COMPSCI 689
Lecture 8: Support Vector Machines

Brendan O'Connor

College of Information and Computer Sciences
University of Massachusetts Amherst

Slides by Benjamin M. Marlin (marlin@cs.umass.edu).

Overview: Model = Loss + Representation

Losses

Smooth → Non-smooth

Sq. Err

Logistic loss

Hinge loss

LinReg

Linear SVM

(Ker.) SVM

Primal

Logis. Reg.

Kernelized Lin. Reg.

Dual (Kernelization)

$f(x) = \theta^T x$

$\theta \in \mathbb{R}^D$

$f(x) = \sum_i \alpha_i K(x_i, x)$

$\alpha \in \mathbb{R}^N$

Representations

Outline

Support Vector Machines

- Support vector machines are a class of supervised learning approaches based on linear models that use non-differentiable loss functions.

Support Vector Machines

- Support vector machines are a class of supervised learning approaches based on linear models that use non-differentiable loss functions.
- They are learned using regularized risk minimization and can be used with basis expansions.

Support Vector Machines

- Support vector machines are a class of supervised learning approaches based on linear models that use non-differentiable loss functions.
- They are learned using regularized risk minimization and can be used with basis expansions.
- The loss functions used have some interesting properties relative to the squared loss and logistic loss.

Support Vector Machines

- Support vector machines are a class of supervised learning approaches based on linear models that use non-differentiable loss functions.
- They are learned using regularized risk minimization and can be used with basis expansions.
- The loss functions used have some interesting properties relative to the squared loss and logistic loss.
- There are specific approaches for both regression (SVR) and classification (SVC).

Problems with OLS Linear Regression

- One of the drawbacks of squared loss is that it can be very sensitive to the presence of *outliers*.

Problems with OLS Linear Regression

- One of the drawbacks of squared loss is that it can be very sensitive to the presence of *outliers*.
- An outlier is a data point with an output value that is much larger or smaller than the output values of other nearby data points.

Problems with OLS Linear Regression

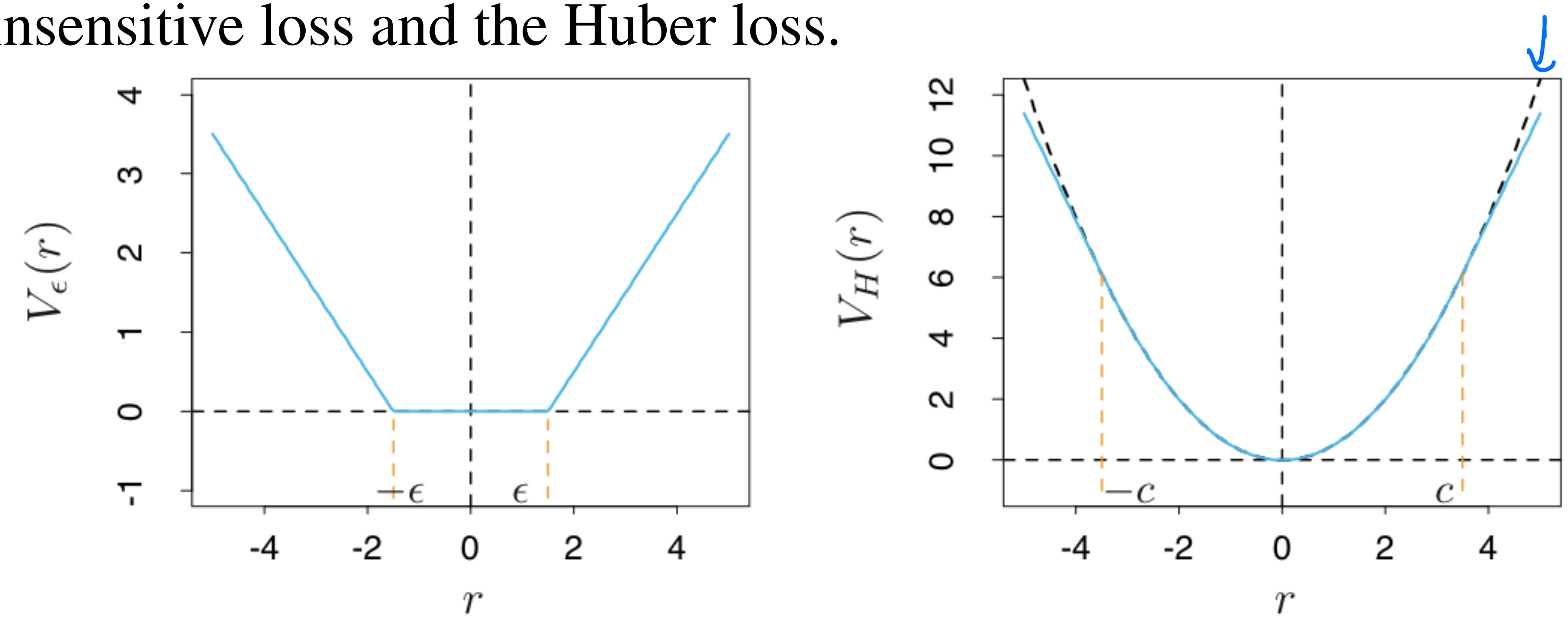
- One of the drawbacks of squared loss is that it can be very sensitive to the presence of *outliers*.
- An outlier is a data point with an output value that is much larger or smaller than the output values of other nearby data points.
- Outliers can be caused by errors in the data collection, or by a data generating process that has heavy tails.

