Effects of Eigenvalues on Convergence

Ruoyu Sun

1/27

Questions for Last Time

- **Q1**: Logistic regression is different from linear regression, although both are convex. In terms of optimization terminology, what are the differences?

- **Q2**: What are the two convergence behavior of logistic regression? The cause?

2/27

Today

- Theory starts to breaks down when it comes to high-dimensional problems...
- Topic today: Effects of eigenvalues on convergence
- After today's course, you will be able to
  - understand the effects of eigenvalues for convergence behavior
  - explain the practical behavior of GD for high-dim linear regression

3/27

Outline

Linear Regression by GD
- Lower Dimension Case
- Higher Dimension Case

4/27

Review of Theories You Learned

- We study convex quadratic problems
  - every stationary point is globally optimal (so converge to global optima)
  - with lower bound (so $f(x^r)$ doesn't diverge)
  - achievable (so x^r doesn't diverge)
- When does GD "converge"?
  - Stepsize choice: $< 2/L$ with $L = \lambda_{\max}$
- Convergence rate?
  - $O(L/\epsilon)$ for convex problems
  - $\kappa \log(1/\epsilon)$ iterations for strongly convex problems

The convergence rate results of $O(\kappa)$ and $O(L/\epsilon)$, is a much simplified explanation for practical behavior.

5/27

Example: Salary Prediction

Kaggle dataset ”Predict NHL Player Salary”:
https://www.kaggle.com/datasets

- about 150 features
- about 600 samples

	A	B	C	D	E	F	G	H	I	J	K
1	Salary	Born	City	Pr/St	Cntry	Nat	Ht	Wt	DftYr	DftRd	Ovrl
2	925000	97-01-30	Sainte-Marie	QC	CAN	CAN	74	190	2015	1	18
3	2250000	93-12-21	Ottawa	ON	CAN	CAN	74	207	2012	1	15
4	8000000	88-04-16	St. Paul	MN	USA	USA	72	218	2006	1	7
5	3500000	92-01-07	Ottawa	ON	CAN	CAN	77	220	2010	1	3
6	1750000	94-03-29	Toronto	ON	CAN	CAN	76	217	2012	1	16
7	1500000	79-05-23	Strathroy	ON	CAN	CAN	70	192	1997	6	156
8	950000	90-11-21	Stockholm		SWE	SWE	71	185	2009	2	53
9	842500	93-07-28	Toronto	ON	CAN	CAN	70	183			
10	1250000	92-06-14	Scarborough	ON	CAN	CAN	72	214	2010	2	42
11	925000	93-04-27	Petawawa	ON	CAN	CAN	68	178	2011	7	201
12	800000	86-07-27	Lively	ON	CAN	CAN	73	195			
13	600000	92-04-09	Ile Bizard	QC	CAN	CAN	74	204	2010	3	89
14	1000000	88-03-17	Brandon	MB	CAN	CAN	72	200	2006	3	66
15	925000	96-06-20	Holland Land	ON	CAN	CAN	73	186	2014	1	4
16	950000	89-04-20	Red Deer	AB	CAN	CAN	72	195	2007	4	112
17	680000	94-01-06	Edmonton	AB	CAN	CAN	74	187	2012	4	105
18	590000	92-04-24	Oakville	ON	CAN	CAN	71	183	2011	7	209
19	650000	91-05-13	St. Catharine	ON	CAN	CAN	74	203	2009	3	70
20	2250000	86-04-28	Ostrava		CZE	CZE	74	235	2004	6	180
21	600000	90-10-08	Faribault	MN	USA	USA	76	210	2009	4	114
22	5000000	94-03-09	Vancouver	BC	CAN	CAN	73	215	2012	1	5

6/27

Use Linear Regression

Data preprocessing:

- Just pick numerical values
- Fill in the missing values by the average
- Normalize each feature (each column of data matrix X)
- Divide income y by 10^6, so typical range is about 0.9 to 10

