria met

Step Size

- Let η_t be the step size at the t 'th step.
- What should first step size be?
- How should η_t 's decrease with each step?

Step Size

- Let η_t be the step size at the t 'th step.
- What should first step size be?
- How should η_t 's decrease with each step?

Robbins-Monro Conditions

Many classical convergence results depend on the following two conditions:

$$\sum_{t=1}^{\infty} \eta_t^2 < \infty \quad \sum_{t=1}^{\infty} \eta_t = \infty$$

Step Size

- Let η_t be the step size at the t 'th step.
- What should the first step size be?
- How should η_t 's decrease with each step?

Robbins-Monro Conditions

Many classical convergence results depend on the following two conditions:

$$\sum_{t=1}^{\infty} \eta_t^2 < \infty \quad \sum_{t=1}^{\infty} \eta_t = \infty$$

- As fast as $\eta_t = O\left(\frac{1}{t}\right)$ would satisfy this... but should be faster than $O\left(\frac{1}{\sqrt{t}}\right)$.
- A useful reference for practical techniques: Leon Bottou's "Tricks":

[http:](http://research.microsoft.com/pubs/192769/tricks-2012.pdf)

[//research.microsoft.com/pubs/192769/tricks-2012.pdf](http://research.microsoft.com/pubs/192769/tricks-2012.pdf)