COS324: Introduction to Machine Learning
Lecture 10: Gradient Methods in Machine Learning

Prof. Elad Hazan & Prof. Yoram Singer
Recap & Today

- Reminder of convexity, GD, and SGD

- Linear regression
  1. Problem definition
  2. Direct solution
  3. SGD for linear regression

- Binary classification
  1. Surrogate losses
  2. Sub-gradients
  3. Perceptron revisited
  4. SGD for binary classification

- Beyond binary learning problems
Recap & Today

- Reminder of convexity, GD, and SGD
- Linear regression
Recap & Today

- Reminder of convexity, GD, and SGD
- Linear regression
  1. Problem definition
Recap & Today

- Reminder of convexity, GD, and SGD
- Linear regression
  1. Problem definition
  2. Direct solution
Recap & Today

• Reminder of convexity, GD, and SGD

• Linear regression
  1. Problem definition
  2. Direct solution
  3. SGD for linear regression
Recap & Today

- Reminder of convexity, GD, and SGD
- Linear regression
  1. Problem definition
  2. Direct solution
  3. SGD for linear regression
- Binary classification
Recap & Today

- Reminder of convexity, GD, and SGD

- Linear regression
  1. Problem definition
  2. Direct solution
  3. SGD for linear regression

- Binary classification
  1. Surrogate losses
Recap & Today

• Reminder of convexity, GD, and SGD

• Linear regression
  1. Problem definition
  2. Direct solution
  3. SGD for linear regression

• Binary classification
  1. Surrogate losses
  2. Sub-gradients
Recap & Today

- Reminder of convexity, GD, and SGD

- Linear regression
  1. Problem definition
  2. Direct solution
  3. SGD for linear regression

- Binary classification
  1. Surrogate losses
  2. Sub-gradients
  3. Perceptron revisited
Recap & Today

- Reminder of convexity, GD, and SGD

- Linear regression
  1. Problem definition
  2. Direct solution
  3. SGD for linear regression

- Binary classification
  1. Surrogate losses
  2. Sub-gradients
  3. Perceptron revisited
  4. SGD for binary classification
Recap & Today

- Reminder of convexity, GD, and SGD

- Linear regression
  1. Problem definition
  2. Direct solution
  3. SGD for linear regression

- Binary classification
  1. Surrogate losses
  2. Sub-gradients
  3. Perceptron revisited
  4. SGD for binary classification

- Beyond binary learning problems
Convex Sets

$\Omega$ is convex set: $\forall \mathbf{u}, \mathbf{v} \in \Omega$, line segment between $\mathbf{u}$ and $\mathbf{v}$ is in $\Omega$

$$\forall \alpha \in [0, 1] \quad \alpha \mathbf{u} + (1 - \alpha) \mathbf{v} \in \Omega$$
Function $f : \Omega \to \mathbb{R}$ is convex if $\forall \mathbf{u}, \mathbf{v} \in \mathbb{C}$ and $\alpha \in [0, 1]$, 

$$f(\alpha \mathbf{u} + (1 - \alpha) \mathbf{v}) \leq \alpha f(\mathbf{u}) + (1 - \alpha) f(\mathbf{v})$$
Tangents Lie Below $f$

Gradient of $f$ at $\mathbf{w}$:
\[
\nabla f(\mathbf{w}) = \left( \frac{\partial f(\mathbf{w})}{\partial w_1}, \ldots, \frac{\partial f(\mathbf{w})}{\partial w_d} \right)
\]

If $f$ is convex and differentiable, then
\[
\forall \mathbf{u}, \quad f(\mathbf{u}) \geq f(\mathbf{w}) + \nabla f(\mathbf{w}) \cdot (\mathbf{u} - \mathbf{w})
\]
Convex optimization,
\[
\min_{w \in \Omega} f(w)
\]
where \(f\) is a convex function and \(\Omega\) is a convex set

C.O. for Machine learning,
\[
f(w) = \frac{1}{m} \sum_{i=1}^{m} \ell(w, (x_i, y_i))
\]
where \(\ell()\) is a convex loss function in \(w\) and assume \(\Omega = \mathbb{R}^d\)

Often abbreviate \(f_i(w) \overset{\text{def}}{=} \ell(w, (x_i, y_i))\) or \(\ell_i(w) \overset{\text{def}}{=} \ell(w, (x_i, y_i))\)
Gradient Descent

- Initialize $w^1$ (typically $w^1 = 0$)
Gradient Descent

- Initialize $w^1$ (typically $w^1 = 0$)
- For $t = 1, \ldots, T$ :
  
  
  \[
  w_{t+1} = w_t - \alpha \nabla f(w_t)
  \]

- Output $\bar{w}_T = \frac{1}{T} \sum_{t=1}^{T} w_t$
Gradient Descent

- Initialize $w^1$ (typically $w^1 = 0$)

- For $t = 1, \ldots, T$:
  - Set learning-rate $\eta^t$ (often fixed)
Gradient Descent

- Initialize $w^1$ (typically $w^1 = 0$)

- For $t = 1, \ldots, T$:
  - Set learning-rate $\eta^t$ (often fixed)
  - Perform gradient descent step:
    \[
    w^{t+1} = w^t - \eta^t \nabla f(w^t) = w^t - \eta^t \frac{1}{|S|} \sum_{i \in S} \nabla f_i(w^t)
    \]
Gradient Descent

- Initialize $w^1$ (typically $w^1 = 0$)

- For $t = 1, \ldots, T$
  - Set learning-rate $\eta^t$ (often fixed)
  - Perform gradient descent step:

$$w^{t+1} = w^t - \eta^t \nabla f(w^t)$$

$$= w^t - \eta^t \frac{1}{|S|} \sum_{i \in S} \nabla f_i(w^t)$$

- Output $\bar{w}^T = \frac{1}{T} \sum_{t=1}^{T} w^t$
Gradient Descent - Properties

- Assume or constrain $||\mathbf{w}|| \leq D/2$ therefore

$$\Rightarrow \quad ||\mathbf{w}^t - \mathbf{w}^*|| \leq ||\mathbf{w}^t|| + ||\mathbf{w}^*|| \leq D$$
Gradient Descent - Properties

- Assume or constrain $\|\mathbf{w}\| \leq D/2$ therefore

  $\Rightarrow \|\mathbf{w}^t - \mathbf{w}^*\| \leq \|\mathbf{w}^t\| + \|\mathbf{w}^*\| \leq D$

- Assume $\|\nabla f(\mathbf{w}^t)\| \leq G$
Gradient Descent - Properties

- Assume or constrain $\|w\| \leq D/2$ therefore

$$\Rightarrow \|w^t - w^*\| \leq \|w^t\| + \|w^*\| \leq D$$

- Assume $\|\nabla f(w^t)\| \leq G$

- Convergence rate of GD:

$$f(\bar{w}^T) - f(w^*) \leq \frac{DG}{\sqrt{T}}$$
Gradient Descent - Properties

- Assume or constrain $\|\mathbf{w}\| \leq D/2$ therefore

$$\Rightarrow \|\mathbf{w}^t - \mathbf{w}^*\| \leq \|\mathbf{w}^t\| + \|\mathbf{w}^*\| \leq D$$