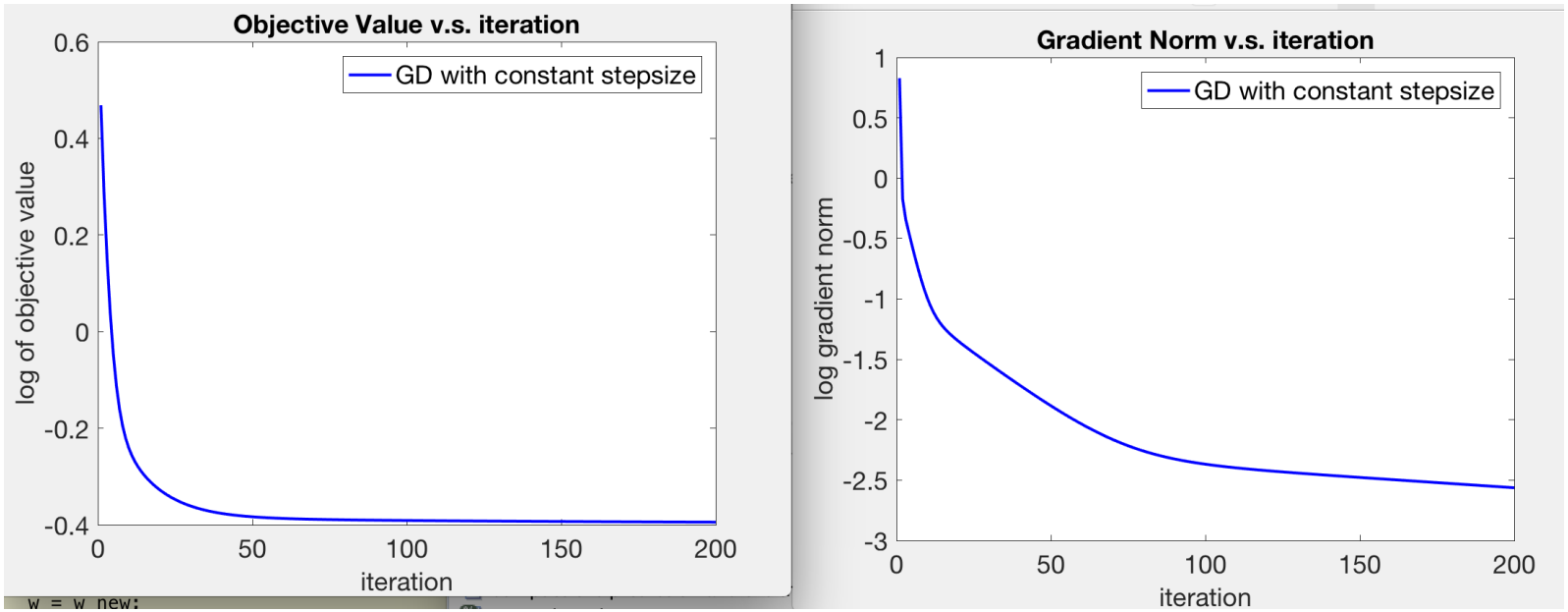
Problem and Algorithm:

- Solve $\min_w f(w) = 0.5\|Xw - y\|^2$;
- Report $f(w)/N$, so it should be $O(1)$
- Use GD with stepsize $1/L$, where $L = \lambda_{\max}(X^T X)$

Initial trial: Use a subset of the data $N = 10, d = 5$

7/27

GD Performance



Observation: Seems like a sublinear rate of convergence?

Compute maximal and minimal eigenvalues of $Q = X^T X$:
0.0139;
3.5742

- Strongly convex!
- However, looks like sublinear rate of convergence

8/27

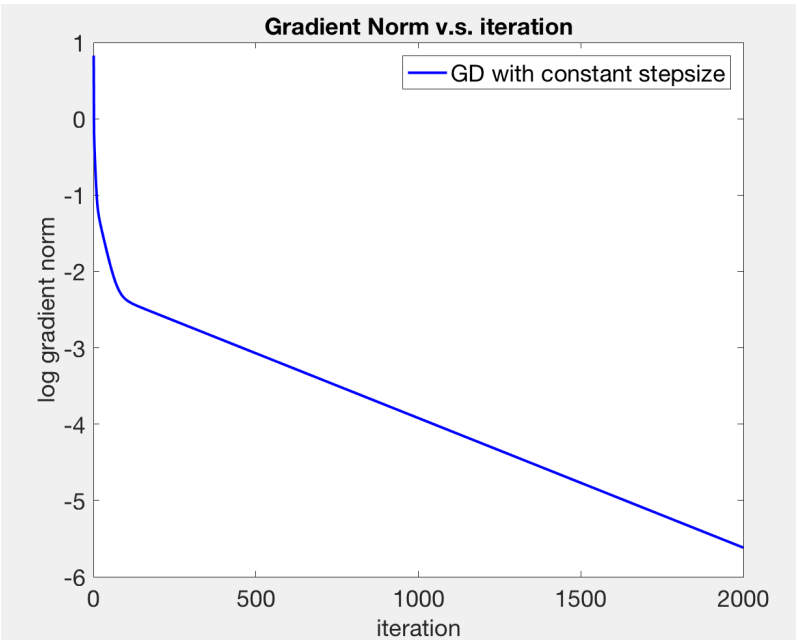
Finally Linear Convergence

Possible explanation: Theory only provides a bound, not a precise characterization?

Yes and no...