Problems with OLS Linear Regression

- One of the drawbacks of squared loss is that it can be very sensitive to the presence of *outliers*.
- An outlier is a data point with an output value that is much larger or smaller than the output values of other nearby data points.
- Outliers can be caused by errors in the data collection, or by a data generating process that has heavy tails.
- Squared loss prefers to have many small errors instead of a few large errors, and will thus deflect the regression surface toward outliers in order to minimize the MSE.

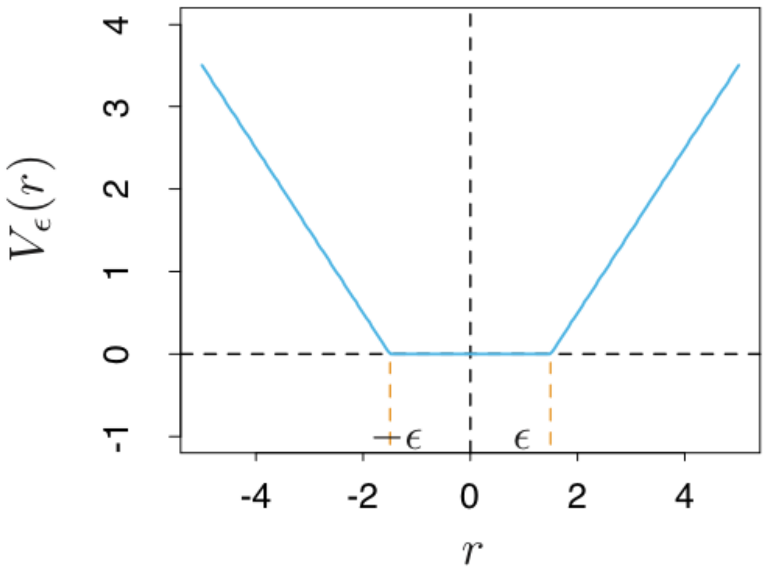
Regression with Other Losses

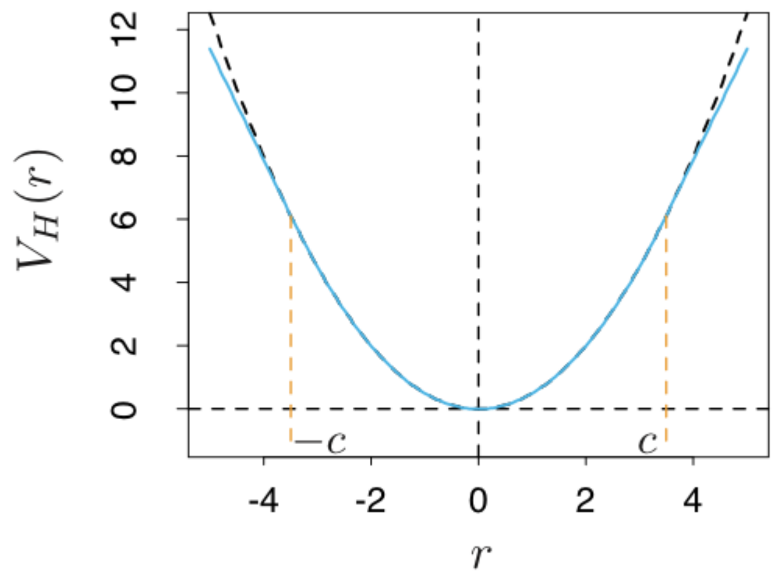
Common alternative losses for regression include the epsilon insensitive loss and the Huber loss.



