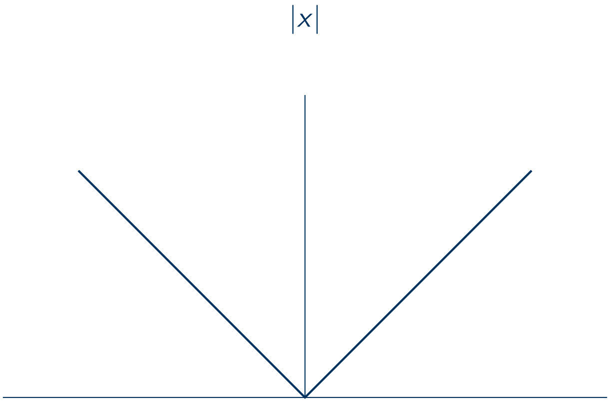
Machine Learning: Optimization 3

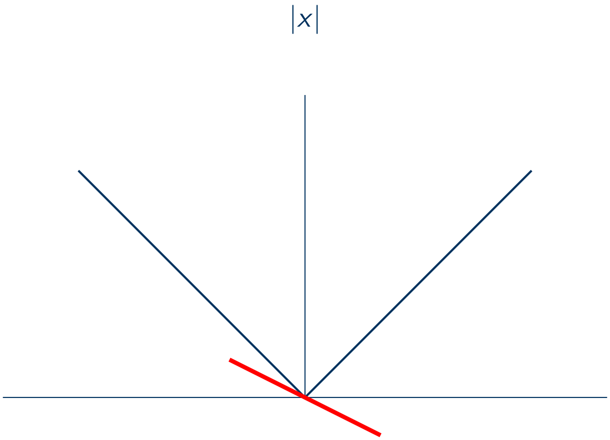
Hao Tang

February 14, 2024

Subgradients for absolute values

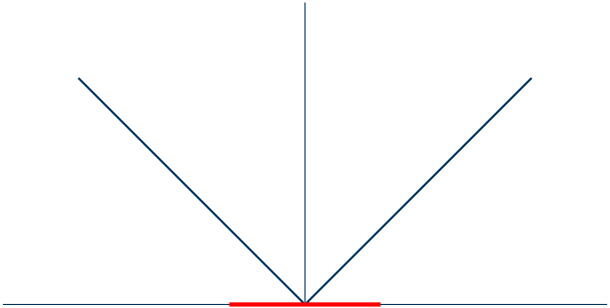


Subgradients for absolute values

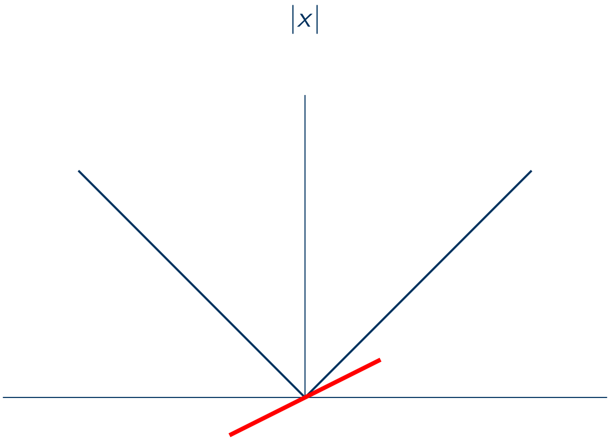


Subgradients for absolute values

$|x|$



Subgradients for absolute values



Subgradient

- A subgradient at x is a vector g that satisfies

$$f(y) \geq f(x) + g^\top (y - x) \tag{1}$$

  for any y, and the set of subgradients at x is denoted as $\partial f(x)$.
- Obviously, $\nabla f(x) \in \partial f(x)$, if $\nabla f(x)$ exists.
- Convergence theorems can be ported to subgradient descent.

Hinge loss

- The hinge loss is defined as

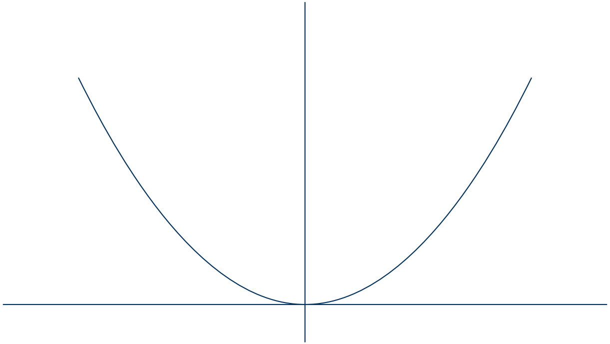
$$\ell_{\text{hinge}}(w; x, y) = \max(0, 1 - yw^\top x). \tag{2}$$

- Just like the absolute value, the hinge loss is continuous and convex, but it is not differentiable.