First, it will finally converge in linear rate (increase MaxIter):

- Recall: to analyze, always consider checking x^0, α, MaxIter



9/27

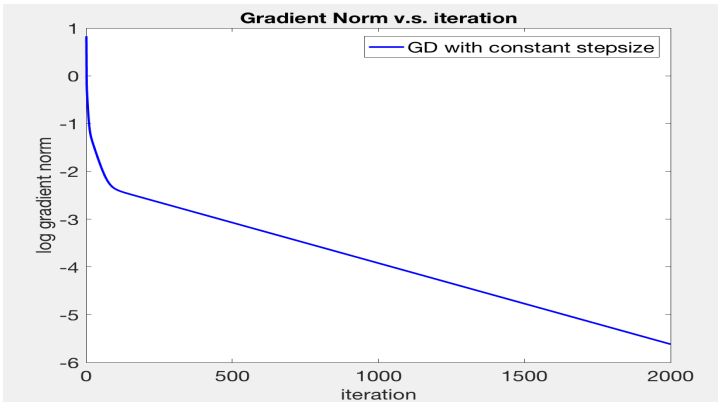
Predict # of iterations by condition number

First, it will finally converge in linear rate, with rate $1 - 1/\kappa$

Condition number: $\kappa = 3.57/0.014 \approx 256$.
Predicted iteration: $\kappa \log \frac{1}{\epsilon}$ (1st row below)

- Error scaled by $1/10$, iteration count increases by $\kappa \log 10 \approx 256 \log 10 \approx 590$ (2nd row below)

error ϵ	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}
predicted r s.t. $\|x^r - x^*\| \leq \epsilon$	590	1180	1770	2160	2750
Predicted iteration difference	590	590	590	590	590
true r s.t. $\|\nabla f(x^r)\| \leq \epsilon$	10	50	500	1090	1680
True iteration difference	10	40	450	590	590



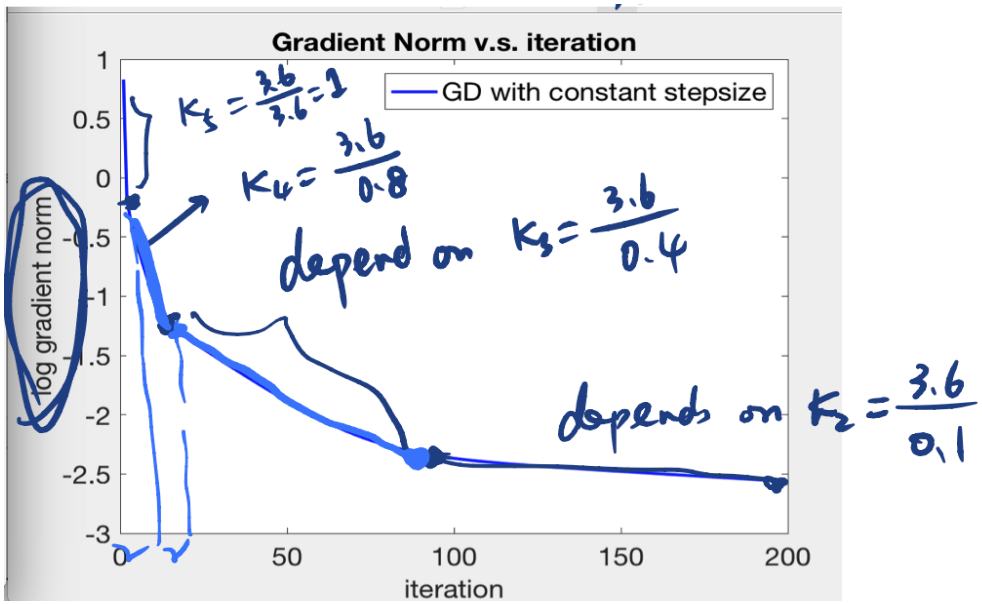
10/27

GD Performance

Second, **at the early stage**, although it seems like sublinear rate , but... **actually, multi-stage linear rate!**

To explain this, we need to compute ALL eigenvalues of $Q = X^T X$: 0.0139; 0.1497; 0.4417; 0.8206; 3.5742

Each segment decreases at rate $1 - \kappa_i$, where $\kappa_i = \lambda_1/\lambda_i, i = 1, 2, 3, 4, 5.$



11/27

Explanation of Multi-stage Behavior (I)

Consider a diagonal matrix $\mathbf{Q} = \text{diag}(1, 0.01, 0.0001)$.

Solve $\min_{\mathbf{x}} 1/2\mathbf{x}^T\mathbf{Q}\mathbf{x}$ by GD with stepsize $\alpha = 1/L$, where $L = 1$.

- $\nabla f(\mathbf{x}) = \mathbf{Q}\mathbf{x} = (\lambda_1 x_1; \lambda_2 x_2; \lambda_3 x_3)$
- $\mathbf{x}^{r+1} = \mathbf{x}^r - \alpha\nabla f(\mathbf{x}) = \mathbf{x}^r - (\lambda_1 x_1^r; \lambda_2 x_2^r; \lambda_3 x_3^r) = \begin{pmatrix} (1-\lambda_1)x_1^r \\ (1-\lambda_2)x_2^r \\ (1-\lambda_3)x_3^r \end{pmatrix}$.

