EE445 Mod4-Lec4: Convex Optimization Problems: ML Models II

References: [Optimization Models] Chapter 8, sections 8.1-8.3 (except 8.2.3) and Chapter 13 (sections 13.1, 13.2, 13.3.1-5)

Topics for Module 4

- Lec1: Convex problems: convex sets and functions
- Lec2: Smooth unconstrained convex minimization & gradient descent
- Lec3 & 4: Convex Optimization Problems: ML models

This lecture's topics:

- Logistic Regression: derivation, properties, intuition, variations
- Penalty Function Approximation
- Other examples
- Wrap-up of Module 4

Logistic Regression: Overview

- Data: Continuous features $\{a_i\}$ and discrete labels $y_i \in \{0, 1\}$

- Goal: Find linear predictor

$x = \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} \in \mathbb{R}^2$

$$x_0 + x_1 a_i = \begin{cases} \text{positive} & \Rightarrow \quad y_i = 1 \\ \text{negative} & \Rightarrow \quad y_i = 0 \end{cases}$$

- Approach: Combine Bernoulli model with a linear predictor

- Examples: Hours studied vs. Pass/Fail, measurements vs. disease

Logistic Regression: Derivation

[this page : just FYI]

Rewriting the Bernoulli model in standard form,

for the i^{th} data point:

$$P\Big((a_i, y_i); p_i\Big) = p_i^{y_i}(1-p_i)^{1-y_i}$$

$x = e^{\log x}$

$$= \exp\left(y_i \underbrace{\log\left(\frac{p_i}{1-p_i}\right)} + \log(1-p_i)\right),$$

we can model the term multiplying y_i using our linear predictor,

$$\log\left(\frac{p_i}{1-p_i}\right) = x_0 + x_1 a_i,$$

which gives us,

$$\log(1-p_i) = -\log\left(1+\exp(x_0 + x_1 a_i)\right).$$

Combining the above expressions gives the "likelihood function": (for all m data points)

$$\mathcal{L}\Big(x_0, x_1; (a, y)\Big) = \prod_{i=1}^{m} \exp\Big(y_i(x_0 + x_1 a_i) - \log\left(1+\exp(x_0 + x_1 a_i)\right)\Big).$$

Logistic Regression: Derivation

We can fit our model parameters to the given data by maximizing the likelihood, or by minimizing the negative log-likelihood:

$$-\log \mathcal{L}\Big(x_0, x_1; (a, y)\Big) = \sum_{i=1}^{m} \log\big(1 + \exp(x_0 + x_1 a_i)\big) - y_i(x_0 + x_1 a_i)$$

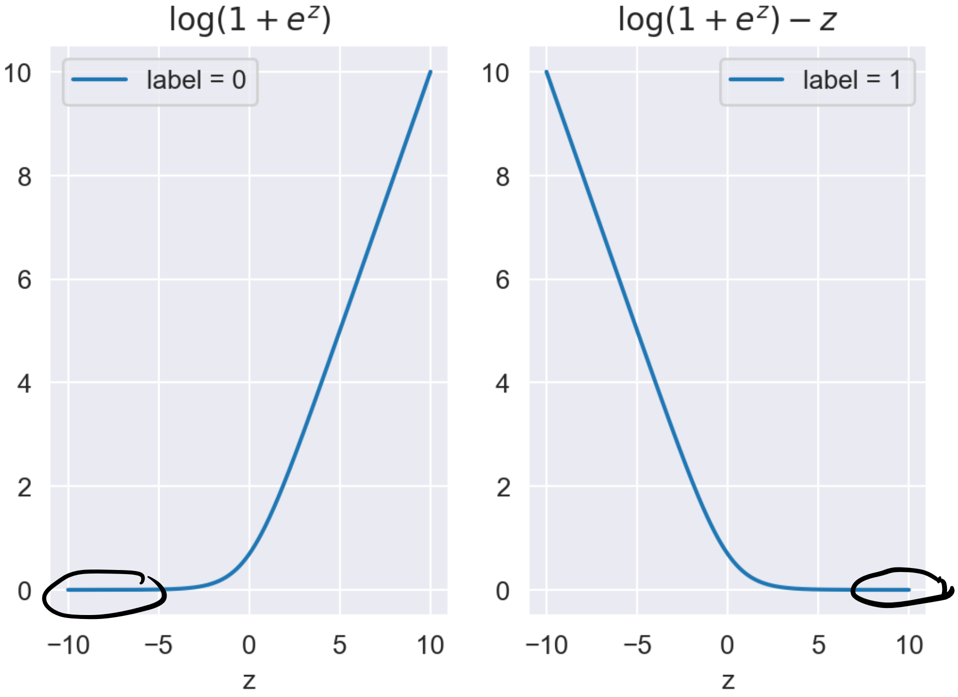
Explicitly, we solve the following problem