Regression with Other Losses

Common alternative losses for regression include the epsilon insensitive loss and the Huber loss.

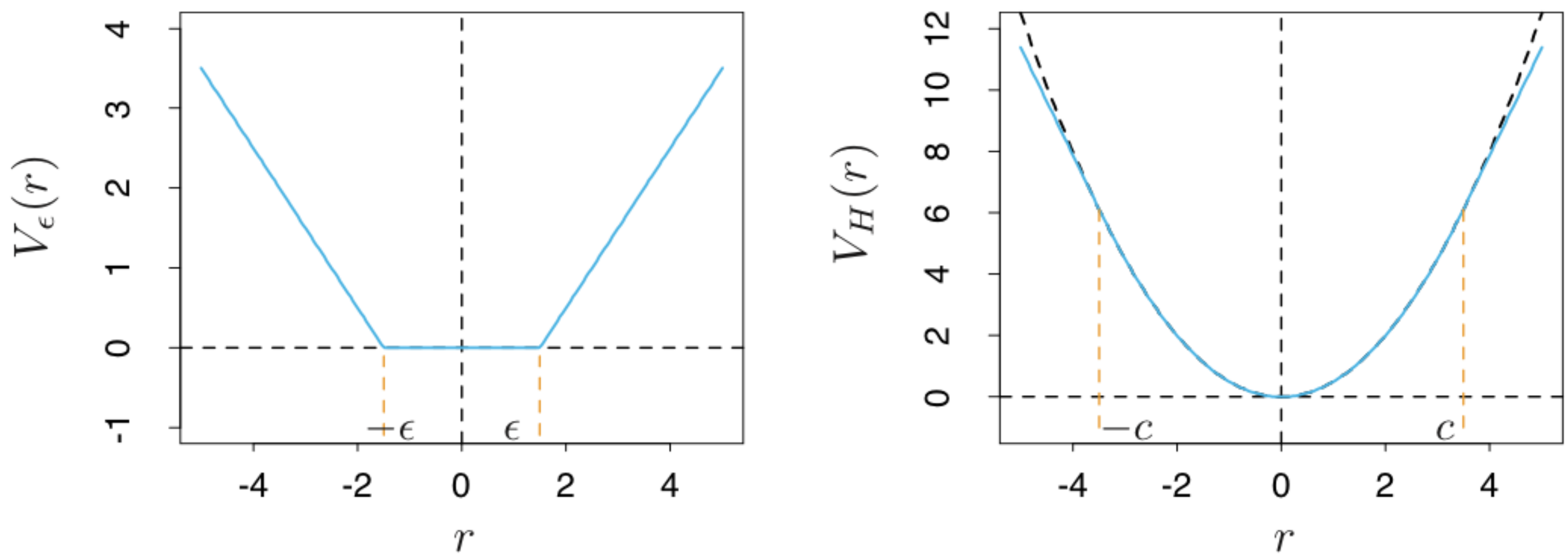




Both of these losses are specifically designed to limit the influence of *outliers* on the model fit.

Regression with Other Losses

Common alternative losses for regression include the epsilon insensitive loss and the Huber loss.



Both of these losses are specifically designed to limit the influence of *outliers* on the model fit.

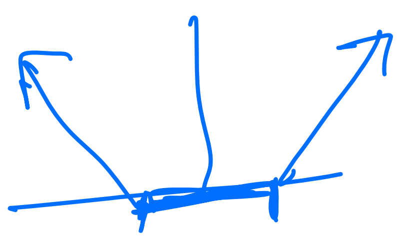
The specific combination of the epsilon insensitive loss with a squared ℓ_2 norm regularizer is referred to as *support vector regression* (SVR).

Support Vector Regression

$$f_\theta(\mathbf{x}) = \mathbf{x}\theta$$

$$\hat{\theta} = \arg\min_{\theta} C \sum_{n=1}^{N} L_\epsilon(y_n, f_\theta(\mathbf{x}_n)) + ||\mathbf{w}||_2^2$$

