Announcements

- New homework out sometime tomorrow, due in two weeks
- Be a TA for CS/ORIE 1380: Data Science for All!
  - syllabus: https://classes.cornell.edu/browse/roster/SP18/class/ORIE/1380
  - apply: https://cis-student-hiring.coecis.cornell.edu/
Regularized empirical risk minimization

choose model by solving

$$\text{minimize} \quad \sum_{i=1}^{n} \ell(x_i, y_i; w) + r(w)$$

with variable $w \in \mathbb{R}^d$

- parameter vector $w \in \mathbb{R}^d$
- loss function $\ell : \mathcal{X} \times \mathcal{Y} \times \mathbb{R}^d \to \mathbb{R}$
- regularizer $r : \mathbb{R}^d \to \mathbb{R}$
Regularized empirical risk minimization

choose model by solving

\[
\text{minimize } \sum_{i=1}^{n} \ell(x_i, y_i; w) + r(w)
\]

with variable \( w \in \mathbb{R}^d \)

- parameter vector \( w \in \mathbb{R}^d \)
- loss function \( \ell : \mathcal{X} \times \mathcal{Y} \times \mathbb{R}^d \to \mathbb{R} \)
- regularizer \( r : \mathbb{R}^d \to \mathbb{R} \)

why?

- want to minimize the risk \( \mathbb{E}_{(x,y) \sim P} \ell(x, y; w) \)
- approximate it by the empirical risk \( \sum_{i=1}^{n} \ell(x, y; w) \)
- add regularizer to help model generalize
Solving regularized risk minimization

how should we fit these models?

▶ with a different software package for each model?
▶ with a different algorithm for each model?
▶ with a general purpose optimization solver?

desiderata

▶ fast
▶ flexible

we’ll use the **proximal gradient** method
Proximal operator

define the **proximal operator** of the function \( r : \mathbb{R}^d \rightarrow \mathbb{R} \)

\[
\text{prox}_r(z) = \arg\min_w (r(w) + \frac{1}{2}\|w - z\|_2^2)
\]
Proximal operator

define the **proximal operator** of the function $r : \mathbb{R}^d \rightarrow \mathbb{R}$

$$\text{prox}_r(z) = \arg\min_w (r(w) + \frac{1}{2}\|w - z\|^2)$$

$\text{prox}_r : \mathbb{R}^d \rightarrow \mathbb{R}^d$
Proximal operator

define the **proximal operator** of the function \( r : \mathbb{R}^d \to \mathbb{R} \)

\[
\text{prox}_r(z) = \arg\min_w (r(w) + \frac{1}{2}\|w - z\|_2^2)
\]

- \( \text{prox}_r : \mathbb{R}^d \to \mathbb{R}^d \)
- **generalized projection**: if \( 1_C \) is the indicator of set \( C \),
  \[
  \text{prox}_{1_C}(w) = \Pi_C(w)
  \]
**Proximal operator**

define the **proximal operator** of the function $r : \mathbb{R}^d \to \mathbb{R}$

$$\text{prox}_r(z) = \arg\min_{w} (r(w) + \frac{1}{2} \|w - z\|^2_2)$$

- **prox$_r : \mathbb{R}^d \to \mathbb{R}^d$**
- **generalized projection:** if $1_C$ is the indicator of set $C$,
  $$\text{prox}_{1_C}(w) = \Pi_C(w)$$

- **implicit gradient step:** if $w = \text{prox}_r(z)$ and $r$ is smooth,
  $$\nabla r(w) + w - z = 0$$
  $$w = z - \nabla r(w)$$
Proximal operator

define the **proximal operator** of the function \( r : \mathbb{R}^d \rightarrow \mathbb{R} \)

\[
\text{prox}_r(z) = \arg\min_w (r(w) + \frac{1}{2}\|w - z\|^2_2)
\]