- Assume $\|\nabla f(\mathbf{w}^t)\| \leq G$

- Convergence rate of GD:

$$f(\bar{\mathbf{w}}^T) - f(\mathbf{w}^*) \leq \frac{DG}{\sqrt{T}}$$

- However, each iteration requires $O(dm)$ operations
  [$d$ – dimension, $m$ – number of examples]
Iterates of Gradient Descent
Stochastic Gradient Descent

- Initialize $w^1$ (typically $w^1 = 0$)
Stochastic Gradient Descent

- Initialize $w^1$ (typically $w^1 = 0$)

- For $t = 1, \ldots, T$:
  - Set learning-rate $\eta_t$ (typically decreasing)
  - Perform stochastic gradient descent step:
    - Choose $S^0 \rightarrow S$ at random
    - Update $w^{t+1} = w^t - \eta_t \hat{f}(w^t) = w^t - \eta_1 |S^0| \sum_{i \in S^0} r_f(w^t)$

- Output $\bar{w}_T = \frac{1}{T} \sum_{t=1}^{T} w^t$
Stochastic Gradient Descent

- Initialize $\mathbf{w}^1$ (typically $\mathbf{w}^1 = \mathbf{0}$)

- For $t = 1, \ldots, T$
  - Set learning-rate $\eta^t$ (typically decreasing)
Stochastic Gradient Descent

- Initialize $w^1$ (typically $w^1 = 0$)

- For $t = 1, \ldots, T$:
  - Set learning-rate $\eta^t$ (typically decreasing)
  - Perform stochastic gradient descent step:
    - Choose $S_0 \rightarrow S$ at random
    - Update $w_{t+1} = w_t - \eta^t \hat{f}(w_t) = w_t - \eta^t |S_0| \sum_{i \in S_0} r f_i(w_t)$

- Output $\bar{w}_T = \frac{1}{T} \sum_{t=1}^{T} w_t$
Stochastic Gradient Descent

- Initialize $w^1$ (typically $w^1 = 0$)

- For $t = 1, \ldots, T$:
  - Set learning-rate $\eta^t$ (typically decreasing)
  - Perform stochastic gradient descent step:
    - Choose $S' \subset S$ at random

Output $\bar{w}_T = \frac{1}{T} \sum_{t=1}^{T} w^t$
Stochastic Gradient Descent

- Initialize $\mathbf{w}^1$ (typically $\mathbf{w}^1 = 0$)

- For $t = 1, \ldots, T$:
  - Set learning-rate $\eta^t$ (typically decreasing)
  - Perform stochastic gradient descent step:
    - Choose $S' \subset S$ at random
    - Update
      $$
      \mathbf{w}^{t+1} = \mathbf{w}^t - \eta^t \hat{\nabla f}(\mathbf{w}^t)
      $$
      $$
      = \mathbf{w}^t - \eta^t \frac{1}{|S'|} \sum_{i \in S'} \nabla f_i(\mathbf{w}^t)
      $$
Stochastic Gradient Descent

• Initialize $\mathbf{w}^1$ (typically $\mathbf{w}^1 = \mathbf{0}$)

• For $t = 1, \ldots, T$ :
  • Set learning-rate $\eta^t$ (typically decreasing)
  • Perform stochastic gradient descent step:
    • Choose $S' \subset S$ at random
    • Update

$$
\mathbf{w}^{t+1} = \mathbf{w}^t - \eta^t \nabla \hat{f}(\mathbf{w}^t)
= \mathbf{w}^t - \eta^t \frac{1}{|S'|} \sum_{i \in S'} \nabla f_i(\mathbf{w}^t)
$$

• Output $\bar{\mathbf{w}}^T = \frac{1}{T} \sum_{t=1}^T \mathbf{w}^t$
Stochastic Gradient Descent - Properties

- Assume that

\[ \forall i : \| \nabla f_i(w^t) \| \leq G \]

in contrast to GD, \[ \| \nabla f(w^t) \| \leq G \]
Assume that
\[ \forall i : \| \nabla f_i(w^t) \| \leq G \]
in contrast to GD, \[ \| \nabla f(w^t) \| \leq G \]

Convergence rate of GD:
\[ \mathbb{E}[f(\bar{w}^T)] - f(w^*) \leq \frac{DG}{\sqrt{T}} \]
Stochastic Gradient Descent - Properties

- Assume that
  \[ \forall i : \| \nabla f_i(w^t) \| \leq G \]
  in contrast to GD, \( \| \nabla f(w^t) \| \leq G \)

- Convergence rate of GD:
  \[
  \mathbb{E}[f(\bar{w}^T)] - f(w^*) \leq \frac{DG}{\sqrt{T}}
  \]

- Each iteration requires \( O(dc) \) operations, \( c \) is sub-sample size
Iterates of SGD

\* \( f(w^t) \)

\bullet \( f \left( \frac{1}{t} \sum_{s \leq t} w^s \right) \)
Regression Problems

- Automatic Kelly Blue Book: value assessment of used cars

<table>
<thead>
<tr>
<th>Year</th>
<th>Acci</th>
<th>Make</th>
<th>Mile</th>
<th>Ownr</th>
<th>Las$</th>
<th>Cur$</th>
</tr>
</thead>
<tbody>
<tr>
<td>97</td>
<td>5</td>
<td>To</td>
<td>120</td>
<td>3</td>
<td>2.5</td>
<td>0.5</td>
</tr>
<tr>
<td>16</td>
<td>1</td>
<td>Te</td>
<td>17</td>
<td>0</td>
<td>80</td>
<td>60</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td>Su</td>
<td>43</td>
<td>1</td>
<td>29</td>
<td>22</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$x_1$</td>
<td>$x_2$</td>
<td>$x_3$</td>
<td>$x_4$</td>
<td>$x_5$</td>
<td>$x_6$</td>
<td>$y$</td>
</tr>
</tbody>
</table>
## Regression Problems

- Automatic Kelly Blue Book: value assessment of used cars
- Collect sale information of cars: sold for $$

<table>
<thead>
<tr>
<th>Year</th>
<th>Acci</th>
<th>Make</th>
<th>Mile</th>
<th>Ownr</th>
<th>Las$</th>
<th>Cur$</th>
</tr>
</thead>
<tbody>
<tr>
<td>97</td>
<td>5</td>
<td>To</td>
<td>120</td>
<td>3</td>
<td>2.5</td>
<td>0.5</td>
</tr>
<tr>
<td>16</td>
<td>1</td>
<td>Te</td>
<td>17</td>
<td>0</td>
<td>80</td>
<td>60</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td>Su</td>
<td>43</td>
<td>1</td>
<td>29</td>
<td>22</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$x_1$</td>
<td>$x_2$</td>
<td>$x_3$</td>
<td>$x_4$</td>
<td>$x_5$</td>
<td>$x_6$</td>
<td>$y$</td>
</tr>
</tbody>
</table>

- How to represent symbolic features (Toyota, Tesla, Subaru)?
- How to represent ordered sets (#accidents: 0 < 1 < 2 < ...)?
- How to represent numeric features ($v$, $\log(v)$, $\log(v^b)$)?
Regression Problems

- Automatic Kelly Blue Book: value assessment of used cars
- Collect sale information of cars: sold for $\$
- For each car gather model year, # accidents, make, mileage # of previous owners, last sold for $\$, ...