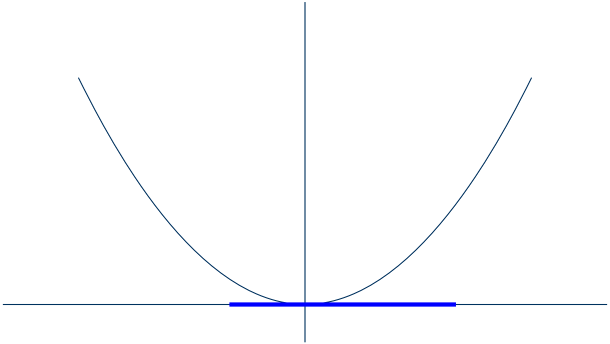
$$\nabla_w \ell = \begin{cases} 0 & \text{if } yw^\top x \geq 1 \\ -yx & \text{if } yw^\top x < 1 \end{cases} \tag{3}$$

- When $yw^\top x = 1$, we can pick and choose any vector that supports the loss function from below as the subgradient. In fact, 0 and $-yx$ both work.

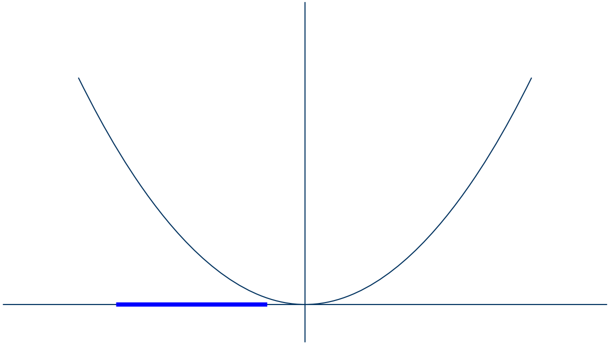
Constrained optimization



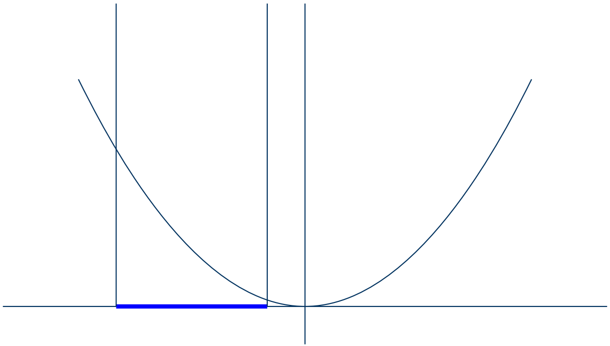
Constrained optimization



Constrained optimization



Setting up a barrier



An example problem with constraints

- The problem

$$\begin{aligned} \min_{x} \quad & x^2 \\ \text{s.t.} \quad & -2.5 \leq x \leq -0.5 \end{aligned} \tag{4}$$

  is an example of a contrained optimization problem.

- The inequality $-2.5 \leq x \leq -0.5$ is called a constraint.

- Solutions that satisfy the constraints are called **feasible** solutions.

Setting up a barrier

- The problem

$$\begin{aligned} \min_x \quad & x^2 \\ \text{s.t.} \quad & -2.5 \leq x \leq -0.5 \end{aligned} \tag{5}$$

is equivalent to

$$\min_x x^2 + V_-(x) \tag{6}$$

if

$$V_-(x) = \begin{cases} 0 & \text{if } -2.5 \leq x \leq -0.5 \\ \infty & \text{otherwise} \end{cases} \tag{7}$$

An example problem with constraints

- The problem

$$\begin{aligned} \min_{w} \quad & L(w) \\ \text{s.t.} \quad & \|w\|_2^2 \leq 1 \end{aligned} \tag{8}$$

  is an example of a contrained optimization problem.

- The inequality $\|w\|_2^2 \leq 1$ is called a constraint.

- Solutions that satisfy the constraints are called **feasible** solutions.

Setting up a barrier

- We can write the optimization problem as

$$\min_{w} \qquad L(w) + V_{-}(\|w\|_2^2 - 1), \tag{9}$$

where

$$V_{-}(s) = \begin{cases} 0 & \text{if } s \leq 0 \\ \infty & \text{if } s > 0 \end{cases}. \tag{10}$$

- This does not change anything; both problems are equally hard (or easy) to solve.