  Thus, $\mathbf{x}^r = ((1-\lambda_1)^r; (1-\lambda_2)^r; (1-\lambda_3)^r). = (0, 0.99^r, 0.9999^r)$.
- Write $2f(\mathbf{x}) = \mathbf{x}^T \begin{pmatrix} 1 & & \\ & 0.01 & \\ & & 0.0001 \end{pmatrix} \mathbf{x} =$

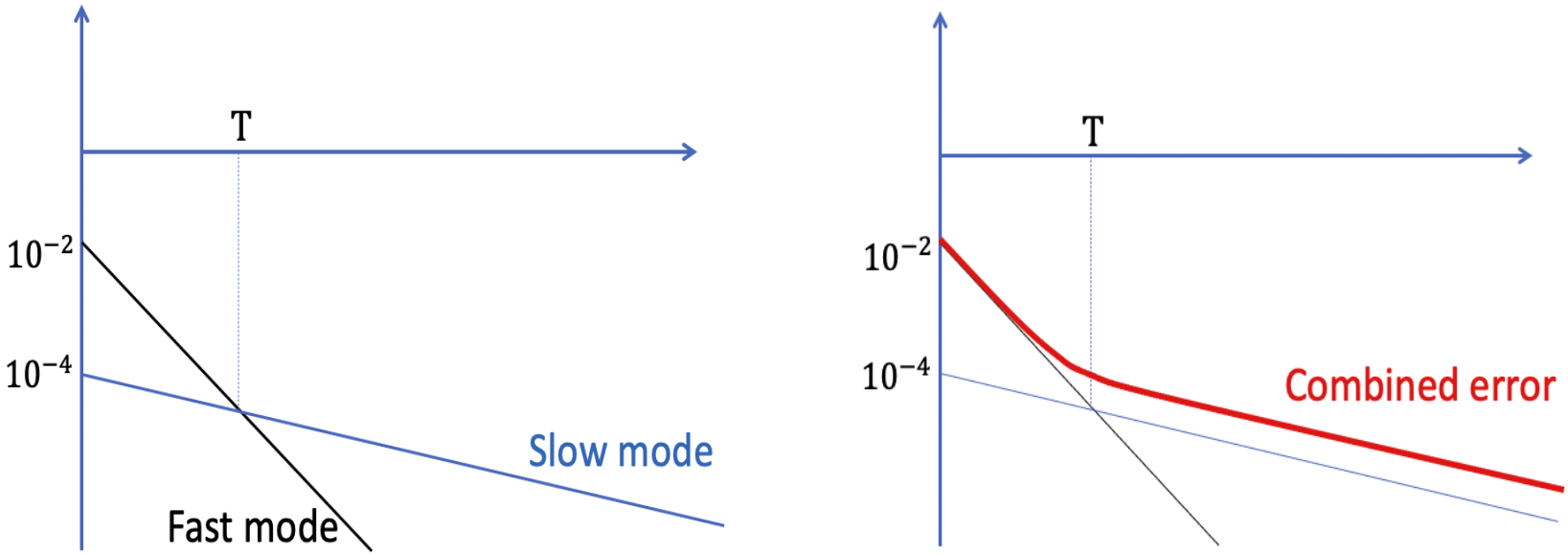
  $(x_1)^2 + 0.01(x_2)^2 + 0.0001(x_3)^2$.

  Then $2f(\mathbf{x}^r) = 0 + 0.01 \cdot 0.99^r + 0.0001 \cdot 0.9999^r$.
- How to analyze its behavior?

12/27

Explanation of Multi-stage Behavior (II)

How to draw $2f(\mathbf{x}^r) = 10^{-2} \cdot 0.99^r + 10^{-4} \cdot 0.9999^r$?



(a) Error plots of two modes (b) Error plot of the sum

Figure:

"Fast" mode: $10^{-2} \cdot 0.99^r$, starting at 10^{-2}, rate 0.99 (faster)
"Slow" mode: $10^{-4} \cdot 0.9999^r$, starting at 10^{-4}, rate 0.9999 (slower)
Their sum:

- **Stage 1**: almost overlap with fast mode;
- **Transition stage**: near iteration T where $10^{-2} \cdot 0.99^T = 10^{-4} \cdot 0.9999^T$
- **Stage 2**: almost parallel with slow mode

13/27

Explanation of Multi-stage Behavior (III)

Quantitative analysis of $f_r \triangleq 2f(\mathbf{x}^r) = \underbrace{10^{-2} \cdot 0.99^r}_{a_r} + \underbrace{10^{-4} \cdot 0.9999^r}_{b_r}$.

Stage 1: $1 \le r \le 298$.

- In this stage, $0.99^r \ge 0.99^{298} \approx 0.050$.
- So $a_r \ge 10^{-2} \cdot 0.050 = 5 \times 10^{-4} \ge 5b_r$. Thus a_r term dominates.
- More precisely, $a_r \le f_r = a_r + b_r \le a_r + a_r/5 = 1.2a_r$.
  Thus $f_r \approx a_r$, decreasing at rate 0.99.

Transition stage: $299 \le r < 630$.

Stage 2: $r \ge 630$.