- \( \text{prox}_r : R^d \rightarrow R^d \)
- **generalized projection:** if \( 1_C \) is the indicator of set \( C \),

\[
\text{prox}_{1_C}(w) = \Pi_C(w)
\]

- **implicit gradient step:** if \( w = \text{prox}_r(z) \) and \( r \) is smooth,

\[
\nabla r(w) + w - z = 0
\]

\[
\Rightarrow w = z - \nabla r(w)
\]

- **simple to evaluate:** closed form solutions for many functions

more info: [Parikh Boyd 2013]
Maps from functions to functions

no consistent notation for map from functions to functions.

for a function $f : \mathbb{R}^d \rightarrow \mathbb{R}$,

- \textbf{prox} maps $f$ to a new function $\text{prox}_f : \mathbb{R}^d \rightarrow \mathbb{R}^d$
  - $\text{prox}_f(x)$ evaluates this function at the point $x$

- $\nabla$ maps $f$ to a new function $\nabla f : \mathbb{R}^d \rightarrow \mathbb{R}^d$
  - $\nabla f(x)$ evaluates this function at the point $x$

- $\frac{\partial}{\partial x}$ maps $f$ to a new function $\frac{\partial f}{\partial x} : \mathbb{R}^d \rightarrow \mathbb{R}^d$
  - $\frac{\partial f}{\partial x}(x)|_{x=x}$ evaluates this function at the point $\bar{x}$
  - this one has the most confusing notation of all...
Maps from functions to functions

in a nice programming language, you can write something like

$$\text{prox}(f)(x)$$

or

$$\text{prox}(f, x)$$
Let’s evaluate some proximal operators!

define the \textbf{proximal operator} of the function \( r : \mathbb{R}^d \to \mathbb{R} \)

\[
\text{prox}_r(z) = \arg\min_w (r(w) + \frac{1}{2}\|w - z\|^2_2)
\]

\( r(w) = 0 \) (identity)

\( r(w) = \sum_{i=1}^d r_i(w_i) \) (separable)

\( r(w) = \|w\|_2^2 \) (shrinkage)

\( r(w) = \|w\|_1 \) (soft-thresholding)

\( r(w) = \mathbf{1}(w \geq 0) \) (projection)

\( r(w) = \sum_{i=1}^{d-1} (w_{i+1} - w_i)^2 \) (smoothing)
Proximal gradient method

want to solve

\[ \text{minimize } \ell(w) + r(w) \]

- \( \ell: \mathbb{R}^d \to \mathbb{R} \) smooth
- \( r: \mathbb{R}^d \to \mathbb{R} \) with a fast prox operator

proximal gradient method.

- pick step size sequence \( \{\alpha_t\}_{t=1}^{\infty} \) and \( w^0 \in \mathbb{R}^d \)
- repeat
  - \( w^{t+1} = \text{prox}_{\alpha_t r}(w^t - \alpha_t \nabla \ell(w^t)) \)
Proximal gradient method

want to solve

\[
\text{minimize } \ell(w) + r(w)
\]

▶ \( \ell : \mathbb{R}^d \to \mathbb{R} \) smooth
▶ \( r : \mathbb{R}^d \to \mathbb{R} \) with a fast prox operator

proximal gradient method.

▶ pick step size sequence \( \{\alpha_t\}_{t=1}^\infty \) and \( w^0 \in \mathbb{R}^d \)
▶ repeat
  ▶ \( w^{t+1} = \text{prox}_{\alpha_t r}(w^t - \alpha_t \nabla \ell(w^t)) \)

complexity:

▶ \( O(nd) \) to evaluate gradient of loss function
▶ \( O(d) \) to prox and to update
**Example: NNLS**

want to solve

$$\text{minimize } \frac{1}{2} \|y - Xw\|^2 + 1(w \geq 0)$$

recall

- $\nabla \left( \frac{1}{2} \|y - Xw\|^2 \right) = -X^T(y - Xw)$
- $\text{prox}_{1(\cdot \geq 0)}(w) = \max(0, w)$

