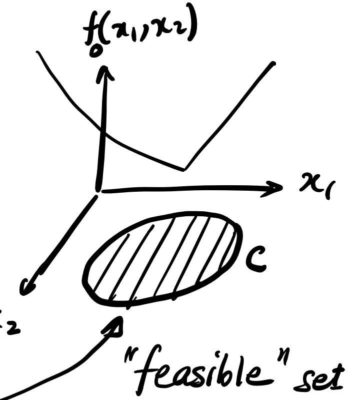
EE445 Mod4-Lec3: Convex Optimization Problems: ML models

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

Convex Optimization problems



standard form convex optimization problem

$$
\begin{array}{ll}
\underset{x}{\text{minimize}} & f_0(x) \\
\text{subject to} & f_i(x) \le 0, \quad i = 1, \dots, m \\
 & a_i^T x = b_i, \quad i = 1, \dots, p
\end{array}
$$

objective

- f_0, $f_1, \dots, f_m$ are convex
- equality constraints are affine

important property: local optima are globally optimal!

Local optima are *global* in convex problems

any **locally optimal** point of a convex problem is **globally optimal**

prove via contradiction

proof: suppose x is locally optimal, and y is optimal with $f_0(y) < f_0(x)$

x locally optimal means there is an $R > 0$ such that

$$z \text{ feasible}, \quad \|z - x\|_2 \le R \quad \Longrightarrow \quad f_0(z) \ge f_0(x)$$

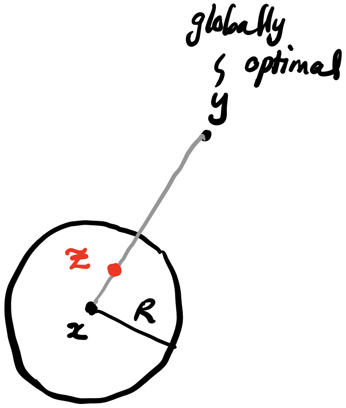
consider $z = \theta y + (1-\theta)x$ with $\theta = \frac{R}{2\|y-x\|_2}$

- $\|y - x\|_2 > R$, so $0 < \theta < 1/2$
- z is a convex combination of two feasible points, hence also feasible
- $\|z - x\|_2 = R/2$ and

$$f_0(z) \le \theta f_0(x) + (1-\theta) f_0(y) < f_0(x)$$

which contradicts our assumption that x is locally optimal.

Local optima are *global* in convex problems



any **locally optimal** point of a convex problem is **globally optimal**

proof: suppose x is locally optimal, and y is optimal with $f_0(y) < f_0(x)$
x locally optimal means there is an $R > 0$ such that

$$z \text{ feasible}, \quad \|z - x\|_2 \le R \quad \Longrightarrow \quad f_0(z) \ge f_0(x)$$

consider $z = \theta y + (1 - \theta)x$ with $\theta = \dfrac{R}{2\|y - x\|_2}$

- $\|y - x\|_2 > R$, so $0 < \theta < 1/2$
- z is a convex combination of two feasible points, hence also feasible
- $\|z - x\|_2 = R/2$ and

$$f_0(z) \le \theta f_0(x) + (1 - \theta) f_0(y) < f_0(x)$$

which contradicts our assumption that x is locally optimal.

Local optima are *global* in convex problems

any **locally optimal** point of a convex problem is **globally optimal**

proof: suppose x is locally optimal, and y is optimal with $f_0(y) < f_0(x)$
x locally optimal means there is an $R > 0$ such that

$$z \text{ feasible}, \quad \|z - x\|_2 \le R \quad \Longrightarrow \quad f_0(z) \ge f_0(x)$$

consider $z = \theta y + (1 - \theta)x$ with $\theta = \frac{R}{2\|y-x\|_2}$

- $\|y - x\|_2 > R$, so $0 < \theta < 1/2$
- z is a convex combination of two feasible points, hence also feasible
- $\|z - x\|_2 = R/2$ and

$$f_0(z) \le \theta f_0(x) + (1 - \theta) f_0(y) < f_0(x)$$

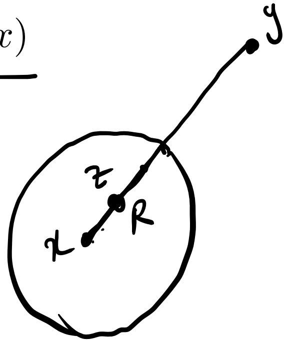
which contradicts our assumption that x is locally optimal.