- In this stage, $0.99^r \le 0.99^{630} \le 0.18 \times 10^{-2}$.
  $a_r \le 0.18 \times 10^{-4}$.
- At $r = 630$, $b_r = 10^{-4} \cdot 0.9999^{630} \ge 0.93 \times 10^{-4} > 5a_r$.
- Since $\frac{b_r}{a_r} = 10^{-2} \times (\frac{0.9999}{0.99})^r$ is increasing in r, thus when $r > 630$, we have $\frac{b_r}{a_r} > \frac{b_{630}}{a_{630}} > 5$. Thus b_r term dominates.
- More precisely, $b_r \le f_r = a_r + b_r \le b_r/5 + b_r = 1.2b_r$.
  Thus $f_r \approx b_r$, decreasing at rate 0.9999.

14/27

Practical Lessons

Lessons:

- GD will go through multiple stages in practice
- The theoretical complexity $\kappa \log(1/\epsilon)$ only captures the speed of the final stage
- Each stage has speed $\lambda_{\max}/\lambda_k$, and ends at roughly error λ_k
  - Need to consider "relative error"; usually we draw absolute error
  - The theoretical result $\kappa \log(1/\epsilon)$ provides a hint: indeed, "ratio of eigenvalues" is important for convergence speed. In this sense, $\kappa \log(1/\epsilon)$ bound is useful

15/27

Formal Result (reading)

Consider $\min_w F(w) \triangleq 0.5 w^\top Q w$.

Suppose the eigenvalues of Q are $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d$, and the corresponding eigenvectors are $v_1, \ldots, v_d$, where $\|v_i\| = 1, \forall i$.
For certain $\epsilon > 0$ and $m \in \{0, 1, \ldots, d\}$, we say a vector w is (m, ϵ)-skewed if (define $z_i = \langle w - w^*, v_i \rangle, \forall i$)

$$\sum_{i=d-m+1}^{d} \lambda_i z_i^2 \leq \epsilon \sum_{i=1}^{d} \lambda_i z_i^2 = 2\epsilon (F(w) - F^*), \tag{1}$$

Theorem 1: Suppose the eigenvalues of a PD matrix Q are $\lambda_1 \geq \cdots \geq \lambda_d \geq 0$, and $L = \lambda_1, \kappa = \frac{\lambda_1}{\lambda_d}$. Suppose we use GD with stepsize $\eta = 1/L$ to solve $\min_w 0.5 w^\top Q w$, generating iterates $\{w_r\}_0^\infty$. Suppose for certain $\epsilon \geq 0$ and $m \in \{0, 1, \ldots, d\}$, the initial point w_0 is (m, ϵ)-skewed as defined in (1). Then $F(w_r) \leq \left[\epsilon + (1 - \frac{\lambda_{d-m}}{\lambda_1})^2\right] F(w_0)$.

Corollary: $F(w_r) - F^* \leq (1 - \frac{1}{\kappa})^{2K} (F(w_0) - F^*)$.
Proof of Coro 1: Picking $\epsilon = 0$, then (1) holds for $m = 0$ (LHS is zero). Then $\frac{\lambda_{d-m}}{\lambda_1}$ becomes $\frac{\lambda_d}{\lambda_1} = \frac{1}{\kappa}$.

Remark: For different desired accuracy ϵ, the speed is different: determined by $\lambda_{\max}/\lambda_k$ for certain k, instead of $\lambda_{\max}/\lambda_{\min}$.

16/27

Formal Proof of Thm 1 (I) (reading)

Suppose the eigen-decomposition of Q is $Q = S\Lambda S^T$, where $\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_d)$ is a diagonal matrix and $S = (v_1, \ldots, v_d) \in R^{d\times d}$ is an orthogonal matrix with columns being eigenvectors of Q.

A minimizer of F satisfies $Qw^* = -b$, thus

$$F(w^*) = 0.5(w^*)^\top Qw^* + b^\top w^* = -0.5(w^*)^\top Qw^*.$$

Then

$$\begin{aligned} 0.5(w - w^*)^\top Q(w - w^*) &= 0.5w^\top Qw - w^\top Qw^* + 0.5(w^*)^\top Qw^* \\ &= 0.5w^\top Qw + w^\top b + 0.5(w^*)^\top Qw^* = F(w) - F(w^*). \end{aligned}$$

Define $z_r = S^\top(w_r - w^*)$, then $w_r - w^* = Sz_r$.