**proximal gradient method.**

- pick step size sequence $\{\alpha_t\}_{t=0}^\infty$ and $w^0 \in \mathbb{R}^d$
- for $t = 0, 1, \ldots$
  - $w^{t+1} = \max(0, w^t + \alpha_t X^T(y - Xw^t))$
Example: NNLS

want to solve

$$\text{minimize } \frac{1}{2} \| y - Xw \|^2 + 1(w \geq 0)$$

recall

- \( \nabla \left( \frac{1}{2} \| y - Xw \|^2 \right) = -X^T(y - Xw) \)
- \( \text{prox}_{1(\cdot \geq 0)}(w) = \max(0, w) \)

proximal gradient method.

- pick step size sequence \( \{\alpha_t\}_{t=0}^{\infty} \) and \( w^0 \in \mathbb{R}^d \)
- for \( t = 0, 1, \ldots \)
  - \( w^{t+1} = \max(0, w^t + \alpha_t X^T(y - Xw^t)) \)

note: this is not the same as \( \max(0, (X^TX)^{-1}X^Ty) \)
Example: NNLS

proximal gradient method for NNLS.

- pick step size sequence \( \{\alpha_t\}_{t=0}^{\infty} \) and \( w^0 \in \mathbb{R}^d \)
- for \( t = 0, 1, \ldots \)
  - compute \( g^t = X^T(y - Xw^t) \) (\( \mathcal{O}(nd) \) flops)
  - \( w^{t+1} = \max(0, w^t + \alpha_t g^t) \) (\( \mathcal{O}(d) \) flops)

\( \mathcal{O}(nd) \) flops per iteration
Example: NNLS

option: do work up front to reduce per-iteration complexity

proximal gradient method for NNLS.

- pick step size sequence \( \{\alpha_t\}_{t=0}^\infty \) and \( w^0 \in \mathbb{R}^d \)
- form \( b = X^T y \) (\( \mathcal{O}(dn) \) flops), \( G = X^T X \) (\( \mathcal{O}(nd^2) \) flops)
- for \( t = 0, 1, \ldots \)
  - \( w^{t+1} = \max(0, w^t + \alpha_t(b - Gw^t)) \) (\( \mathcal{O}(d^2) \) flops)

\( \mathcal{O}(nd^2) \) flops to begin, \( \mathcal{O}(d^2) \) flops per iteration
Example: NNLS

**option:** do work up front to reduce per-iteration complexity

**proximal gradient method for NNLS.**

- pick step size sequence $\{\alpha_t\}_{t=0}^\infty$ and $w^0 \in \mathbb{R}^d$
- form $b = X^T y$ ($O(dn)$ flops), $G = X^T X$ ($O(nd^2)$ flops)
- for $t = 0, 1, \ldots$
  - $w^{t+1} = \max(0, w^t + \alpha_t(b - Gw^t))$ ($O(d^2)$ flops)

$O(nd^2)$ flops to begin, $O(d^2)$ flops per iteration

**note:** can compute $b$ and $G$ in parallel...
Example: Lasso

want to solve

\[
\text{minimize } \frac{1}{2} \| y - Xw \|^2 + \lambda \| w \|_1
\]

recall

\[
\nabla \left( \frac{1}{2} \| y - Xw \|^2 \right) = -X^T(y - Xw)
\]

\[
\text{prox}_{\mu \| \cdot \|_1}(w) = s_{\mu}(w) \text{ where}
\]

\[
(s_{\mu}(w))_i = \begin{cases}
  w_i - \mu & w_i \geq \mu \\
  0 & |w_i| \leq \mu \\
  w_i + \mu & w_i \leq -\mu
\end{cases}
\]

proximal gradient method.