Soften the constraints

- We can approximate

$$\min_{w} \quad L(w) + V_{-}(\|w\|_2^2 - 1) \tag{11}$$

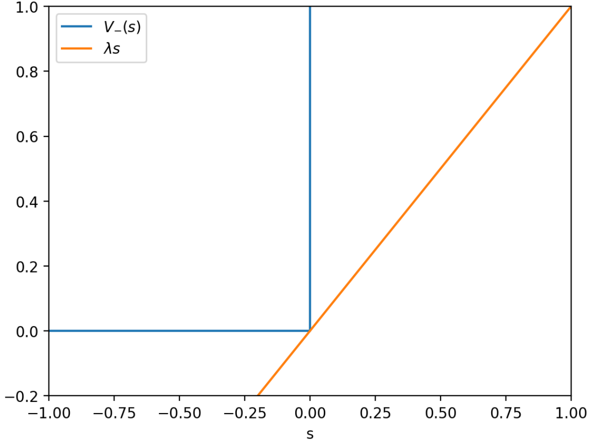
with

$$\min_{w} \quad L(w) + \lambda(\|w\|_2^2 - 1), \tag{12}$$

for some $\lambda \geq 0$.

- Note that $\lambda s \leq V_{-}(s)$ for all s.

Soften the constraints



Lagrangian

- In general, if you have a optimization problem

$$\begin{aligned} \min_{x} \quad & f(x) \\ \text{s.t.} \quad & h(x) \leq 0 \end{aligned} \tag{13}$$

  the **Lagrangian** is defined as

$$f(x) + \lambda h(x) \tag{14}$$

  for $\lambda \geq 0$.

- The value λ is called the Lagrange multiplier.

Solving the Lagrangian

- Solve $g(\lambda) = \min_x[f(x) + \lambda h(x)]$ for a particular λ.
- Find $\hat{\lambda}$ such that $\min_x[f(x) + \hat{\lambda} h(x)]$ gives a feasible solution.
- Suppose $\hat{x} = \operatorname{argmin}_x[f(x) + \hat{\lambda} h(x)]$ and $x^* = \operatorname{argmin}_{x:h(x)\leq 0} f(x)$.

$$f(\hat{x}) + \hat{\lambda} h(\hat{x}) \leq f(x^*) + \hat{\lambda} f(x^*) \leq f(x^*) \tag{15}$$

Solving the Lagrangian

- We want $f(\hat{x}) = f(\hat{x}) + \hat{\lambda} h(\hat{x})$ leading to $f(\hat{x}) \leq f(x^*)$, so that we can conclude $f(\hat{x}) = f(x^*)$.

- If we want $\hat{\lambda} h(\hat{x}) = 0$, then either $\hat{\lambda} = 0$ or $h(\hat{x}) = 0$.

- When $\hat{\lambda} = 0$, the minimizer of f is a feasible solution already.

- When $h(\hat{x}) = 0$, the minimizer of f is not a feasible solution, and we are on the edge of a constraint.

A unigram model

Row, row, row your boat, gently down the stream
Merrily, merrily, merrily, merrily, life is but a dream

A unigram model

Row, row, row your boat, gently down the stream
Merrily, merrily, merrily, merrily, life is but a dream

- There are 18 words.
- Intuitively,

$$p(\text{row}) = \frac{3}{18} \qquad p(\text{merrily}) = \frac{4}{18} \qquad p(\text{is}) = \frac{1}{18} \tag{16}$$

A unigram model

- There are 13 unique words.
- We refer to the set of unique words $V = \{\text{row}, \text{your}, \text{boat}, \text{gently}, \text{down}, \text{the}, \text{stream}, \text{merrily}, \text{life}, \text{is}, \text{but}, \text{a}, \text{dream}\}$ as the vocabulary.
- We assign each word v a probability β_v.
- The probability of a word is

$$p(w) = \prod_{v \in V} \beta_v^{\mathbb{1}_{v=w}}. \tag{17}$$

A unigram model

- We assume that each word is independent of others.
- This assumption is obviously wrong, but can go really far.
- The likelihood of β given the data is

$$\log p(w_1, \ldots, w_N) = \log \prod_{i=1}^{N} p(w_i) = \log \prod_{i=1}^{N} \prod_{v \in V} \beta_v^{\mathbb{1}_{v=w_i}}. \tag{18}$$

- Since β is a probability vector, we have the assumption

$$\sum_{v \in V} \beta_v = 1. \tag{19}$$

A unigram model

- We arrive at the optimization problem

$$\begin{aligned} \min_{\beta} \quad & -\sum_{i=1}^{N} \sum_{v \in V} \mathbb{1}_{v=w_i} \log \beta_v \\ \text{s.t.} \quad & \sum_{v \in V} \beta_v = 1 \end{aligned} \tag{20}$$

- Its Lagrangian is

$$F = -\sum_{i=1}^{N} \sum_{v \in V} \mathbb{1}_{v=w_i} \log \beta_v + \lambda \left(\sum_{v \in V} \beta_v - 1 \right). \tag{21}$$