Local optima are *global* in convex problems

any locally optimal point of a convex problem is globally optimal

proof: suppose x is locally optimal, and y is optimal with $f_0(y) < f_0(x)$

x locally optimal means there is an $R > 0$ such that



$$z \text{ feasible}, \quad \|z - x\|_2 \leq R \quad \Longrightarrow \quad f_0(z) \geq f_0(x)$$

consider $z = \theta y + (1 - \theta)x$ with $\theta = \frac{R}{2\|y-x\|_2}$

- $\|y - x\|_2 > R$, so $0 < \theta < 1/2$
- z is a convex combination of two feasible points, hence also feasible
- $\|z - x\|_2 = R/2$ and

convexity of f_0

$$f_0(z) \leq \theta f_0(x) + (1 - \theta)\underbrace{f_0(y)}_{<f_0(x)} < f_0(x)$$

which contradicts our assumption that x is locally optimal. so $f_0(y) \not< f_0(x)$

Classes of Convex Optimization Problems

- Linear Program: linear objective function f_0 and constraint functions f_i
- Quadratic Program: convex quadratic f_0, linear f_i
- Quadratically-constrained Quadratic Program: convex quadratic f_0, convex quadratic f_i
- Second-order Cone Program
- Semi-definite Program
- ...

While we won't discuss this, note that recongizing a practical problem as an instance of one of these classes helps with picking the right algorithm.

convex sets
convex function
" optimization problems

Optimization: Machine Learning Models

Recall: many ML problems seek to build a prediction model

$$g(a; x) \approx y$$

given a data set

$$\Big\{(a_1, y_1), \ldots, (a_m, y_m)\Big\},$$

with components

- $a_i = (a_{i1}, \ldots, a_{in})$ - data features
- $y_i \in \mathbf{R}$ or $\{0, 1\}$ - data value or label/class
- $g : \mathbf{R}^n \to \mathbf{R}$ or $\{0, 1\}$ - prediction function
- $x = (x_1, \ldots, x_n)$ - model parameters
- m - # of data points
- n - # of data features

Optimization: Machine Learning (ML) Models

We can fit a model to the given data by solving an optimization problem of the form

$$\text{minimize}_x \; \sum_{i=1}^m f_i\Big(\underbrace{g(a_i;x)}_{\text{predictor output}}, \overset{\text{label}}{y_i}\Big) + \underbrace{r(x)}_{}, \qquad \text{fitting error}$$

with components

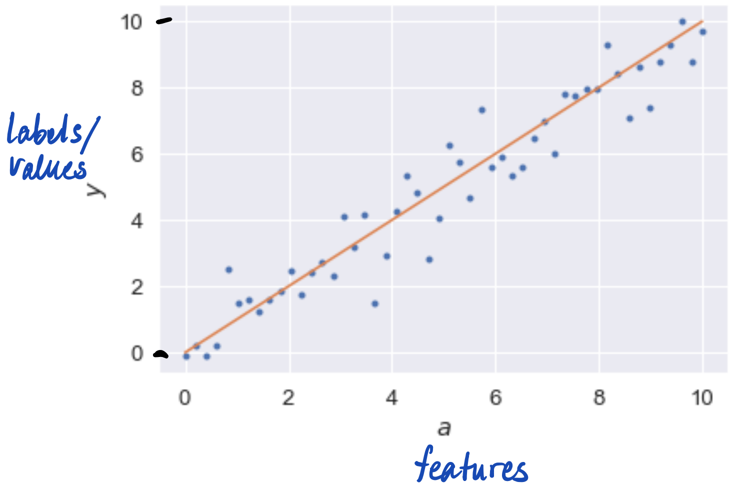
- $x = (x_1, \ldots, x_n)$ - model parameters we want to learn
- $f_i : \mathbf{R}^n \to \mathbf{R}$ - "loss" functions: measure how well the model fits the data for given parameters; e.g., $(g(a_i; x) - y_i)^2$ ⇝ least squares
- $r(x) : \mathbf{R}^n \to \mathbf{R}$ - regularization function