Combining the above two expressions, we have

$$F(w_r) - F^* = 0.5z_r^T S^\top QSz_r = 0.5z_r^T \Lambda z_r = 0.5\sum_{i=1}^{d} z_{r,i}^2\lambda_i = 0.5\sum_{i=1}^{d} \lambda_i(1-\eta\lambda_i)^{2K} z_{r,i}^2.$$

17/27

Formal Proof of Thm 1 (II) (reading)

Then

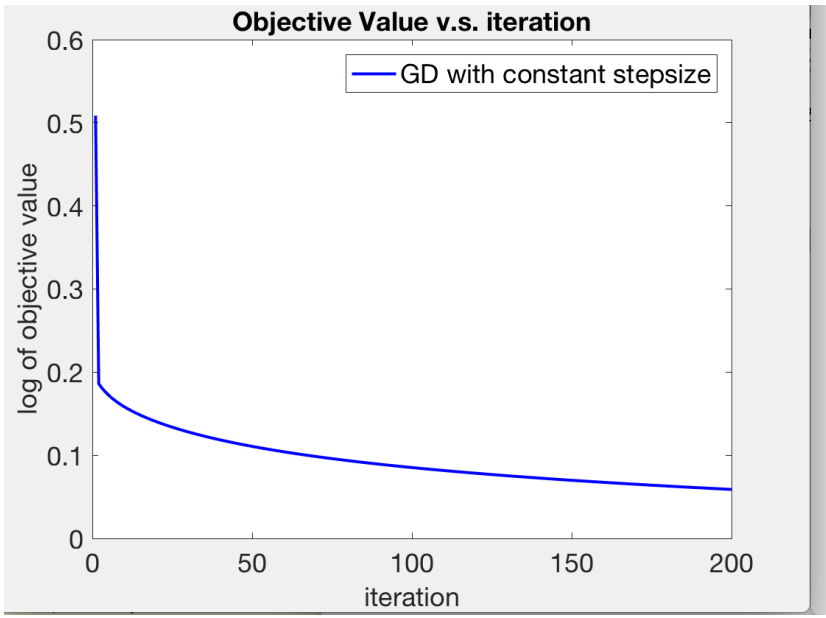
$$
\begin{aligned}
2(F(w_r) - F^*) &= \sum_{i=1}^{d} \lambda_i (1-\eta\lambda_i)^{2K} z_{0,i}^2 \\
&\leq \sum_{i=d-m+1}^{d} \lambda_i (1-\eta\lambda_i)^{2K} z_{0,i}^2 + \sum_{i=1}^{d-m} \lambda_i (1-\eta\lambda_i)^{2K} z_{0,i}^2 \\
&\leq \sum_{i=d-m+1}^{d} \lambda_i z_{0,i}^2 + \sum_{i=1}^{d-m} \lambda_i (1-\eta\lambda_i)^{2K} z_{0,i}^2 \\
&\leq 2\epsilon F(w_0) + \sum_{i=1}^{d-m} \lambda_i z_{0,i}^2 (1-\eta\lambda_{d-m})^{2r} \\
&\leq 2\left[\epsilon + (1-\frac{\lambda_{d-m}}{\lambda_1})^{2r}\right](F(w_0) - F^*).
\end{aligned}
$$

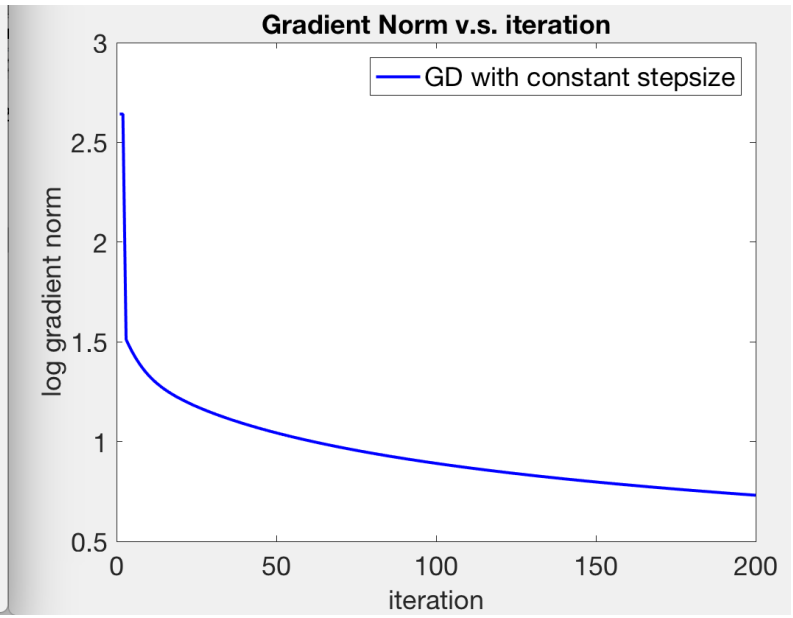
18/27

Original Data

Now we run GD on the whole dataset.

- $N = 612, d = 143$
- $L = 93$, $\alpha = 1/L = 0.0107$.





Question: Still multiple stages?

19/27

Eigenvalue Distribution

Eigenvalues

- Top 5: 93, 12.7, 6.1, 3.9, 3.3
- Bottom 8: $0, 0, 0, 0, 0, 0, 8 \times 10^{-7}, 5 \times 10^{-7}, 3 \times 10^{-7}$.

Non-strongly convex.