\[
\nabla \left( \frac{1}{2} \| y - Xw \|^2 \right) = -X^T(y - Xw)
\]

\[
\text{prox}_{\mu \| \cdot \|_1}(w) = s_{\mu}(w) \text{ where}
\]

\[
(s_{\mu}(w))_i = \begin{cases}
  w_i - \mu & w_i \geq \mu \\
  0 & |w_i| \leq \mu \\
  w_i + \mu & w_i \leq -\mu
\end{cases}
\]

proximal gradient method.

\[
\text{pick step size sequence } \{\alpha_t\}_{t=0}^\infty \text{ and } w^0 \in \mathbb{R}^d
\]

\[
\text{form } b = X^T y \ (\mathcal{O}(dn) \text{ flops}), \ G = X^T X \ (\mathcal{O}(nd^2) \text{ flops})
\]

\[
\text{for } t = 0, 1, \ldots
\]

\[
\text{w}^{t+1} = s_{\alpha_t \lambda}(w^t + \alpha_t (b - Gw^t)) \ (\mathcal{O}(d^2) \text{ flops})
\]
Example: Lasso

want to solve

$$\text{minimize } \frac{1}{2} \|y - Xw\|^2 + \lambda \|w\|_1$$

recall

- $\nabla \left( \frac{1}{2} \|y - Xw\|^2 \right) = -X^T (y - Xw)$
- $\text{prox}_{\mu \|\cdot\|_1}(w) = s_\mu(w)$ where

$$\left(s_\mu(w)\right)_i = \begin{cases} w_i - \mu & w_i \geq \mu \\ 0 & |w_i| \leq \mu \\ w_i + \mu & w_i \leq -\mu \end{cases}$$

proximal gradient method.

- pick step size sequence $\{\alpha_t\}_{t=0}^\infty$ and $w^0 \in \mathbb{R}^d$
- form $b = X^T y$ ($\mathcal{O}(dn)$ flops), $G = X^T X$ ($\mathcal{O}(nd^2)$ flops)
- for $t = 0, 1, \ldots$
  - $w^{t+1} = s_{\alpha_t \lambda}(w^t + \alpha_t (b - Gw^t))$ ($\mathcal{O}(d^2)$ flops)

notice: the hard part ($\mathcal{O}(d^2)$) is computing the gradient...!
Convergence

two questions to ask:
  ▶ will the iteration ever stop?
  ▶ what kind of point will it stop at?

if the iteration stops, we say it has converged
Convergence: what kind of point will it stop at?

- let’s suppose $r$ is differentiable\(^1\)
- if we find $w$ so that

$$w = \text{prox}_{\alpha t r}(w - \alpha t \nabla \ell(w))$$

then

$$w = \arg\min_{w'} (\alpha t r(w') + \frac{1}{2} \|w' - (w - \alpha t \nabla \ell(w))\|^2_2)$$

$$0 = \nabla \alpha t r(w) + w - w + \alpha t \nabla \ell(w)$$

$$= \nabla (r(w) + \ell(w))$$

- so the gradient of the objective is 0
- if $\ell$ and $r$ are convex, that means $w$ minimizes $\ell + r$

\(^1\)take Convex Optimization for the proof for non-differentiable $r$
Convergence: will it stop?

definitions:

- \( p^* = \inf_w \ell(w) + r(w) \)

assumptions:

- loss function is continuously differentiable with \( L \)-Lipschitz derivative:
  \[
  \ell(w') \leq \ell(w) + \langle \nabla \ell(w), w' - w \rangle + \frac{L}{2} \| w' - w \|^2
  \]