$$\min_{x_0, x_1} \sum_{i=1}^{m} \log(1 + \exp(x_0 + x_1 a_i)) - y_i(x_0 + x_1 a_i)$$

Logistic Regression: Intuition and Properties

$$\min_{x_0,x_1} \sum_{i=1}^{m} \log(1+\exp(x_0+x_1a_i)) - y_i(x_0+x_1a_i)$$

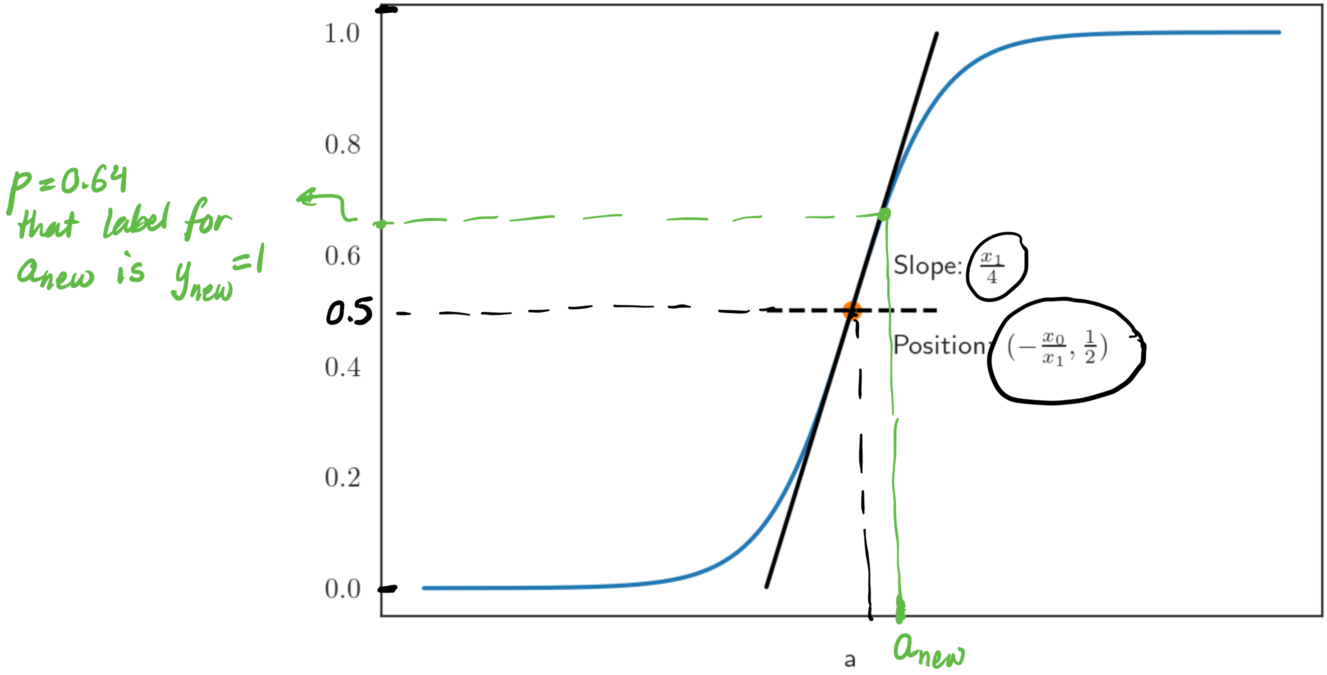
- If the label is 0, we want to make $\log(1+\exp(x_0+x_1a_i))$ as small as possible, equivalent to making $x_0+x_1a_i \ll 0$
- If the label is 1, can show objective decreases with respect to $x_0+x_1a_i$, so we want $x_0+x_1a_i \gg 0$