$r(x) = \|x\|_2^2$ ⇝ ridge-regression

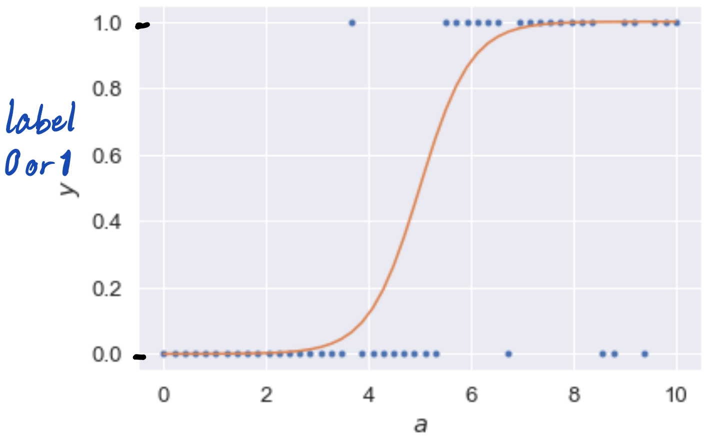
Optimization: Machine Learning Models

We consider two common problems in ML:

Linear Regression



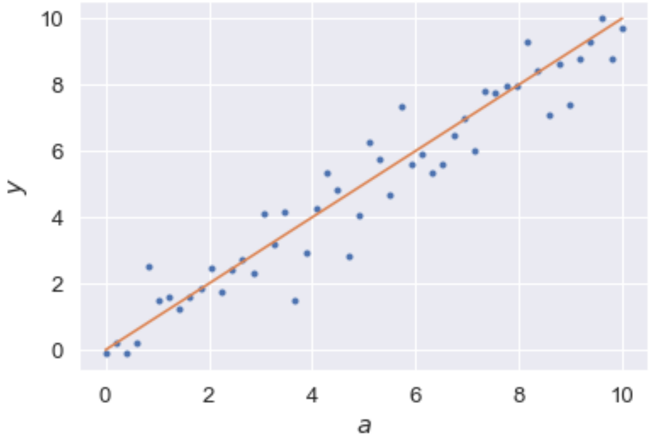
Logistic Regression (Classification)



Optimization: Machine Learning Models

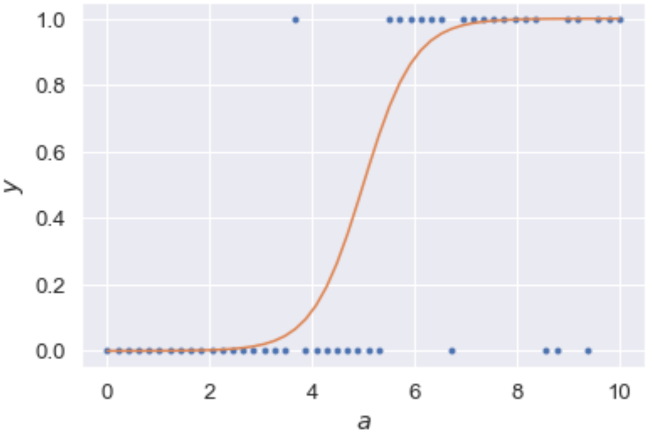
We consider two common problems in ML:

Linear Regression (seen in Mod 2)



(will see next lecture)

Logistic Regression (Classification)

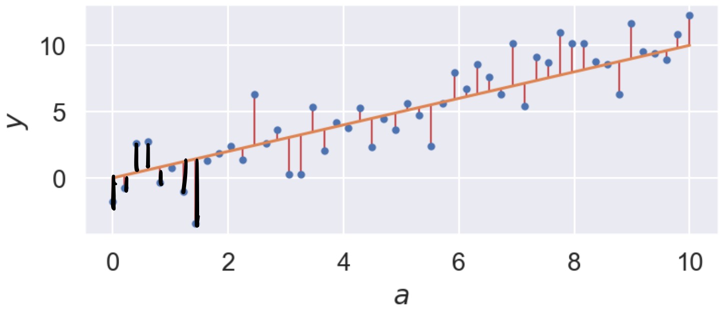


Linear Regression: Overview

- Data: Continuous features $\{a_i\}$ and outputs $\{y_i\}$
- Goal: Find linear predictor $\quad x_0 + x_1 a_i \approx y_i$
- Studied in Module 2 in detail

Linear Regression: Intuition and Properties

$$\min_{x_0, x_1} \sum_{i=1}^{m} (y_i - x_0 - x_1 a_i)^2$$



- Minimize the least-squares distance between observations y_i and predictions $x_0 + x_1 a_i$.
- The problem is convex, smooth, and easy to solve.
- Linear regression has a closed-form solution (as seen in Mod2)
- but often solved more efficiently by iterative algorithms

Regularization: Overview

Many problems in machine learning add a regularization term $r(x)$ to the objective function to