- for simplicity, consider constant step size \( \alpha_t = \alpha \)
Proximal point method converges

prove it for $\ell = 0$ first (aka the proximal point method)
for any $t = 0, 1, \ldots$,

$$w^{t+1} = \arg\min_w \alpha r(w) + \frac{1}{2} \|w - w^t\|^2$$

so in particular,

$$\alpha r(w^{t+1}) + \frac{1}{2} \|w^{t+1} - w^t\|^2 \leq \alpha r(w^t) + \frac{1}{2} \|w^t - w^t\|^2$$

$$\frac{1}{2\alpha} \|w^{t+1} - w^t\|^2 \leq r(w^t) - r(w^{t+1})$$

now add up these inequalities for $t = 0, 1, \ldots, T$:

$$\frac{1}{2\alpha} \sum_{t=0}^{T} \|w^{t+1} - w^t\|^2 \leq \sum_{t=0}^{T} (r(w^t) - r(w^{t+1})) \leq r(w^0) - p^*$$

it converges!
Proximal gradient method converges (I)

now prove it for \( l \neq 0 \). for any \( t = 0, 1, \ldots \),

\[
\begin{align*}
    w^{t+1} &= \arg\min_w \alpha r(w) + \frac{1}{2} \| w - (w^t - \alpha \nabla \ell(w^t)) \|^2 \\
\end{align*}
\]

so in particular,

\[
\begin{align*}
    r(w^{t+1}) + \frac{1}{2\alpha} \| w^{t+1} - (w^t - \alpha \nabla \ell(w^t)) \|^2 \\
    &\leq r(w^t) + \frac{1}{2\alpha} \| w^t - (w^t - \alpha \nabla \ell(w^t)) \|^2 \\
    r(w^{t+1}) + \frac{1}{2\alpha} \| w^{t+1} - w^t \|^2 + \frac{\alpha}{2} \| \nabla \ell(w^t) \|^2 + \langle \nabla \ell(w^t), w^{t+1} - w^t \rangle \\
    &\leq r(w^t) + \frac{\alpha}{2} \| \nabla \ell(w^t) \|^2 \\
    r(w^{t+1}) + \frac{1}{2\alpha} \| w^{t+1} - w^t \|^2 + \langle \nabla \ell(w^t), w^{t+1} - w^t \rangle \\
    &\leq r(w^t)
\end{align*}
\]
Proximal gradient method converges (II)

now use $\ell(w') \leq \ell(w) + \langle \nabla \ell(w), w' - w \rangle + \frac{L}{2} \|w' - w\|^2$

with $w' = w^{t+1}$, $w = w^t$

\[
\ell(w^{t+1}) + r(w^{t+1}) + \frac{1}{2\alpha} \|w^{t+1} - w^t\|^2 + \langle \nabla \ell(w^t), w^{t+1} - w^t \rangle \\
\leq \ell(w^t) + r(w^t) + \langle \nabla \ell(w^t), w^{t+1} - w^t \rangle + \frac{L}{2} \|w^{t+1} - w^t\|^2
\]

\[
\ell(w^{t+1}) + r(w^{t+1}) + \frac{1}{2\alpha} \|w^{t+1} - w^t\|^2 \\
\leq \ell(w^t) + r(w^t) + \frac{L}{2} \|w^{t+1} - w^t\|^2
\]
Proximal gradient method converges (III)

now add up these inequalities for $t = 0, 1, \ldots, T$:

$$
\frac{1}{2} \left( \frac{1}{\alpha} - L \right) \sum_{t=0}^{T} \| w^{t+1} - w^t \|^2 \leq \sum_{t=0}^{T} \ell(w^t) + r(w^t) - \ell(w^{t+1}) + r(w^{t+1}) \\
\leq \ell(w^0) + r(w^0) - p^* 
$$

if

$$
\frac{1}{\alpha} - L \geq 0 \\
\implies \alpha \leq \frac{1}{L}
$$

it converges!
loss function is differentiable and $L$-Lipschitz:

$$\ell(w') \leq \ell(w) + \langle \nabla \ell(w), w' - w \rangle + \frac{L}{2} \|w' - w\|^2$$

what is $L$?
loss function is differentiable and $L$-Lipshitz:

$$\ell(w') \leq \ell(w) + \langle \nabla \ell(w), w' - w \rangle + \frac{L}{2} \|w' - w\|^2$$

what is $L$?