<table>
<thead>
<tr>
<th>Year</th>
<th>Acci</th>
<th>Make</th>
<th>Mile</th>
<th>Ownr</th>
<th>Las$</th>
<th>Cur$</th>
</tr>
</thead>
<tbody>
<tr>
<td>97</td>
<td>5</td>
<td>To</td>
<td>120</td>
<td>3</td>
<td>2.5</td>
<td>0.5</td>
</tr>
<tr>
<td>16</td>
<td>1</td>
<td>Te</td>
<td>17</td>
<td>0</td>
<td>80</td>
<td>60</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td>Su</td>
<td>43</td>
<td>1</td>
<td>29</td>
<td>22</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$x_1$</td>
<td>$x_2$</td>
<td>$x_3$</td>
<td>$x_4$</td>
<td>$x_5$</td>
<td>$x_6$</td>
<td>$y$</td>
</tr>
</tbody>
</table>
Regression Problems

- Automatic Kelly Blue Book: value assessment of used cars
- Collect sale information of cars: sold for $$
- For each car gather model year, # accidents, make, mileage # of previous owners, last sold for $$$$, ...

<table>
<thead>
<tr>
<th>Year</th>
<th>Acci</th>
<th>Make</th>
<th>Mile</th>
<th>Ownr</th>
<th>Las$</th>
<th>Cur$</th>
</tr>
</thead>
<tbody>
<tr>
<td>97</td>
<td>5</td>
<td>To</td>
<td>120</td>
<td>3</td>
<td>2.5</td>
<td>0.5</td>
</tr>
<tr>
<td>16</td>
<td>1</td>
<td>Te</td>
<td>17</td>
<td>0</td>
<td>80</td>
<td>60</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td>Su</td>
<td>43</td>
<td>1</td>
<td>29</td>
<td>22</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$x_1$</td>
<td>$x_2$</td>
<td>$x_3$</td>
<td>$x_4$</td>
<td>$x_5$</td>
<td>$x_6$</td>
<td>$y$</td>
</tr>
</tbody>
</table>

- How to represent symbolic features (Toyota, Tesla, Subaru)?
Regression Problems

- Automatic Kelly Blue Book: value assessment of used cars
- Collect sale information of cars: sold for $\$
- For each car gather model year, # accidents, make, mileage # of previous owners, last sold for $\$$\$$, ...

<table>
<thead>
<tr>
<th>Year</th>
<th>Acci</th>
<th>Make</th>
<th>Mile</th>
<th>Ownr</th>
<th>Las$</th>
<th>Cur$</th>
</tr>
</thead>
<tbody>
<tr>
<td>97</td>
<td>5</td>
<td>To</td>
<td>120</td>
<td>3</td>
<td>2.5</td>
<td>0.5</td>
</tr>
<tr>
<td>16</td>
<td>1</td>
<td>Te</td>
<td>17</td>
<td>0</td>
<td>80</td>
<td>60</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td>Su</td>
<td>43</td>
<td>1</td>
<td>29</td>
<td>22</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$x_1$</td>
<td>$x_2$</td>
<td>$x_3$</td>
<td>$x_4$</td>
<td>$x_5$</td>
<td>$x_6$</td>
<td>$y$</td>
</tr>
</tbody>
</table>

- How to represent symbolic features (Toyota, Tesla, Subaru) ?
- How to represent ordered sets (#accidents: 0 < 1 < 2 < ...) ?
Regression Problems

- Automatic Kelly Blue Book: value assessment of used cars
- Collect sale information of cars: sold for $$
- For each car gather model year, # accidents, make, mileage # of previous owners, last sold for $$\$, ...

<table>
<thead>
<tr>
<th>Year</th>
<th>Acci</th>
<th>Make</th>
<th>Mile</th>
<th>Ownr</th>
<th>Las$</th>
<th>Cur$</th>
</tr>
</thead>
<tbody>
<tr>
<td>97</td>
<td>5</td>
<td>To</td>
<td>120</td>
<td>3</td>
<td>2.5</td>
<td>0.5</td>
</tr>
<tr>
<td>16</td>
<td>1</td>
<td>Te</td>
<td>17</td>
<td>0</td>
<td>80</td>
<td>60</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td>Su</td>
<td>43</td>
<td>1</td>
<td>29</td>
<td>22</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$x_1$</td>
<td>$x_2$</td>
<td>$x_3$</td>
<td>$x_4$</td>
<td>$x_5$</td>
<td>$x_6$</td>
<td>$y$</td>
</tr>
</tbody>
</table>

- How to represent symbolic features (Toyota, Tesla, Subaru)?
- How to represent ordered sets (#accidents: $0 < 1 < 2 < ...$)?
- How to represent numeric features ($v$, $\log(v)$, $\log(v) - b$)?
Linear Regression

- Each row is an example $\mathbf{x}_i \in \mathbb{R}^d$
Linear Regression

- Each row is an example $\mathbf{x}_i \in \mathbb{R}^d$
- Last column is a target $y_i \in \mathbb{R}$
Linear Regression

- Each row is an example $\mathbf{x}_i \in \mathbb{R}^d$
- Last column is a target $y_i \in \mathbb{R}$
- Create $m \times d$ matrix s.t. $X_{i,j}$ is $j$'th entry of $\mathbf{x}_i$
Linear Regression

- Each row is an example $\mathbf{x}_i \in \mathbb{R}^d$
- Last column is a target $y_i \in \mathbb{R}$
- Create $m \times d$ matrix s.t. $X_{i,j}$ is $j$'th entry of $\mathbf{x}_i$
- Create column vector $\mathbf{y}$ from $y_1, \ldots, y_n$
Linear Regression

- Each row is an example \( x_i \in \mathbb{R}^d \)
- Last column is a target \( y_i \in \mathbb{R} \)
- Create \( m \times d \) matrix s.t. \( X_{i,j} \) is \( j \)'th entry of \( x_i \)
- Create column vector \( y \) from \( y_1, \ldots, y_n \)
- Find a solution for the linear set of equations \( Xw = y \)
Linear Regression

- Each row is an example $\mathbf{x}_i \in \mathbb{R}^d$
- Last column is a target $y_i \in \mathbb{R}$
- Create $m \times d$ matrix s.t. $X_{i,j}$ is $j$’th entry of $\mathbf{x}_i$
- Create column vector $\mathbf{y}$ from $y_1, \ldots, y_n$
- Find a solution for the linear set of equations $X\mathbf{w} = \mathbf{y}$
  - Solution may not exist
Linear Regression