- incorporate prior knowledge about *structure* in x, e.g., sparsity or smoothness
  $\approx$ many zeros $\approx$ small changes
- help avoid overfitting,
- get more robust (to data perturbations) solutions, or
- improve the stability of the solution process. (numerical behavior of iterative alg's)

$\lambda \in \mathbb{R}$: knob

Two popular forms of regularized linear regression:

- Lasso - $\min_x f(x) + \lambda \|x\|_1$, where $\|x\|_1 = \sum_{i=1}^n |x_i|$ → find sparse x
- Ridge - $\min_x f(x) + \lambda \|x\|_2^2$, where $\|x\|_2^2 = \sum_{i=1}^n x_i^2$

Regularization: Geometric Interpretation

Consider the constrained least-squares problem

$$\left[\begin{array}{ll} \text{minimize}_x & \frac{1}{2}\|Ax - y\|_2^2 \\ & \|x\|_{\underline{p}} \le t \end{array}\right. \qquad t=1$$

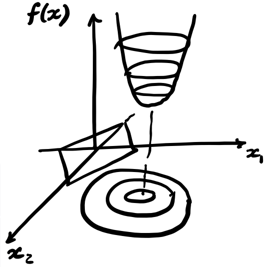
Choice of norm influences properties of solution x: with $\underline{p = 1}$, solutions tend to occur on the vertices, where many $x_i = 0$

$$x = \begin{bmatrix} 0 \\ 0 \\ \times \\ 0 \\ \times \end{bmatrix} \quad \text{sparse}$$

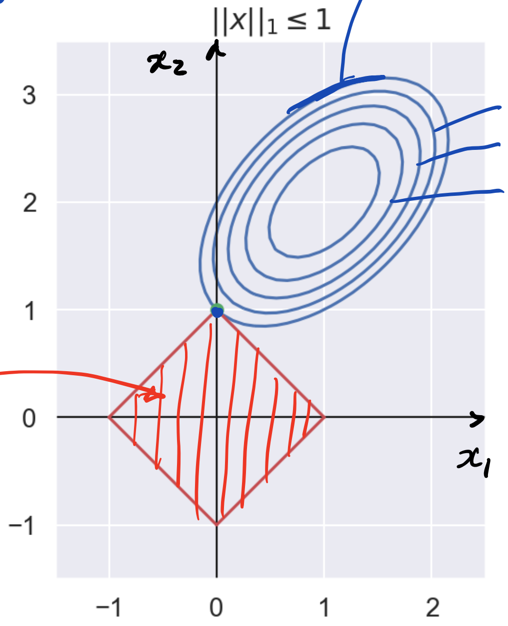
points where fit error $=10$

$\{x \mid f(x)=10\}$

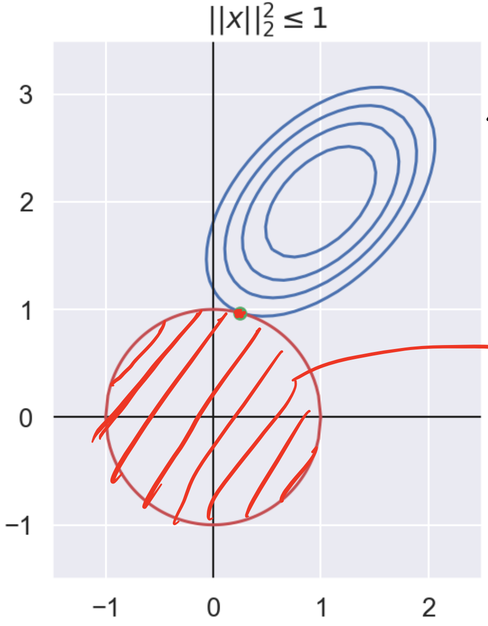
$f(x) = \|Ax-y\|^2$



$x^* = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$



$\{x \mid \|x\|_1 \le 1\}$



$\{x \mid \|x\|_2^2 \le 1\}$

here: $x^* = \begin{bmatrix} 0.1 \\ 0.93 \end{bmatrix}$

Regularization: Relaxed Constraints

We can move the norm from a constraint into the objective function to get

$$\text{minimize}_x \quad \frac{1}{2}\|Ax - y\|_2^2 + \lambda \|x\|_p$$

where regularization parameter λ balances model error with how much we regularize.

The Lasso $(p = 1)$ is often used to find sparse solutions. Ridge regression $(p = 2)$ is often used for ill-conditioned problems.

More generally: regularizers can promote other structures:
For example, if the parameters form a matrix X, a low-rank matrix is often desired (e.g., the 'matrix completion problem' for recommender systems).