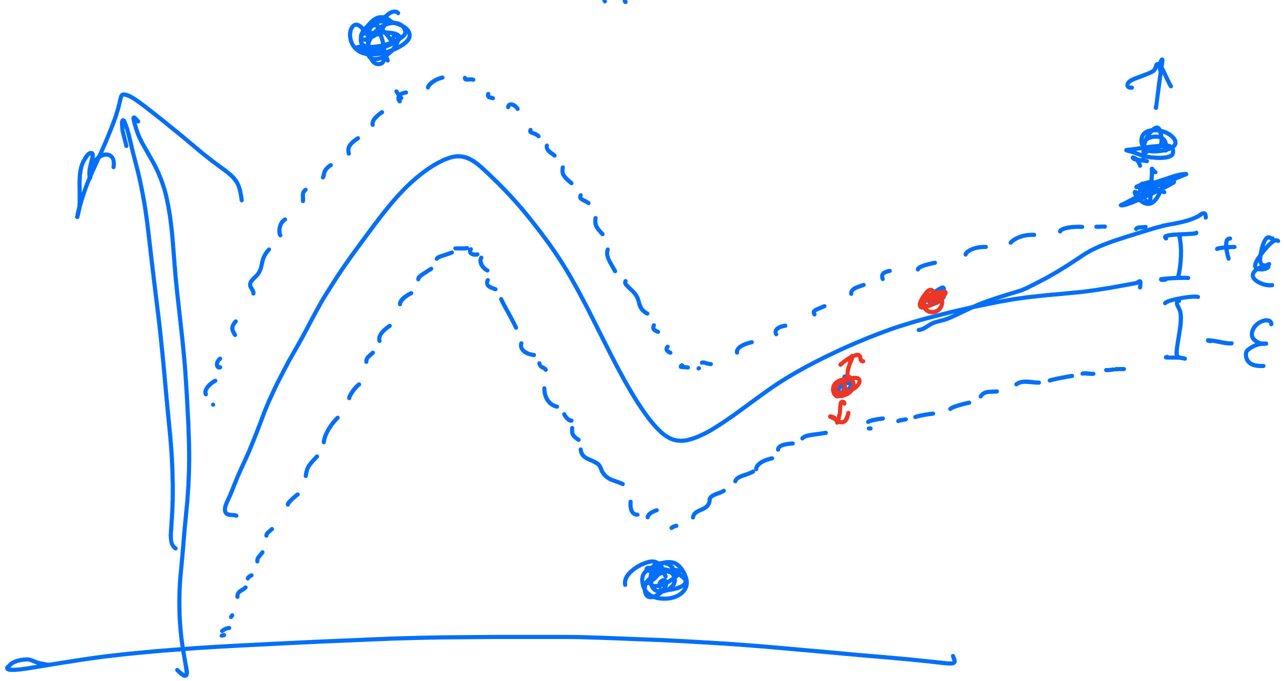
$$L_\epsilon(y, y') = \begin{cases} 0 & \text{... if } |y - y'| < \epsilon \\ |y - y'| - \epsilon & \text{... otherwise} \end{cases}$$



ϵ-insensitive loss for regression

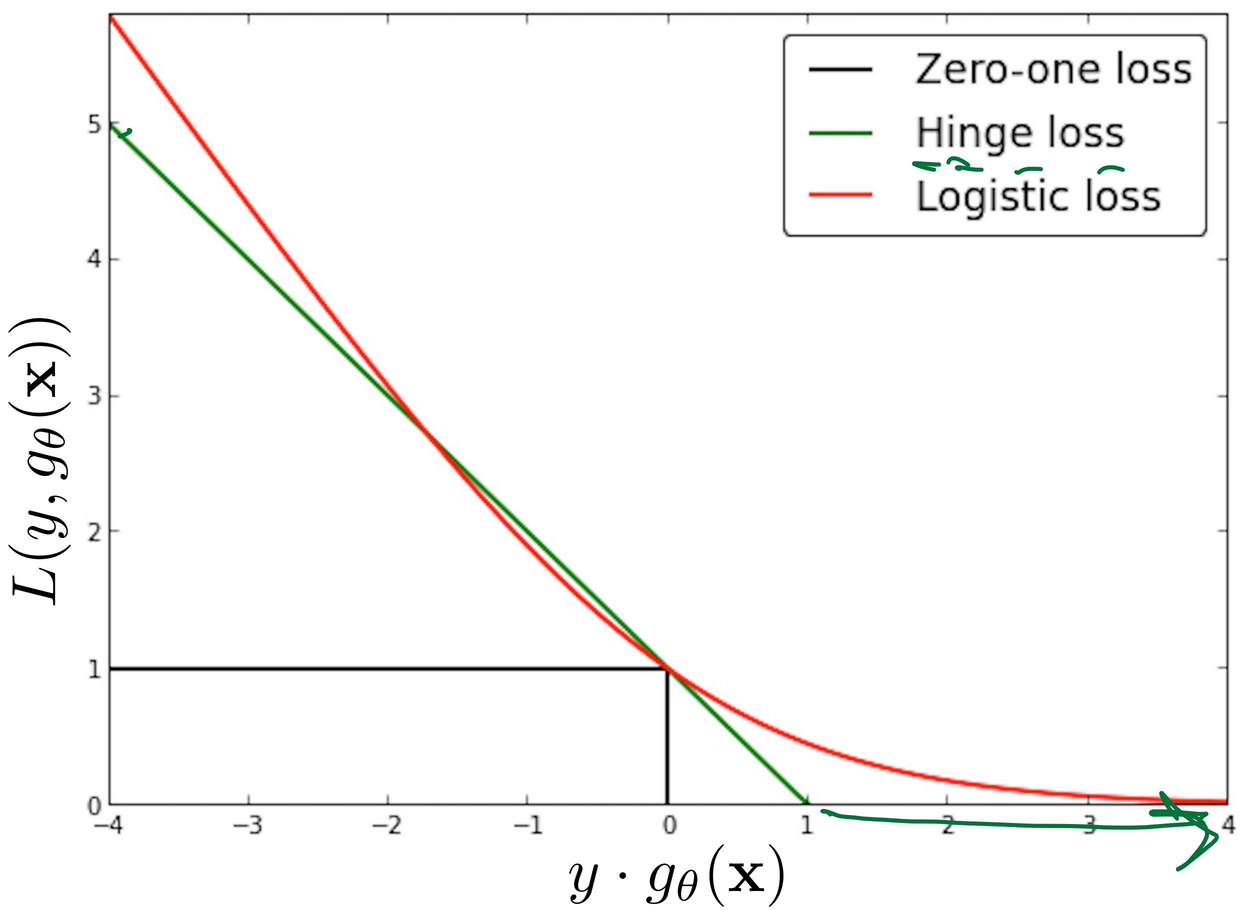
(MLPP 14.5.1)