A unigram model

- Solving the optimality condition gives

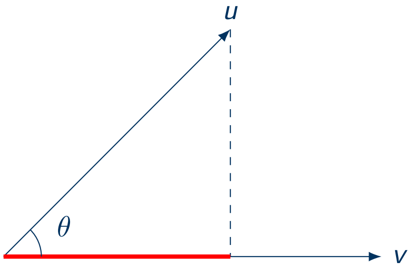
$$\frac{\partial F}{\partial \beta_k} = \sum_{i=1}^{N} \mathbb{1}_{k=w_i} \frac{1}{\beta_k} - \lambda = 0 \implies \beta_k = \frac{1}{\lambda} \sum_{i=1}^{N} \mathbb{1}_{k=w_i}. \tag{22}$$

A unigram model

$$\sum_{v\in V}\beta_v = \sum_{v\in V}\frac{1}{\lambda}\sum_{i=1}^{N}\mathbb{1}_{v=w_i} = 1 \implies \lambda = \sum_{v\in V}\sum_{i=1}^{N}\mathbb{1}_{v=w_i} = N \tag{23}$$

$$\beta_k = \frac{\sum_{i=1}^{N}\mathbb{1}_{k=w_i}}{\sum_{v\in V}\sum_{i=1}^{N}\mathbb{1}_{v=w_i}} = \frac{1}{N}\sum_{i=1}^{N}\mathbb{1}_{k=w_i} \tag{24}$$

Projection



$$\|u\|_2 |\cos\theta| = \|u\|_2 \frac{|u^\top v|}{\|u\|_2 \|v\|_2} = \frac{|u^\top v|}{\|v\|_2} \tag{25}$$

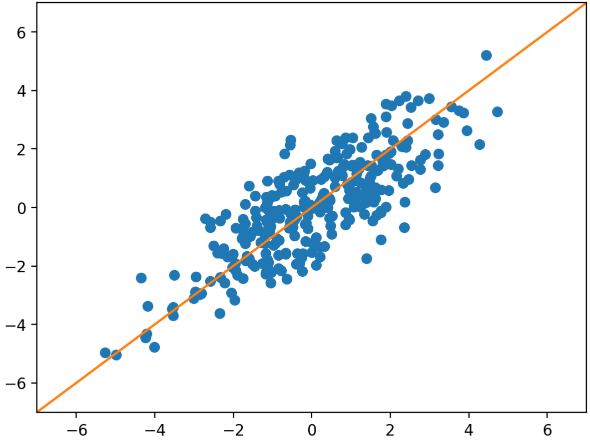
Projection

- The projection of x onto w is $\frac{|x^\top w|}{\|w\|_2}$.
- If we have N data points $\{x_1, \ldots, x_N\}$, then the sum of the (squared) projection is

$$\sum_{i=1}^{N} \left(\frac{|x_i^\top w|}{\|w\|_2} \right)^2 = \frac{w^\top X^\top X w}{w^\top w}. \tag{26}$$

- The sum of squared projection can be seen as the spread of the data.

Maximal projection



Maximal projection

- We want to find the maximum direction to project.
- The optimization problem is

$$\max_{w} \frac{w^\top X^\top X w}{w^\top w}. \tag{27}$$

Maximal projection

- The problem is scale invariant.

$$\frac{(aw)^\top X^\top X(aw)}{(aw)^\top (aw)} = \frac{w^\top X^\top X w}{w^\top w}. \tag{28}$$

- The problem is equivalent to

$$\max_w w^\top X^\top X w \qquad \text{s.t. } \|w\|_2^2 = 1. \tag{29}$$

Maximal projection

- The Lagrangian is

$$F = w^\top X^\top X w + \lambda(1 - \|w\|_2^2). \tag{30}$$

- Finding the optimal solution gives

$$\frac{\partial F}{\partial w} = (X^\top X + X^\top X)w - 2\lambda w = 0 \implies X^\top X w = \lambda w. \tag{31}$$

- It turns out that λ is an eigenvalue, and w an eigenvector.

Maximal projection

- Plugging the solution back to the objective,

$$\frac{w^\top X^\top X w}{w^\top w} = \frac{\lambda w^\top w}{w^\top w} = \lambda \tag{32}$$

- Since the goal is to find the maximal projection, this is now equivalent to finding the largest eigenvalue of $X^\top X$.

Maximal projection

- The term

$$\frac{w^\top X^\top X w}{w^\top w} \tag{33}$$

is called the Rayleigh quotient.

- The optimal w is called the first principal component.

- We will learn more about this when we talk about principal component analysis.