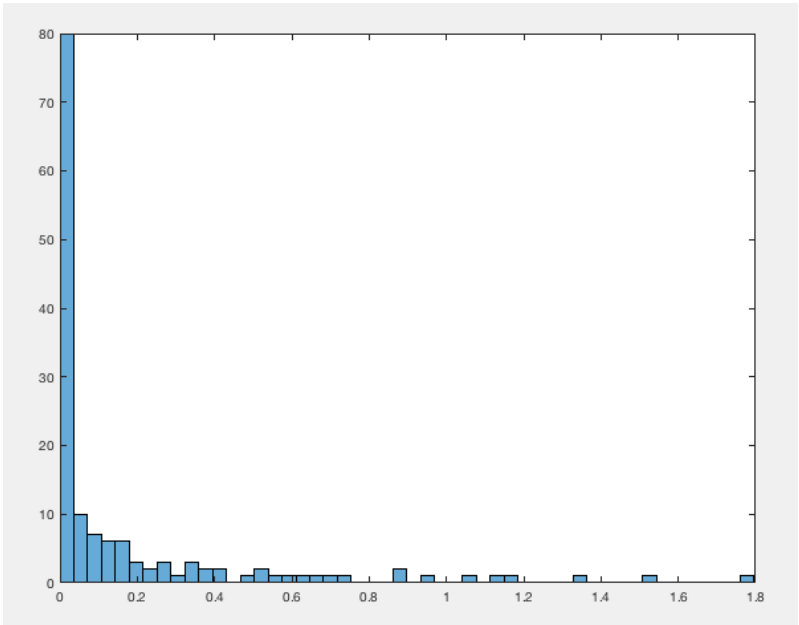
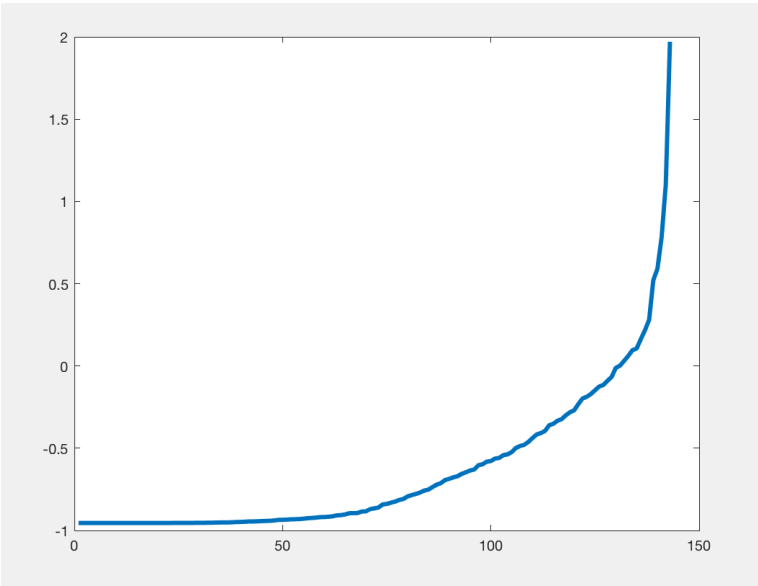


Figure: Histogram of all eigenvalues, except the top 5 $93, 12.7, 6.1, 3.9, 3.3$

20/27

Algorithm v.s. Eigenvalues

Observation: Algorithm plot highly correlated with eigenvalue distribution



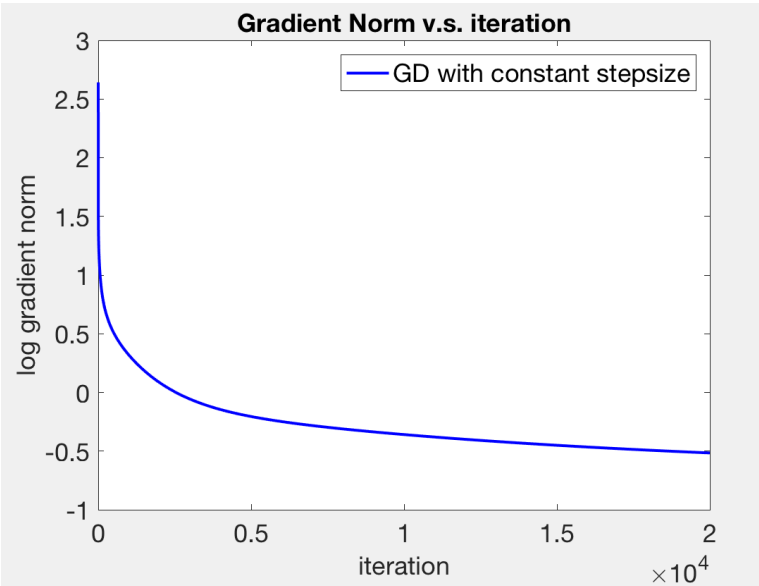


Figure: Left: eigenvalues; Right: gradient norm

21/27

Non-convex Problems

What about convex problems with changing Hessians, or NON-convex problems?