- Each row is an example $\mathbf{x}_i \in \mathbb{R}^d$
- Last column is a target $y_i \in \mathbb{R}$
- Create $m \times d$ matrix s.t. $X_{i,j}$ is $j$'th entry of $\mathbf{x}_i$
- Create column vector $\mathbf{y}$ from $y_1, \ldots, y_n$
- Find a solution for the linear set of equations $X\mathbf{w} = \mathbf{y}$
  - Solution may not exist
  - Multiple solutions may exist

- Complexity $\mathcal{O}(md + d^3)$
- Approximately solve, $X\mathbf{w} \approx \mathbf{y}$ namely $\forall i$:
  $\mathbf{w} \cdot \mathbf{x}_i \approx y_i$
Linear Regression

- Each row is an example $\mathbf{x}_i \in \mathbb{R}^d$
- Last column is a target $y_i \in \mathbb{R}$
- Create $m \times d$ matrix s.t. $X_{i,j}$ is j’th entry of $\mathbf{x}_i$
- Create column vector $\mathbf{y}$ from $y_1, \ldots, y_n$
- Find a solution for the linear set of equations $X\mathbf{w} = \mathbf{y}$
  - Solution may not exist
  - Multiple solutions may exist
  - Complexity $O(md + d^3)$
Linear Regression

- Each row is an example $\mathbf{x}_i \in \mathbb{R}^d$
- Last column is a target $y_i \in \mathbb{R}$
- Create $m \times d$ matrix s.t. $X_{i,j}$ is $j$’th entry of $\mathbf{x}_i$
- Create column vector $\mathbf{y}$ from $y_1, \ldots, y_n$
- Find a solution for the linear set of equations $X \mathbf{w} = \mathbf{y}$
  - Solution may not exist
  - Multiple solutions may exist
  - Complexity $O(md + d^3)$
- Approximately solve, $X \mathbf{w} \approx \mathbf{y}$ namely $\forall i : \mathbf{w} \cdot \mathbf{x}_i \approx y_i$
Linear Regression

- Each row is an example \( \mathbf{x}_i \in \mathbb{R}^d \)
- Last column is a target \( y_i \in \mathbb{R} \)
- Create \( m \times d \) matrix s.t. \( X_{i,j} \) is \( j \)'th entry of \( \mathbf{x}_i \)
- Create column vector \( \mathbf{y} \) from \( y_1, \ldots, y_n \)
- Find a solution for the linear set of equations \( X\mathbf{w} = \mathbf{y} \)
  - Solution may not exist
  - Multiple solutions may exist
  - Complexity \( O(md + d^3) \)
- Approximately solve, \( X\mathbf{w} \approx \mathbf{y} \) namely \( \forall i : \mathbf{w} \cdot \mathbf{x}_i \approx y_i \)
- Notion of \( \approx \) ?
Regression Losses

- Convex loss $\ell : \mathbb{R} \to \mathbb{R}_+$ ; $\ell(z) = \ell(w \cdot x - y)$
Regression Losses

- Convex loss $\ell : \mathbb{R} \rightarrow \mathbb{R}_+$ ; $\ell(z) = \ell(w \cdot x - y)$

- Example $i$ induces convex loss

$$\ell_i(w) = \ell(w \cdot x_i - y_i)$$
Regression Losses

- Convex loss $\ell : \mathbb{R} \rightarrow \mathbb{R}_+$ ; $\ell(z) = \ell(w \cdot x - y)$

- Example $i$ induces convex loss

  $$\ell_i(w) = \ell(w \cdot x_i - y_i)$$

- Total loss:

  $$f(w) = \frac{1}{m} \sum_{i=1}^{m} \ell(w \cdot x_i - y_i)$$
Regression Losses

- Convex loss $\ell : \mathbb{R} \rightarrow \mathbb{R}_+$ : $\ell(z) = \ell(w \cdot x - y)$

- Example $i$ induces convex loss

$$\ell_i(w) = \ell(w \cdot x_i - y_i)$$

- Total loss:

$$f(w) = \frac{1}{m} \sum_{i=1}^{m} \ell(w \cdot x_i - y_i)$$

- Concrete losses $\ell(z) = \ldots$

$$z^2 \quad |z| \quad z^4 \quad \ldots \quad \min\{|z| - \gamma, 0\} \quad \exp(z) + \exp(-z)$$
Least Squares Regression $\ell(z) = \frac{1}{2}z^2$

- Parameters: radius $D$, learning rate $\eta$, number of iterations $T$
Least Squares Regression $\ell(z) = \frac{1}{2} z^2$

- Parameters: radius $D$, learning rate $\eta$, number of iterations $T$
- Initialize: $\mathbf{w}^1 = \mathbf{0}$
Least Squares Regression $\ell(z) = \frac{1}{2}z^2$

- Parameters: radius $D$, learning rate $\eta$, number of iterations $T$
- Initialize: $w^1 = 0$
- For $t = 1, \ldots, T$:
Least Squares Regression $\ell(z) = \frac{1}{2} z^2$

- Parameters: radius $D$, learning rate $\eta$, number of iterations $T$
- Initialize: $w^1 = 0$
- For $t = 1, \ldots, T$ :
  - Choose $S' \subset S$ and calculate stochastic gradient
    \[ \nabla f(w^t) = \frac{1}{S'} \sum_{i \in S'} (w^t \cdot x_i - y_i) x_i \]
    \[ \equiv \Delta_i \]
Least Squares Regression $\ell(z) = \frac{1}{2} z^2$

- Parameters: radius $D$, learning rate $\eta$, number of iterations $T$
- Initialize: $w^1 = 0$
- For $t = 1, \ldots, T$:
  - Choose $S' \subset S$ and calculate stochastic gradient
    \[
    \nabla \hat{f}(w^t) = \frac{1}{S'} \sum_{i \in S'} (w^t \cdot x_i - y_i) x_i 
    \]
  - Update
    \[
    w^{t+\frac{1}{2}} = w^t - \eta^t \nabla \hat{f}(w^t)
    \]
    \[
    w^{t+1} = \min \left\{ 1, \frac{D}{\|w^{t+\frac{1}{2}}\|} \right\} w^{t+\frac{1}{2}}
    \]
Least Squares Regression $\ell(z) = \frac{1}{2}z^2$

- Parameters: radius $D$, learning rate $\eta$, number of iterations $T$
- Initialize: $w^1 = 0$
- For $t = 1, \ldots, T$
  - Choose $S' \subset S$ and calculate stochastic gradient
    \[
    \nabla \hat{f}(w^t) = \frac{1}{S'} \sum_{i \in S'} (w^t \cdot x_i - y_i) x_i \]
    \[
    \Delta_i \overset{\text{def}}{=} \hat{g}_i = w^t \cdot x_i
    \]
  - Update
    \[
    w^{t+\frac{1}{2}} = w^t - \eta^t \nabla \hat{f}(w^t)
    \]
    \[
    w^{t+1} = \min \left\{ 1, \frac{D}{\|w^{t+\frac{1}{2}}\|} \right\} w^{t+\frac{1}{2}}
    \]
- Output $\bar{w}^T = \frac{1}{T} \sum_{t=1}^{T} w^t$
\[ \Omega = \{ \omega \mid \| \omega \| \leq 0 \} \]

* From the picture it is "clear" that \( V = aU \) where \( a \in \mathbb{R}_+(0 < a < 1) \)

* Assume by contradiction that \( V = a\overrightarrow{U} + b\overrightarrow{q} \) where \( U + q \) (Why?)