Support Vectors

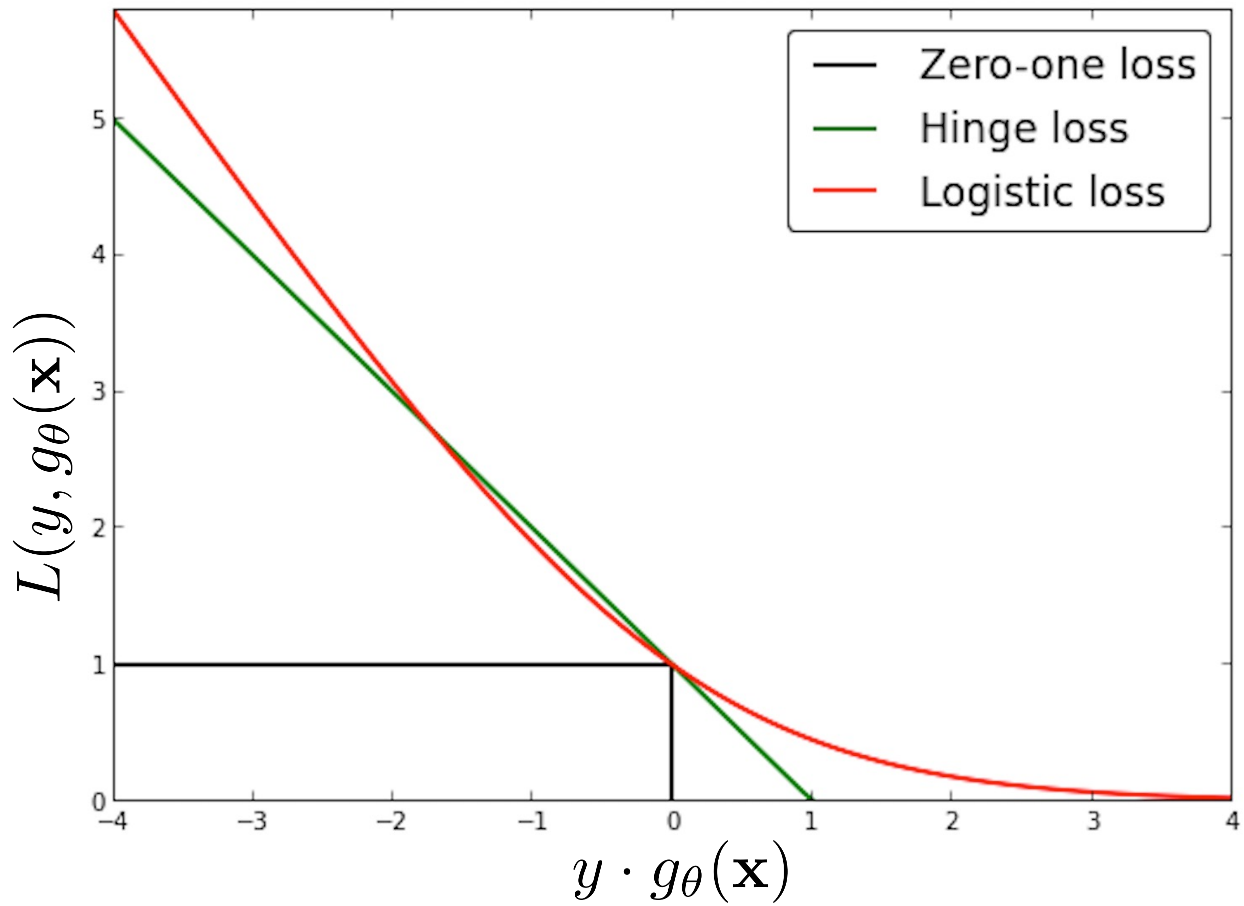


Red: "Non-supports"
irrelevant to f

Classification Losses



Classification Losses

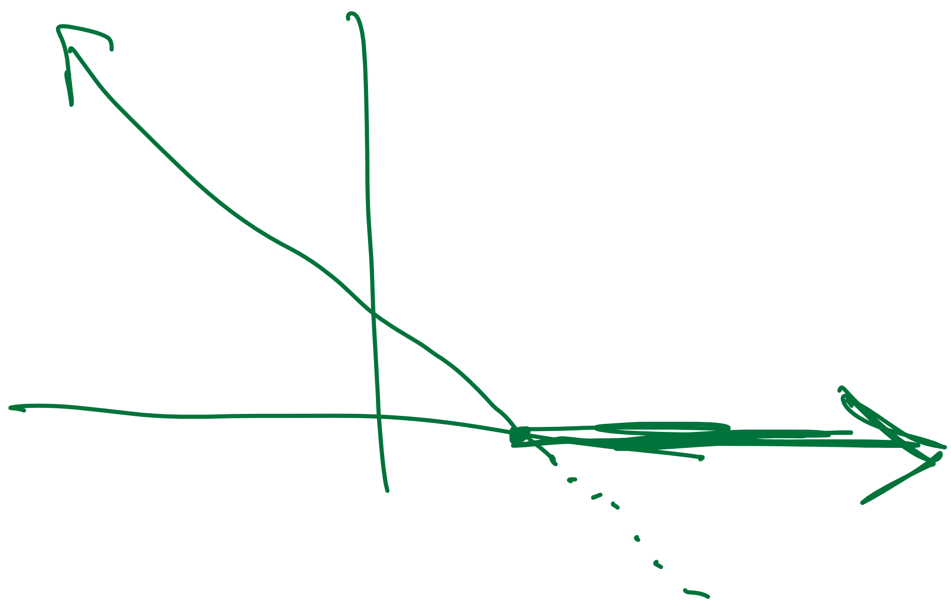


The specific combination of the hinge loss with a squared ℓ_2 norm regularizer is referred to as a *support vector classifier*.

Hinge Loss

- Like the logistic loss, the hinge loss provides an upper bound on the classification error. The basic structure of the function is also similar to that of the logistic loss:

$$L_h(y, g_\theta(\mathbf{x})) = \max(0, 1 - yg_\theta(\mathbf{x}))$$



Hinge Loss

- Like the logistic loss, the hinge loss provides an upper bound on the classification error. The basic structure of the function is also similar to that of the logistic loss:

$$L_h(y, g_\theta(\mathbf{x})) = \max(0, 1 - y g_\theta(\mathbf{x}))$$

- However, while the hinge loss is continuous and convex, it is not differentiable.