If the Hessian spectrum does not change too much, then using the spectrum to estimate the convergence speed is OK.

It will be much more accurate than using the $O(1/r)$ result.

Even if using $O(1/r)$ result leads to a seemingly good prediction, it is coincidence.

- because the spectrum is special; changing to a different problem would make it fail

22/27

Side: Eigenvalue Distribution

The above distribution can be partially explained by existing theory;

- random matrix theory has verified the distribution

see, e.g., Sagun et al.'17. Empirical Analysis of the Hessian of Over-Parametrized Neural Networks

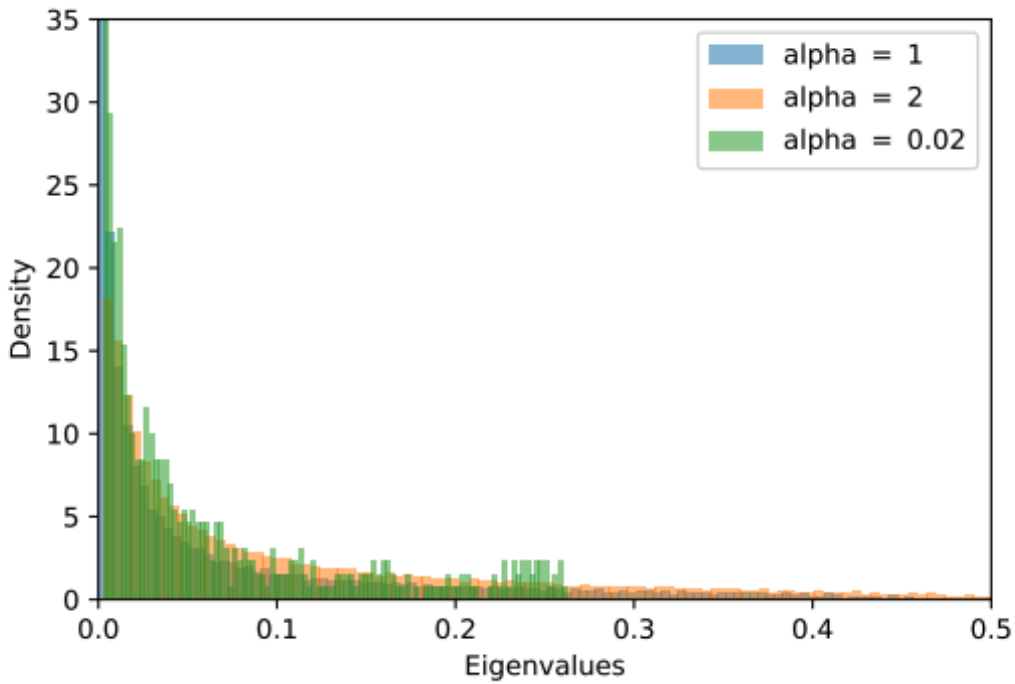


Figure: Sagun et al.'17 Fig 10b. The spectrum has outlier eigenvalues at 454.4, 819.5, and 92.7, respectively

23/27

Linear and Sublinear Rate

Insight on linear and sublinear:

- Because of the multi-stage behavior, linear rate looks like sublinear rate in early stages.
- For strongly convex problem, **finally** the linear rate will dominate
- Too many eigenvalues + tend to zero gradually:
  - the linear rate will never dominate (in reasonable amount of time)
  - it will be always like sublinear

This is also true for 1-dim logistic regression

- in finite region, it should be linear rate,
- but it may exhibit sublinear rate throughout

24/27

Practical Implication

What about a huge problem with, e.g., 10^6+ samples, 10^5+ features?

Condition number? Usually huge, say, $> 10^5$.

If you want low-accuracy solution (e.g. in machine learning), iteration complexity $\kappa \log(1/\epsilon)$ is not an issue.

- Convergence behavior of GD roughly depends on eigenvalues above or around the accuracy

However, if you want high-accuracy solution, GD-type methods are bad... (e.g. in PDE)

25/27

Summary: Effects of Eigenvalues

Performance of GD depends on spectrum of Hessian.

Effects of eigenvalues on performance include:

- Worst-case rate: depends on condition number (with ∞ # of iterations)
- Usually multiple stages, depends by magnitude of eigenvalues.

Tip: Use eigenvalues to study algorithms in your project (for small case first).

- This analysis is NOT limited to machine learning. Apply to GD for any problem!

26/27

Related References

Related papers:

- Hessian of neural-nets

  Sagun et al. “Empirical Analysis of the Hessian of Over-Parametrized Neural Networks”.

2020 Feb at CSL B02: I invited Lechao Xiao (Google Brain) to give a talk on the paper “”Disentangling Trainability and Generalization in Deep Neural Networks”

Abstract: “By analyzing the spectrum of the NTK, we formulate necessary conditions for trainability and generalization across a range of architectures, including FCN and CNN”.

27/27