* Then, \( \| U - V \|^2 = (1-a)^2\| U \|^2 + b\| q \|^2 \), but then all \( U \) is also in \( \Omega \) and has smaller distance to \( V \). Thus, \( b \) must be 0.
Pesky Learning Rate

- Recall that $\eta = \frac{D}{G\sqrt{T}}$ where

$$\| \nabla f(w^t) \| \leq G \quad \| w^t - w^* \| \leq D$$
Pesky Learning Rate

• Recall that \( \eta = \frac{D}{G\sqrt{T}} \) where

\[
\|\nabla f(w^t)\| \leq G \quad \|w^t - w^*\| \leq D
\]

• Assume or normalize such that \( \forall i : \|x_i\| \leq b \quad |y_i| \leq c \)
Pesky Learning Rate

- Recall that $\eta = \frac{D}{G\sqrt{T}}$ where 
  $$\|\nabla f(w^t)\| \leq G \quad \|w^t - w^*\| \leq D$$

- Assume or normalize such that $\forall i : \|x_i\| \leq b \quad |y_i| \leq c$

- Constrain $\forall t : \|w^t\| \leq D/2$

$\Rightarrow$ projection step
Pesky Learning Rate

- Recall that $\eta = \frac{D}{G\sqrt{T}}$ where $\|\nabla f(w^t)\| \leq G$, $\|w^t - w^*\| \leq D$

- Assume or normalize such that $\forall i : \|x_i\| \leq b$, $|y_i| \leq c$

- Constrain $\forall t : \|w^t\| \leq D/2$

- We thus get: $\|w^t - w^*\| \leq \|w^t\| + \|w^*\| \leq D$
Pesky Learning Rate

- Recall that $\eta = \frac{D}{G\sqrt{T}}$ where
  $$\|\nabla f(w^t)\| \leq G \quad \|w^t - w^*\| \leq D$$

- Assume or normalize such that $\forall i: \|x_i\| \leq b \ |y_i| \leq c$

- Constrain $\forall t: \|w^t\| \leq D/2$

- We thus get: $\|w^t - w^*\| \leq \|w^t\| + \|w^*\| \leq D$

- In addition, we get a bound on gradients,
  $$\|(w \cdot x_i - y_i)x_i\| \leq |w \cdot x_i - y_i|\|x_i\|$$
  $$\leq |w \cdot x_i - y_i|b$$
  $$\leq (|w \cdot x_i| + |y_i|)b$$
  $$\leq (Db + c)b$$

- In addition, we get a bound on gradients,
  $$\|(w \cdot x_i - y_i)x_i\| \leq |w \cdot x_i - y_i|\|x_i\|$$
  $$\leq |w \cdot x_i - y_i|b$$
  $$\leq (|w \cdot x_i| + |y_i|)b$$
  $$\leq (Db + c)b$$

- Hölder Ineq.
  $$W \cdot X \in \mathbb{R}^d$$
  $$W \cdot X \leq \|W\|_\infty \|X\|_1$$
  $$\|W\|_\infty \|X\|_1$$
  $[Cauchy–Schwarz]$
Pesky Learning Rate

- Recall that $\eta = \frac{D}{G\sqrt{T}}$ where
  $$\|\nabla f(w^t)\| \leq G \quad \|w^t - w^*\| \leq D$$

- Assume or normalize such that $\forall i : \|x_i\| \leq b \; |y_i| \leq c$

- Constrain $\forall t : \|w^t\| \leq D/2$

- We thus get: $\|w^t - w^*\| \leq \|w^t\| + \|w^*\| \leq D$

- In addition, we get a bound on gradients,
  $$\|(w \cdot x_i - y_i)x_i\| \leq |w \cdot x_i - y_i||x_i|$$
  $$\leq |w \cdot x_i - y_i|b$$
  $$\leq (|w \cdot x_i| + |y_i|)b$$
  $$\leq (Db + c)b \quad [Cauchy–Schwarz]$$

- And we can set $\eta = \frac{D}{(Db+c)b\sqrt{T}}$ ... but in practice ...
Binary Classification

- Examples $x_i \in \mathbb{R}^d$
Binary Classification

- Examples $x_i \in \mathbb{R}^d$
- Labels $y_i \in \{-1, +1\}$
Binary Classification

- Examples $x_i \in \mathbb{R}^d$
- Labels $y_i \in \{-1, +1\}$
- Predictor $h_w(x) = \text{sign}(w \cdot x)$
Binary Classification

- Examples $x_i \in \mathbb{R}^d$
- Labels $y_i \in \{-1, +1\}$
- Predictor $h_w(x) = \text{sign}(w \cdot x)$

Goal,

$$\min_w \frac{1}{m} \sum_{i=1}^{m} 1[\text{sign}(w \cdot x_i) \neq y_i]$$
Binary Classification

- Examples $x_i \in \mathbb{R}^d$
- Labels $y_i \in \{-1, +1\}$
- Predictor $h_w(x) = \text{sign}(w \cdot x)$
- Goal,
  \[
  \min_w \frac{1}{m} \sum_{i=1}^{m} \mathbb{1}[\text{sign}(w \cdot x_i) \neq y_i]
  \]
- First attempt: define $z = y(w \cdot x)$ and $\ell_{0-1}(z) = \mathbb{1}[z \leq 0]$
Binary Classification

- Examples $x_i \in \mathbb{R}^d$
- Labels $y_i \in \{-1, +1\}$
- Predictor $h_w(x) = \text{sign}(w \cdot x)$
- Goal,
  \[
  \min_w \frac{1}{m} \sum_{i=1}^{m} 1[\text{sign}(w \cdot x_i) \neq y_i]
  \]
- First attempt: define $z = y(w \cdot x)$ and $\ell_{0-1} (z) = 1[z \leq 0]$
- Can we use (stochastic) gradient descent?
0-1 Loss

“Utopia”: combinatorial problem which is NP-Hard
Surrogate Losses for Classification

- **Convex** losses w.r.t $z = y(w \cdot x)$ which satisfy
  \[ \ell(z) \geq \ell_{0-1}(z) \]
Surrogate Losses for Classification

- **Convex** losses w.r.t $z = y(w \cdot x)$ which satisfy
  \[ \ell(z) \geq \ell_{0-1}(z) \]
- Exp-loss,
  \[ \exp(-z) \]
Surrogate Losses for Classification