$$\ell(w') = \|Xw' - y\|^2$$

$$= \|X(w' - w) + Xw - y\|^2$$

$$= \|Xw - y\|^2 + 2(Xw - y)^T X(w' - w) + \|X(w' - w)\|^2$$

$$= \ell(w) + \langle \nabla \ell(w), w' - w \rangle + \|X(w' - w)\|^2$$

$$\leq \ell(w) + \langle \nabla \ell(w), w' - w \rangle + \|X\|^2 \|w' - w\|^2$$

so $L = 2\|X\|^2$, where $\|X\|$ is the maximum singular value of $X$
Speeding it up

- recall: computing the gradient is slow
- idea: approximate the gradient!

A stochastic gradient $\tilde{\nabla} \ell(w)$ is a random variable with

$$\mathbb{E} \tilde{\nabla} \ell(w) = \nabla \ell(w)$$
Stochastic gradient: examples

stochastic gradient obeys

$$\mathbb{E} \tilde{\nabla} \ell(w) = \nabla \ell(w)$$

**examples:** for $$\ell(w) = \sum_{i=1}^{n} (y_i - w^T x_i)^2$$,
Stochastic gradient: examples

stochastic gradient obeys

$$\mathbb{E} \tilde{\nabla} \ell(w) = \nabla \ell(w)$$

examples: for \( \ell(w) = \sum_{i=1}^{n} (y_i - w^T x_i)^2 \),

- single stochastic gradient. pick a random example \( i \). set

  $$\tilde{\nabla} \ell(w) = n \nabla (y_i - w^T x_i)^2 = -2n(y_i - w^T x_i) x_i$$
Stochastic gradient: examples

stochastic gradient obeys
\[ \mathbb{E} \tilde{\nabla} \ell(w) = \nabla \ell(w) \]

examples: for \( \ell(w) = \sum_{i=1}^{n} (y_i - w^T x_i)^2 \),

- **single stochastic gradient.** pick a random example \( i \). set
  \[ \tilde{\nabla} \ell(w) = n \nabla (y_i - w^T x_i)^2 = -2n(y_i - w^T x_i)x_i \]

- **minibatch stochastic gradient.** pick a random set of examples \( S \). set
  \[ \tilde{\nabla} \ell(w) = \frac{n}{|S|} \nabla \left( \sum_{i \in S} (y_i - w^T x_i)^2 \right) \]
  \[ = \frac{n}{|S|} \nabla \left( -2 \sum_{i \in S} (y_i - w^T x_i)x_i \right) \]
  (here \( |S| \) is the number of elements in the set \( S \).)
stochastic proximal gradient method.

- pick step size sequence \( \{\alpha_t\}_{t=1}^{\infty} \) and \( w^0 \in \mathbb{R}^d \)
- repeat
  - pick \( i \in \{1, \ldots, n\} \) uniformly at random
  - \( w^{t+1} = \text{prox}_{\alpha_t r}(w^t + \alpha_t 2n(y_i - w^T x_i)x_i) \) (\( O(d) \) flops)
Stochastic proximal gradient

stochastic proximal gradient method.

- pick step size sequence \( \{\alpha_t\}_{t=1}^{\infty} \) and \( w^0 \in \mathbb{R}^d \)
- repeat
  - pick \( i \in \{1, \ldots, n\} \) uniformly at random
  - \( w^{t+1} = \text{prox}_{\alpha_t r}(w^t + \alpha_t 2n(y_i - w^T x_i) x_i) \) \( (\mathcal{O}(d) \text{ flops}) \)

per iteration complexity: \( \mathcal{O}(d)! \)
Extensions

next lecture, we’ll see **loss functions** that are not differentiable can still use proximal gradient method!