Support Vector Classifier

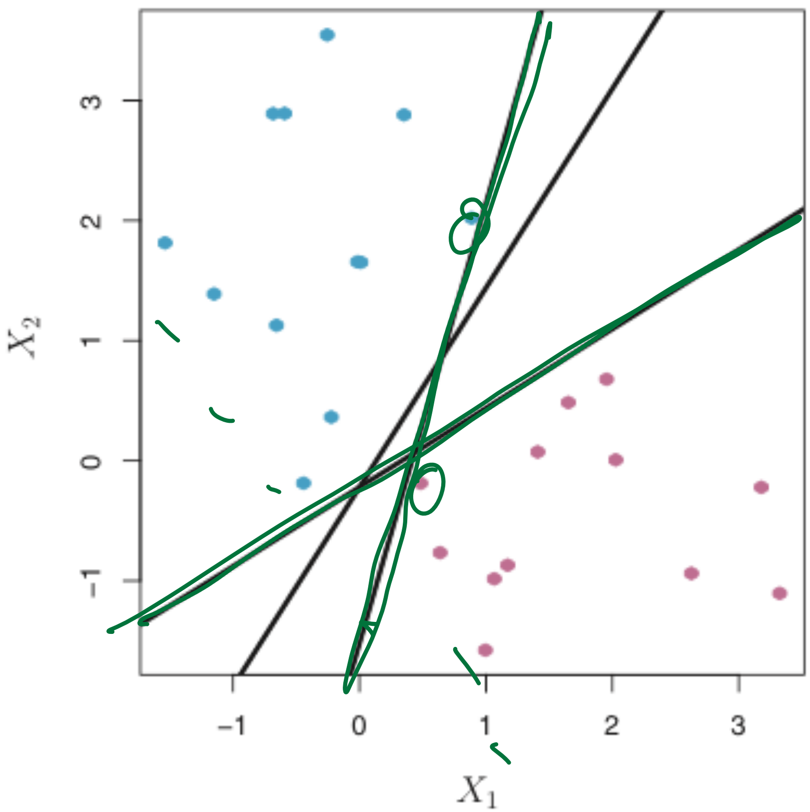
$$f_\theta(\mathbf{x}) = \text{sign}(g_\theta(\mathbf{x}))$$

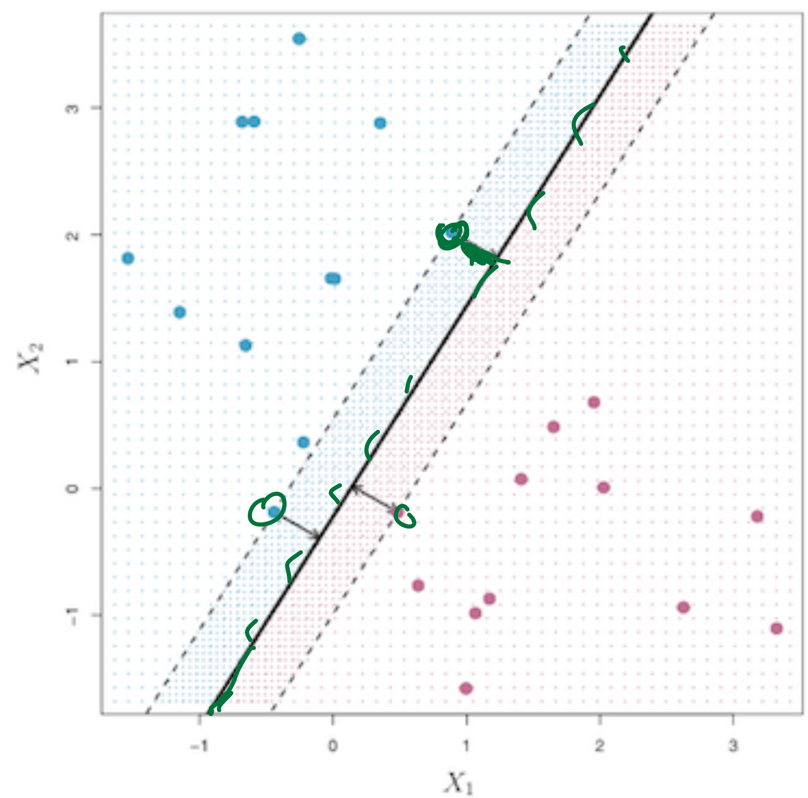
$$g_\theta(\mathbf{x}) = \mathbf{x}\theta = \mathbf{x}\mathbf{w} + b$$

$$\hat{\theta} = \arg\min_\theta C \sum_{n=1}^{N} \max(0, 1 - y_n g_\theta(\mathbf{x}_n)) + ||\mathbf{w}||_2^2$$