$\log(1+e^z)$
label = 0
$\log(1+e^z) - z$
label = 1
10
8
6
4
2
0
−10
−5
0
5
10
z

Logistic Regression: Intuition and Properties

- We look for intercept x_0 and slope x_1 that do the best job for all the data in the set.



Logistic Regression: Intuition and Properties

- The logistic loss function

$$f(x_0, x_1) = -\sum_{i=1}^{m} [\log(1 + \exp(x_0 + x_1 a_i)) - y_i(x_0 + x_1 a_i)]$$

  is **convex** (see HW 5, P6)
- It is also differentiable, and 'nice' to solve, e.g., by gradient descent (you will try this in the last Python notebook, to be posted today)

Logistic Regression: Intuition and Properties

- logistic loss function

$$f(x_0, x_1) = -\sum_{i=1}^{m} \left[\log(1 + \exp(x_0 + x_1 a_i)) - y_i(x_0 + x_1 a_i)\right]$$

- Sometimes a regularizer is added, e.g., $r(x_0, x_1) = x_0^2 + x_1^2$
- $f(x) + r(x)$ is still convex (sum of two convex functions)

after we learn such a model, how is it used for prediction?

- For a future data point with feature a, we have $p = \frac{\exp(x_0+x_1 a)}{1+\exp(x_0+x_1 a)}$ (see p. 9)
- We can add convex constraints on parameters (e.g., upper/lower bounds on values, $x = (x_0, x_1)$ restricted to a ball, etc.

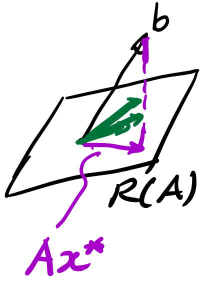
(General) Norm Approximation Problems

$[\quad]_{m \times n}$

$$\underset{x}{\text{minimize}} \quad \|Ax - b\|$$

$(A \in \mathbf{R}^{m \times n}$ with $m \geq n$, $\|\cdot\|$ is a norm on $\mathbf{R}^m)$

for $\|Ax-b\|_2^2$:



- **geometric interpretation** of solution $x^\star = \mathrm{argmin}_x \|Ax - b\|$: $Ax^\star$ is point in $\mathcal{R}(A)$ closest to b according to the norm
- **estimation**: linear measurement model

$$y = Ax + v$$

y are measurements, x is unknown, v is measurement error or noise
given $y = b$, best guess of x is $x^\star$

- **optimal design**: x are design variables (input), Ax is result (output)
$x^\star$ is design that best approximates desired result b

Norm Approximation: Examples

- least-squares approximation ($\|\cdot\|_2$): solution satisfies

$$A^T A x = A^T b$$

($x^\star = (A^T A)^{-1} A^T b$ if $\mathbf{Rank}\, A = n$) (from Mod1-Mod2)

- Chebyshev approximation ($\|\cdot\|_\infty$): can be solved as a Linear Program:

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & -t\mathbf{1} \preceq Ax - b \preceq t\mathbf{1} \end{array}$$

- sum of absolute residuals approximation ($\|\cdot\|_1$): can be solved as an Linear Program:

$$\begin{array}{ll} \text{minimize} & \mathbf{1}^T y \\ \text{subject to} & -y \preceq Ax - b \preceq y \end{array}$$

Norm Approximation: Examples

$r = Ax - b \in \mathbb{R}^m$

- least-squares approximation ($\|\cdot\|_2$): solution satisfies

$$A^T Ax = A^T b$$

① $\min_x \ \|\overbrace{Ax - b}^{r}\|_\infty$

($x^\star = (A^T A)^{-1} A^T b$ if $\mathbf{Rank}\, A = n$)