- **Convex** losses w.r.t $z = y(w \cdot x)$ which satisfy
  \[ \ell(z) \geq \ell_{0-1}(z) \]

- **Exp-loss**, 
  \[ \exp(-z) \]

- **Log-loss**, 
  \[ \log(1 + \exp(-z)) \]
Surrogate Losses for Classification

- **Convex** losses w.r.t $z = y(\mathbf{w} \cdot \mathbf{x})$ which satisfy
  \[ \ell(z) \geq \ell_{0-1}(z) \]
- Exp-loss,
  \[ \exp(-z) \]
- Log-loss,
  \[ \log(1 + \exp(-z)) \]
- Hinge-loss,
  \[ \max \{0, 1 - z\} = [1 - z]_+ \]
Surrogate Losses for Classification

- **Convex** losses w.r.t $z = y(w \cdot x)$ which satisfy
  \[\ell(z) \geq \ell_{0-1}(z)\]

- Exp-loss,
  \[\exp(-z)\]

- Log-loss,
  \[\log(1 + \exp(-z))\]

- Hinge-loss,
  \[\max\{0, 1 - z\} = [1 - z]_+\]

- Squared-error with $\Delta = w \cdot x - y$,
  \[\ell(\Delta) = \Delta^2 = (w \cdot x - y)^2\]
  \[= y^2(w \cdot x - y)^2\]
  \[= (y(w \cdot x) - 1)^2 \Rightarrow \ell(z) = (1 - z)^2\]
\[ y(w \cdot x) \]

\[ \|w\| = 1 \]

signed margin
Sub-gradients

- \( v \) is sub-gradient of \( f \) at \( w \) if \( \forall u, \quad f(u) \geq f(w) + v \cdot (u - w) \)
- The differential set, \( \partial f(w) \), is the set of sub-gradients of \( f \) at \( w \)
- Lemma: \( f \) is convex iff for every \( w \), \( \partial f(w) \neq \emptyset \)
Optimality Property

\( f \) is “locally flat” around \( \mathbf{w} \), i.e. \( \mathbf{0} \) is a sub-gradient, 
\[ \text{iff} \]
\( \mathbf{w} \) is a (not “the”) global minimizer

We can replace gradients with sub-gradients:

\[
\mathbf{w}^{t+1} = \mathbf{w}^t - \eta \mathbf{g}^t \quad \text{where} \quad \mathbf{g}^t \in \partial \hat{f}(\mathbf{w}^t)
\]
Hinge Loss

\[ \ell(z) = \max \{0, 1 - z\} = [1 - z]_+ \]

\[ \ell_{\text{hinge}}(w, (x, y)) \overset{\text{def}}{=} \max\{0, 1 - y(w \cdot x)\} \]

Non-differentiable at \( z = 1 \)

Can we use SGD?
SGD for Hinge-Loss

- Fully stochastic case – single example
SGD for Hinge-Loss

- Fully stochastic case – single example

- Subgradient of $[1 - z]_+$,

\[
\partial \ell(z) = \begin{cases} 
0 & z > 1 \\
-1 & z < 1 \\
(-1, 0) & z = 1 
\end{cases}
\]

\[
\partial \ell(w, (x, y)) = yx \partial \ell(z) \text{ where } z = y(w \cdot x)
\]
SGD for Hinge-Loss

- Fully stochastic case – single example

- Subgradient of \([1 - z]_+\),

\[
\partial \ell(z) = \begin{cases} 
0 & z > 1 \\
-1 & z < 1 \\
(-1, 0) & z = 1
\end{cases}
\]

\[
\partial \ell(w, (x, y)) = yx\partial \ell(z) \quad \text{where} \quad z = y(w \cdot x)
\]

- SGD update on iteration \(t\):

\[
w^{t+1} = w^t - \eta g^t \quad \text{where} \quad g^t \in \partial \ell_t(w^t)
\]

\[
w^{t+1} = \begin{cases} 
w^t + \eta y^t x^t & y^t(w^t \cdot x^t) \leq 1 \\
w^t & \text{otherwise}
\end{cases}
\]
SGD vs. Perceptron

- SGD

$$w^{t+1} = \begin{cases} w^t + \eta y^t x^t & y^t (w^t \cdot x^t) \leq 1 \\ w^t & \text{otherwise} \end{cases}$$
SGD vs. Perceptron

- SGD

\[ w^{t+1} = \begin{cases} 
w^t + \eta y^t x^t & y^t(w^t \cdot x^t) \leq 1 \\
w^t & \text{otherwise}
\end{cases} \]

- Perceptron

\[ w^{t+1} = \begin{cases} 
w^t + \eta y^t x^t & y^t(w^t \cdot x^t) \leq 0 \\
w^t & \text{otherwise}
\end{cases} \]
SGD $\approx$ Perceptron

- Analysis of SGD assumes,

$$\|\nabla \ell_t(w^t)\| \leq G \quad \|w^t - w^*\| \leq D$$
SGD ≈ Perceptron

- Analysis of SGD assumes,
  \[ \| \nabla \ell_t(w^t) \| \leq G \quad \| w^t - w^* \| \leq D \]

- Analysis of GD & SGD’s implies,
  \[ \sum_{t=1}^{T} [1 - y_t(w^t \cdot x_t)]_+ \leq \sum_{t=1}^{T} [1 - y_t(w^* \cdot x_t)]_+ + \sqrt{T} GD \]
**SGD ≈ Perceptron**

- Analysis of SGD assumes,
  \[
  \| \nabla \ell_t(w^t) \| \leq G \quad \| w^t - w^* \| \leq D
  \]
- Analysis of GD & SGD’s implies,
  \[
  \sum_{t=1}^{T} [1 - y_t(w^t \cdot x_t)]_+ \leq \sum_{t=1}^{T} [1 - y_t(w^* \cdot x_t)]_+ + \sqrt{T} GD
  \]
- Analysis of Perceptron assumes,
  \[
  \forall i : \| x_i \| \leq 1 \quad \exists w^* : \| w^* \| = 1 \wedge y_i(w^* \cdot x_i) \geq \gamma
  \]
SGD $\approx$ Perceptron

- Analysis of SGD assumes,
  \[ \| \nabla l_t(w^t) \| \leq G \quad \| w^t - w^* \| \leq D \]

- Analysis of GD & SGD’s implies,
  \[ \sum_{t=1}^{T} [1 - y_t(w^t \cdot x_t)]_+ \leq \sum_{t=1}^{T} [1 - y_t(w^* \cdot x_t)]_+ + \sqrt{T}GD \]

- Analysis of Perceptron assumes,
  \[ \forall i : \| x_i \| \leq 1 \quad \exists w^* : \| w^* \| = 1 \wedge y_i(w^* \cdot x_i) \geq \gamma \]

- Perceptron’s mistake bound is,
  \[ \frac{1}{\gamma^2} \Rightarrow \sum_{t=1}^{T} 1[y_t(w^t \cdot x_t) \leq 0] \leq \frac{1}{\gamma^2} \]
SGD ⇒ Perceptron