need to define the **subgradient**
Subgradient

define the subgradient of a convex function $f : \mathbb{R}^d \to \mathbb{R}$ at $x$

$$\partial f(x) = \{g : \forall y, f(y) \geq f(x) + \langle g, y - x \rangle\}$$

- the subgradient $\partial f$ maps points to sets!
- follows the chain rule: if $f = h \circ g$, $h : \mathbb{R} \to \mathbb{R}$, and $g : \mathbb{R}^d \to \mathbb{R}$ is differentiable,

$$\partial f(x) = \partial h(g(x)) \nabla g(x)$$
Subgradient

for \( f : \mathbb{R} \to \mathbb{R} \) and convex, here’s a simpler equivalent condition:

- if \( f \) is differentiable at \( x \), \( \partial f(x) = \{ \nabla f(x) \} \)
- if \( f \) is not differentiable at \( x \), it will still be differentiable just to the left and the right of \( x^2 \), so
  - let \( g^+ = \lim_{\epsilon \to 0} \nabla f(x + \epsilon) \)
  - let \( g^- = \lim_{\epsilon \to 0} \nabla f(x - \epsilon) \)
  - \( \partial f(x) \) is any convex combination (i.e., any weighted average) of those gradients:

\[
\partial f(x) = \{ \alpha g^+ + (1 - \alpha)g^- : \alpha \in [0, 1] \}
\]

\(^2\) (because a convex function is differentiable almost everywhere)
proximal subgradient method.

- pick a step size sequence $\{\alpha_t\}_{t=1}^\infty$ and $w^0 \in \mathbb{R}^d$
- repeat
  - pick any $g \in \partial f(w^t)$
  - $w^{t+1} = \text{prox}_{\alpha_t r}(w^t - \alpha_t \nabla \ell(w^t))$
Convergence for stochastic proximal (sub)gradient

pick your poison:

- stochastic (sub)gradient, fixed step size $\alpha_t = \alpha$:
  - iterates converge quickly, then wander within a small ball
- stochastic (sub)gradient, decreasing step size $\alpha_t = 1/t$:
  - iterates converge slowly to solution
- minibatch stochastic (sub)gradient with increasing minibatch size, fixed step size $\alpha_t = \alpha$:
  - iterates converge quickly to solution
  - later iterations take (much) longer


conditions:

- $\ell$ is convex, subdifferentiable, Lipshitz continuous, and
- $r$ is convex and Lipshitz continuous where it is $< \infty$

or

- all iterates are bounded
Proximal subgradient method

Q: Why can’t we just use gradient descent to solve all our problems?

A: Because some regularizers and loss functions aren’t differentiable!

Q: Why can’t we just use subgradient descent to solve all our problems?

A: Because some of our regularizers don’t even have subgradients defined everywhere. (e.g., $1 + 1$)
Proximal subgradient method

**Q:** Why can’t we just use gradient descent to solve all our problems?

**A:** Because some regularizers and loss functions aren’t differentiable!

**Q:** Why can’t we just use subgradient descent to solve all our problems?

**A:** Because some of our regularizers don’t even have subgradients defined everywhere. (e.g., $1 + x$)
Proximal subgradient method

Q: Why can’t we just use gradient descent to solve all our problems?
A: Because some regularizers and loss functions aren’t differentiable!
Q: Why can’t we just use subgradient descent to solve all our problems?
Proximal subgradient method

Q: Why can’t we just use gradient descent to solve all our problems?
A: Because some regularizers and loss functions aren’t differentiable!

Q: Why can’t we just use subgradient descent to solve all our problems?
A: Because some of our regularizers don’t even have subgradients defined everywhere. (e.g., $1_+$)
References

- Vandenberghe: lecture on proximal gradient method.  
  http://www.seas.ucla.edu/~vandenbe/236C/lectures/proxgrad.pdf
- Yin: lecture on proximal method.  
  http://www.math.ucla.edu/~wotaoyin/summer2013/slides/Lec05_ProximalOperatorDual.pdf
- Bertsekas: convergence proofs for every proximal gradient style method you can dream of.  