- Chebyshev approximation ($\|\cdot\|_\infty$): can be solved as a Linear Program:

① ⟺ ② ⟺ ③

③
$$\begin{array}{ll} \underset{x,t}{\text{minimize}} & t \\ \text{subject to} & -t\mathbf{1} \preceq Ax - b \preceq t\mathbf{1} \end{array}$$

②
$$\begin{array}{ll} \underset{x,t}{\min.} & t \\ \text{s.t.} & -t \le a_i^T x - b_i \le t \end{array} \quad i = 1, \dots, m$$

- sum of absolute residuals approximation ($\|\cdot\|_1$): can be solved as an Linear Program:

$$\begin{array}{ll} \text{minimize} & \mathbf{1}^T y \\ \text{subject to} & -y \preceq Ax - b \preceq y \end{array}$$

Norm Approximation: Examples

- least-squares approximation ($\|\cdot\|_2$): solution satisfies

$$A^T A x = A^T b$$

($x^\star = (A^TA)^{-1}A^Tb$ if $\mathbf{Rank}\, A = n$)

- Chebyshev approximation ($\|\cdot\|_\infty$): can be solved as a Linear Program:

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & -t\mathbf{1} \preceq Ax - b \preceq t\mathbf{1} \end{array}$$

- sum of absolute residuals approximation ($\|\cdot\|_1$): can be solved as an Linear Program:

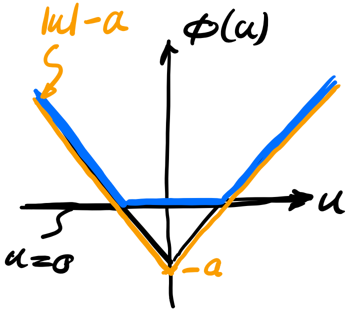
$\min_x \|Ax-b\|_1$

$r = Ax - b = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_m \end{bmatrix}$

$$\begin{array}{ll} \text{minimize} & \mathbf{1}^T y \\ \text{subject to} & -y \preceq Ax - b \preceq y \end{array}$$

– solution x^* gives a spare r

Penalty Function Approximation



$$\begin{array}{ll} \underset{x,r}{\text{minimize}} & \phi(r_1) + \cdots + \phi(r_m) \\ \text{subject to} & r = Ax - b \end{array}$$

($A \in \mathbf{R}^{m \times n}$, $\phi : \mathbf{R} \to \mathbf{R}$ is a convex penalty function)

examples

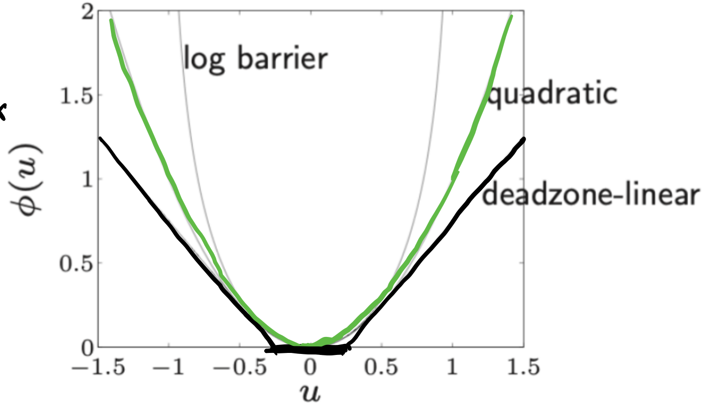
- quadratic: $\phi(u) = u^2$, $\phi(u) = |u|$
- deadzone-linear with width a:

  - pointwise max of 2 convex fcts $\Rightarrow$ convex

  $$\phi(u) = \max\{0, |u| - a\}$$

  - zero penalty for values in $[-a, a]$ "deadzone"
- log-barrier with limit a:

$$\phi(u) = \begin{cases} -a^2 \log(1 - (\frac{u}{a})^2) & |u| < a \\ \infty & \text{else} \end{cases}$$