- Need to accommodate Perceptron’s assumptions,

\[ \forall i : \|x_i\| \leq 1 \ \exists w^* : \|w^*\| = 1 \land y_i(w^* \cdot x_i) \geq \gamma \]
SGD ⇒ Perceptron

- Need to accommodate Perceptron’s assumptions,

\[\forall i : \|x_i\| \leq 1 \ \exists \mathbf{w}^* : \|\mathbf{w}^*\| = 1 \ \land \ y_i(\mathbf{w}^* \cdot x_i) \geq \gamma\]

- Constraining (by projecting) \(\|\mathbf{w}^t\| \leq 1\) imply

\[\mathbf{w}^t \cdot x_i \leq \|\mathbf{w}^t\| \|x_i\| \leq 1\]
SGD ⇒ Perceptron

• Need to accommodate Perceptron’s assumptions,

\[ \forall i : \|x_i\| \leq 1 \ \exists w^* : \|w^*\| = 1 \land y_i(w^* \cdot x_i) \geq \gamma \]

• Constraining (by projecting) \( \|w^t\| \leq 1 \) imply

\[ w^t \cdot x_i \leq \|w^t\| \|x_i\| \leq 1 \]

• Modify loss to be \( [\gamma - y(w \cdot x)]_+ \)
SGD ⇒ Perceptron

- Need to accommodate Perceptron’s assumptions,

\[ \forall i : \|x_i\| \leq 1 \quad \exists w^* : \|w^*\| = 1 \land y_i(w^* \cdot x_i) \geq \gamma \]

- Constraining (by projecting) \( \|w^t\| \leq 1 \) imply

\[ w^t \cdot x_i \leq \|w^t\| \|x_i\| \leq 1 \]

- Modify loss to be \([\gamma - y(w \cdot x)]_+\)

- We start at \( w^1 = 0 \) & progress toward \( w^* \) thus

\[ \|w^t - w^*\| \leq 1 \]
SGD $\Rightarrow$ Perceptron

- Need to accommodate Perceptron’s assumptions,

\[ \forall i : \|x_i\| \leq 1 \ \exists w^* : \|w^*\| = 1 \land y_i(w^* \cdot x_i) \geq \gamma \]

- Constraining (by projecting) $\|w^t\| \leq 1$ imply

\[ w^t \cdot x_i \leq \|w^t\| \|x_i\| \leq 1 \]

- Modify loss to be $[\gamma - y(w \cdot x)]_+$

- We start at $w^1 = 0$ & progress toward $w^*$ thus

\[ \|w^t - w^*\| \leq 1 \]

- Since $\forall t : \|w^t\| \leq 1 \land \|x_i\| \leq 1$ then

\[ G \leq 1 \quad D \leq 1 \]
SGD $\Rightarrow$ Perceptron

- “Ignore” rounds $t$ such that $0 < y_t(w^t \cdot x^t) \leq \gamma$
SGD \Rightarrow \text{Perceptron}

- “Ignore” rounds $t$ such that $0 < y_t(w^t \cdot x^t) \leq \gamma$
- Loss bound becomes,

$$\gamma \sum_{t=1}^{T} \mathbb{1}[y_t(w^t \cdot x_t) \leq 0] \leq \sum_{t=1}^{T} [\gamma - y_t(w^t \cdot x_t)]_+$$

$$\leq \sum_{t=1}^{T} [\gamma - y_t(w^* \cdot x_t)]_+ + \sqrt{T}$$
SGD ⇒ Perceptron

- “Ignore” rounds $t$ such that $0 < y_t(w^t \cdot x^t) \leq \gamma$

- Loss bound becomes,

$$\gamma \sum_{t=1}^{T} 1[y_t(w^t \cdot x_t) \leq 0] \leq \sum_{t=1}^{T} [\gamma - y_t(w^t \cdot x_t)]_+$$

$$\leq \sum_{t=1}^{T} [\gamma - y_t(w^* \cdot x_t)]_{\leq \gamma} + \sqrt{T}$$

- If we saw only mistake-prone examples ⇒ $T = \# \text{mistakes}$

$$\gamma T \leq \sqrt{T} \implies T \leq \frac{1}{\gamma^2}$$
SGD ⇒ Perceptron

- “Ignore” rounds $t$ such that $0 < y_t(w^t \cdot x^t) \leq \gamma$

- Loss bound becomes,

$$\gamma \sum_{t=1}^{T} 1[y_t(w^t \cdot x_t) \leq 0] \leq \sum_{t=1}^{T} [\gamma - y_t(w^t \cdot x_t)]_+$$

$$\leq \sum_{t=1}^{T} [\gamma - y_t(w^* \cdot x_t)]_+ + \sqrt{T}$$

- If we saw only mistake-prone examples ⇒ $T = \#\text{mistakes}$

$$\gamma T \leq \sqrt{T} \Rightarrow T \leq \frac{1}{\gamma^2}$$

- SGD updates $w^t$ on rounds when $y_t(w^t \cdot x^t)$ is small and is thus called the aggressive Perceptron
Logistic Loss

- Define the following estimate,

\[ P[Y = +1|x, w] \overset{\text{def}}{=} \frac{1}{1 + \exp(-w \cdot x)} \]
Logistic Loss

- Define the following estimate,

\[
\mathbb{P} [Y = +1 | \mathbf{x}, \mathbf{w}] \overset{\text{def}}{=} \frac{1}{1 + \exp(-\mathbf{w} \cdot \mathbf{x})}
\]

- We can write,

\[
\mathbb{P} [Y = -1 | \mathbf{x}, \mathbf{w}] = 1 - \frac{1}{1 + \exp(-\mathbf{w} \cdot \mathbf{x})} = \frac{1}{1 + \exp(\mathbf{w} \cdot \mathbf{x})}
\]
Logistic Loss

• Define the following estimate,

\[ P[Y = +1|x, w] \overset{\text{def}}{=} \frac{1}{1 + \exp(-w \cdot x)} \]

• We can write,

\[ P[Y = -1|x, w] = 1 - \frac{1}{1 + \exp(-w \cdot x)} = \frac{1}{1 + \exp(w \cdot x)} \]

• Putting the two outcomes together we get,

\[ P[Y = y|x, w] \overset{\text{def}}{=} \frac{1}{1 + \exp(-y(w \cdot x))} \]
Logistic Loss

- Loss of wrong prediction,

\[- \log \left( \mathbb{P} [Y = -y_i | \mathbf{w}, \mathbf{x}_i] \right) = - \log \left( 1 + e^{-y_i (\mathbf{w} \cdot \mathbf{x}_i)} \right)\]
Logistic Loss

- Loss of wrong prediction,

\[- \log (\mathbb{P}[Y = -y_i | \mathbf{w}, \mathbf{x}_i]) = - \log \left(1 + e^{-y_i (\mathbf{w} \cdot \mathbf{x}_i)}\right)\]

- SGD iterate for sub-sample $S'$

\[
\forall i \in S' : \quad p_i = \frac{1}{1 + \exp(y_i (\mathbf{w}^t \cdot \mathbf{x}_i))}
\]

\[
\mathbf{g}^t = - \sum_{i \in S'} p_i y_i \mathbf{x}_i
\]

\[
\mathbf{w}^{t+1} = \mathbf{w}^t - \eta^t \mathbf{g}^t = \mathbf{w}^t + \eta^t \sum_{i \in S'} p_i y_i \mathbf{x}_i
\]
Generalization Revisited

- Algorithm $\mathcal{A}$ for surrogate loss $\ell(\cdot)$ that guarantees,

$$\mathcal{L}_D^\ell(\mathcal{A}(S)) \leq \min_{w \in \Omega} \mathcal{L}_D^\ell(w) + \varepsilon$$
Generalization Revisited

- Algorithm $\mathcal{A}$ for surrogate loss $\ell(\cdot)$ that guarantees,

$$\mathcal{L}_{D}^{\ell}(\mathcal{A}(S)) \leq \min_{w \in \Omega} \mathcal{L}_{D}^{\ell}(w) + \varepsilon$$

- Since $\ell(z) \geq \ell^{-1}(z)$,

$$\mathcal{L}_{D}^{\ell^{-1}}(\mathcal{A}(S)) \leq \mathcal{L}_{D}^{\ell}(\mathcal{A}(S)) \leq \min_{w \in \Omega} \mathcal{L}_{D}^{\ell}(w) + \varepsilon$$
Generalization Revisited

- Algorithm $\mathcal{A}$ for surrogate loss $\ell(\cdot)$ that guarantees,

$$\mathcal{L}_D^\ell(\mathcal{A}(S)) \leq \min_{w \in \Omega} \mathcal{L}_D^\ell(w) + \varepsilon$$

- Since $\ell(z) \geq \ell^{0-1}(z)$,

$$\mathcal{L}_D^{0-1}(\mathcal{A}(S)) \leq \mathcal{L}_D^\ell(\mathcal{A}(S)) \leq \min_{w \in \Omega} \mathcal{L}_D^\ell(w) + \varepsilon$$

- Decompose upper bound,

$$\mathcal{L}_D^{0-1}(\mathcal{A}(S)) \leq \min_{w \in \Omega} \mathcal{L}_D^{0-1}(w) + \left( \min_{w \in \Omega} \mathcal{L}_D^\ell(w) - \min_{w \in \Omega} \mathcal{L}_D^{0-1}(w) \right) + \varepsilon$$

$$= \varepsilon_{\text{approx}} + \varepsilon_{\text{optim}} + \varepsilon_{\text{estim}}$$
Generalization Revisited

• Algorithm $\mathcal{A}$ for surrogate loss $\ell(\cdot)$ that guarantees,

$$L^\ell_D(\mathcal{A}(S)) \leq \min_{w \in \Omega} L^\ell_D(w) + \epsilon$$

• Since $\ell(z) \geq \ell^{0-1}(z)$,

$$L^{0-1}_D(\mathcal{A}(S)) \leq L^\ell_D(\mathcal{A}(S)) \leq \min_{w \in \Omega} L^\ell_D(w) + \epsilon$$

• Decompose upper bound,

$$L^{0-1}_D(\mathcal{A}(S)) \leq \min_{w \in \Omega} L^{0-1}_D(w) + \left( \min_{w \in \Omega} L^\ell_D(w) - \min_{w \in \Omega} L^{0-1}_D(w) \right) + \epsilon = \epsilon_{\text{approx}} + \epsilon_{\text{optim}} + \epsilon_{\text{estim}}$$

• $\epsilon_{\text{approx}}$ approximation error due to domain $\Omega$
Generalization Revisited

- Algorithm $\mathcal{A}$ for surrogate loss $\ell(\cdot)$ that guarantees,

$$\mathcal{L}_D^\ell(\mathcal{A}(S)) \leq \min_{w \in \Omega} \mathcal{L}_D^\ell(w) + \varepsilon$$

- Since $\ell(z) \geq \ell^{0-1}(z)$,

$$\mathcal{L}_D^{0-1}(\mathcal{A}(S)) \leq \mathcal{L}_D^\ell(\mathcal{A}(S)) \leq \min_{w \in \Omega} \mathcal{L}_D^\ell(w) + \varepsilon$$

- Decompose upper bound,

$$\mathcal{L}_D^{0-1}(\mathcal{A}(S)) \leq \min_{w \in \Omega} \mathcal{L}_D^{0-1}(w) + \left( \min_{w \in \Omega} \mathcal{L}_D^\ell(w) - \min_{w \in \Omega} \mathcal{L}_D^{0-1}(w) \right) + \varepsilon$$

$$= \varepsilon_{\text{approx}} + \varepsilon_{\text{optim}} + \varepsilon_{\text{estim}}$$

- $\varepsilon_{\text{approx}}$ approximation error due to domain $\Omega$
- $\varepsilon_{\text{optim}}$ optimization error due to surrogate $\ell$
Generalization Revisited

- Algorithm $\mathcal{A}$ for surrogate loss $\ell(\cdot)$ that guarantees,

$$\mathcal{L}_D^\ell(\mathcal{A}(S)) \leq \min_{w \in \Omega} \mathcal{L}_D^\ell(w) + \varepsilon$$

- Since $\ell(z) \geq \ell^{0^{-1}}(z)$,

$$\mathcal{L}_D^{0^{-1}}(\mathcal{A}(S)) \leq \mathcal{L}_D^\ell(\mathcal{A}(S)) \leq \min_{w \in \Omega} \mathcal{L}_D^\ell(w) + \varepsilon$$

- Decompose upper bound,

$$\mathcal{L}_D^{0^{-1}}(\mathcal{A}(S)) \leq \min_{w \in \Omega} \mathcal{L}_D^{0^{-1}}(w) + \left(\min_{w \in \Omega} \mathcal{L}_D^\ell(w) - \min_{w \in \Omega} \mathcal{L}_D^{0^{-1}}(w)\right) + \varepsilon$$

$$= \varepsilon_{\text{approx}} + \varepsilon_{\text{optim}} + \varepsilon_{\text{estim}}$$

- $\varepsilon_{\text{approx}}$ approximation error due to domain $\Omega$
- $\varepsilon_{\text{optim}}$ optimization error due to surrogate $\ell$
- $\varepsilon_{\text{estim}}$ estimation error due to finite sample $S$
Summary

- Convex optimization
- Linear regression using SGD
- Convex surrogates
- SGD for binary classification with surrogates
- Revisit Perceptron in context of C.O.
- Next: beyond binary classification
Summary

• Convex optimization
• Linear regression using SGD
• Convex surrogates
• SGD for binary classification with surrogates
• Revisit Perceptron in context of C.O.
• Next: beyond binary classification
Summary

- Convex optimization
- Linear regression using SGD
- Convex surrogates
- SGD for binary classification with surrogates
- Revisit Perceptron in context of C.O.
- Next: beyond binary classification