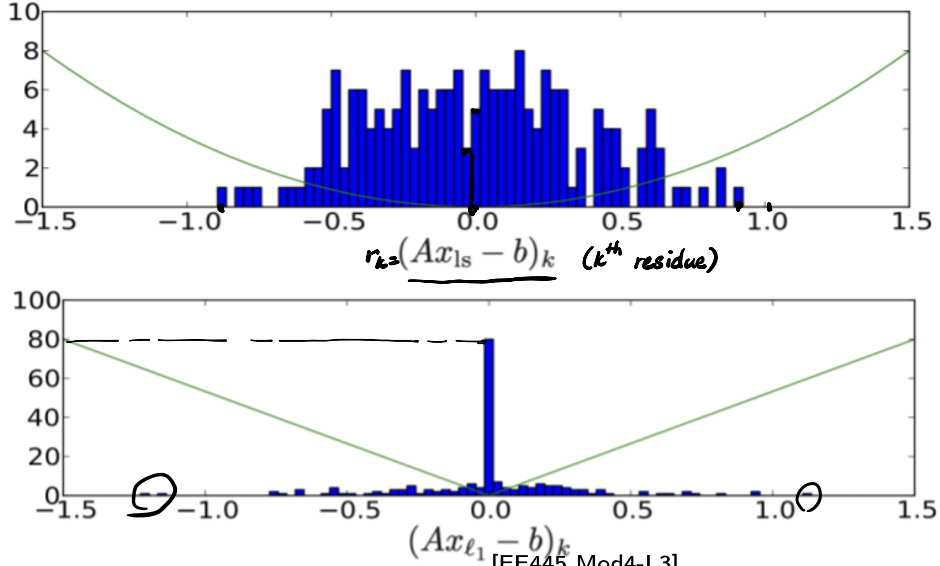
ℓ_2-norm vs ℓ_1-norm Penalties

example: histogram of residuals $Ax - b$ (A is 200×80) for $\quad r = Ax - b \in \mathbb{R}^{200}$

$$x_{\text{ls}} = \operatorname{argmin} \|Ax - b\|_2, \qquad x_{\ell_1} = \operatorname{argmin} \|Ax - b\|_1$$

Recall: similar intuition to regression with ℓ_1 regularization (last lecture)

vertical axis: histogram of r (# of r_i's falling in each bin)



- most residues are not zero (vector $r = Ax - b$ is NOT "sparse")

- many residues are zero! ($\sim$ 80 out of 200)
- values of r_i also have a wider spread; can be larger than those for x_{ls}

Convex Classification Problems

- classification: linear discrimination
- approximate linear discrimination of non-separable sets
- robust linear discrimination
- support vector machine (SVM)

Wrap up (of Module 4)

- Many real-world problems can be expressed as Convex optimization problems
- We focused on examples in ML, but also very common in:
  signal processing (signal reconstruction, denoising), communication system design (power allocation, rate allocation), feedback control design, mechanical systems design, statisitics, finance,...
- The key is to recongnize when a problem can be cast or modeled as a convex one
  - nontrivial, needs skill and practice!
  - important to know basic convex sets/functions and properties that preserve convexity
  - combine with linear algebra and spectral methods seen in Mod1-Mod 3

Wrap up (of Module 4)

- Many real-world problems can be expressed as Convex optimization problems
- We focused on examples in ML, but also very common in:
  signal processing (signal reconstruction, denoising), communication system design (power allocation, rate allocation), feedback control design, mechanical systems design, statisitics, finance,. . .
- The key is to **recongnize** when a problem can be cast or modeled as a **convex** one
  - nontrivial, needs skill and practice!
  - important to know basic convex sets/functions and properties that preserve convexity
  - combine with linear algebra and spectral methods seen in Mod1-Mod 3

Wrap up (of Module 4)

- Historically: the more people understood convexity, the more they looked for (and found) convex problems
- Knowing about convexity can help even when your target problem isnot convex: convex relaxations/approximations, convex subproblems,. . .
- We hope this glimpse into convexity motivates you to learn more:
  grad courses, online material, book "Convex Optimization" by Boyd & Vandenberghe (and many others) (course EE578 @ UW)