Maximum Margin Property

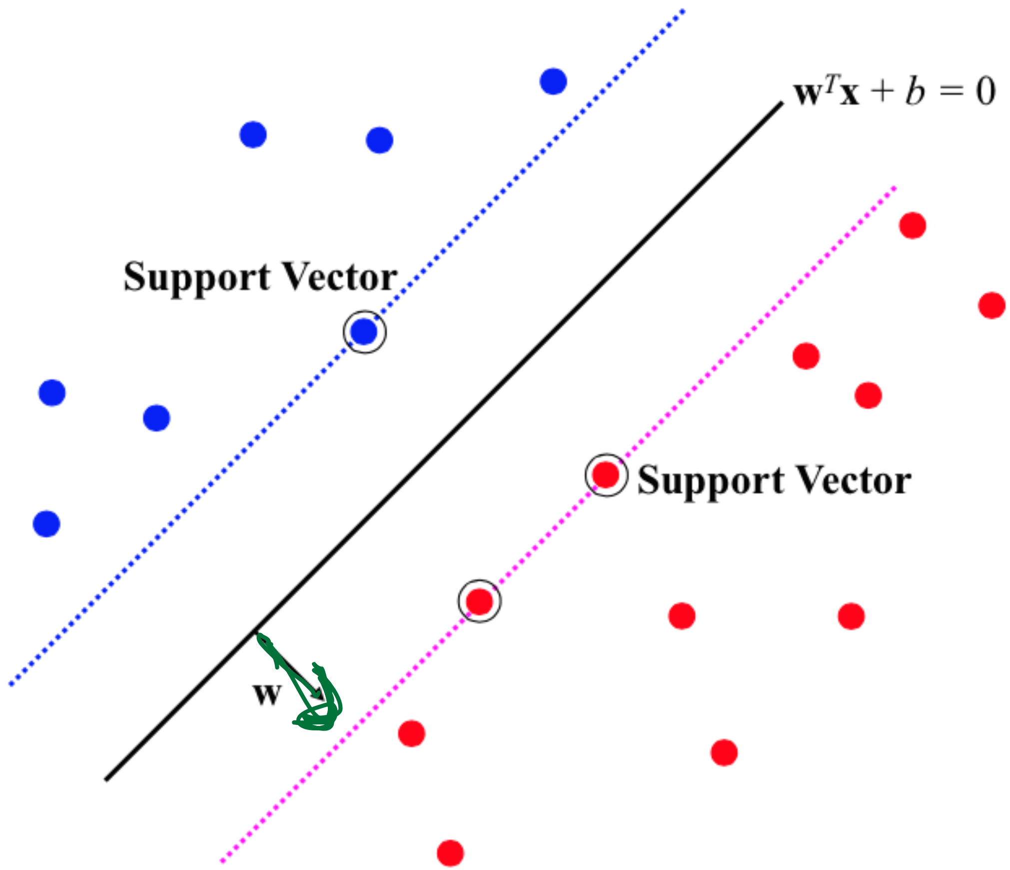
Part of relevance of SVMs stems from the fact that the hinge loss results in the *maximum margin* decision boundary when the training cases are linearly separable.





Support Vector Property

In the linearly separable case, some data points will always fall exactly on the margins. These points are called *support vectors* and they *uniquely determine* the optimal model parameters.



Support Vector Classification

$$f_\theta(\mathbf{x}) = \text{sign}(g_\theta(\mathbf{x}))$$

$$g_\theta(\mathbf{x}) = \mathbf{x}\theta$$

$$\hat{\theta} = \arg\min_\theta C \sum_{n=1}^{N} \max(0, 1 - y_n g_\theta(\mathbf{x}_n)) + ||\mathbf{w}||_2^2$$

Outline

Margin

- Suppose we have a data point $\mathbf{x}_n$. The signed distance from the data point to the separating hyperplane $\mathcal{H} = \{\mathbf{x} | \mathbf{x}\theta = 0\}$ is given by:

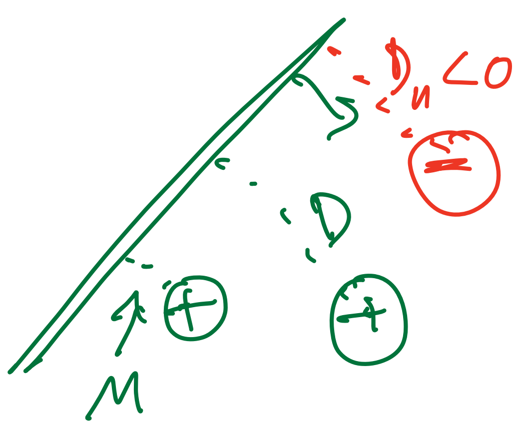
$$D_n(\theta) = \frac{y_n g_\theta(\mathbf{x}_n)}{\|\mathbf{w}\|_2}$$

Margin

- Suppose we have a data point $\mathbf{x}_n$. The signed distance from the data point to the separating hyperplane $\mathcal{H} = \{\mathbf{x}|\mathbf{x}\theta = 0\}$ is given by:

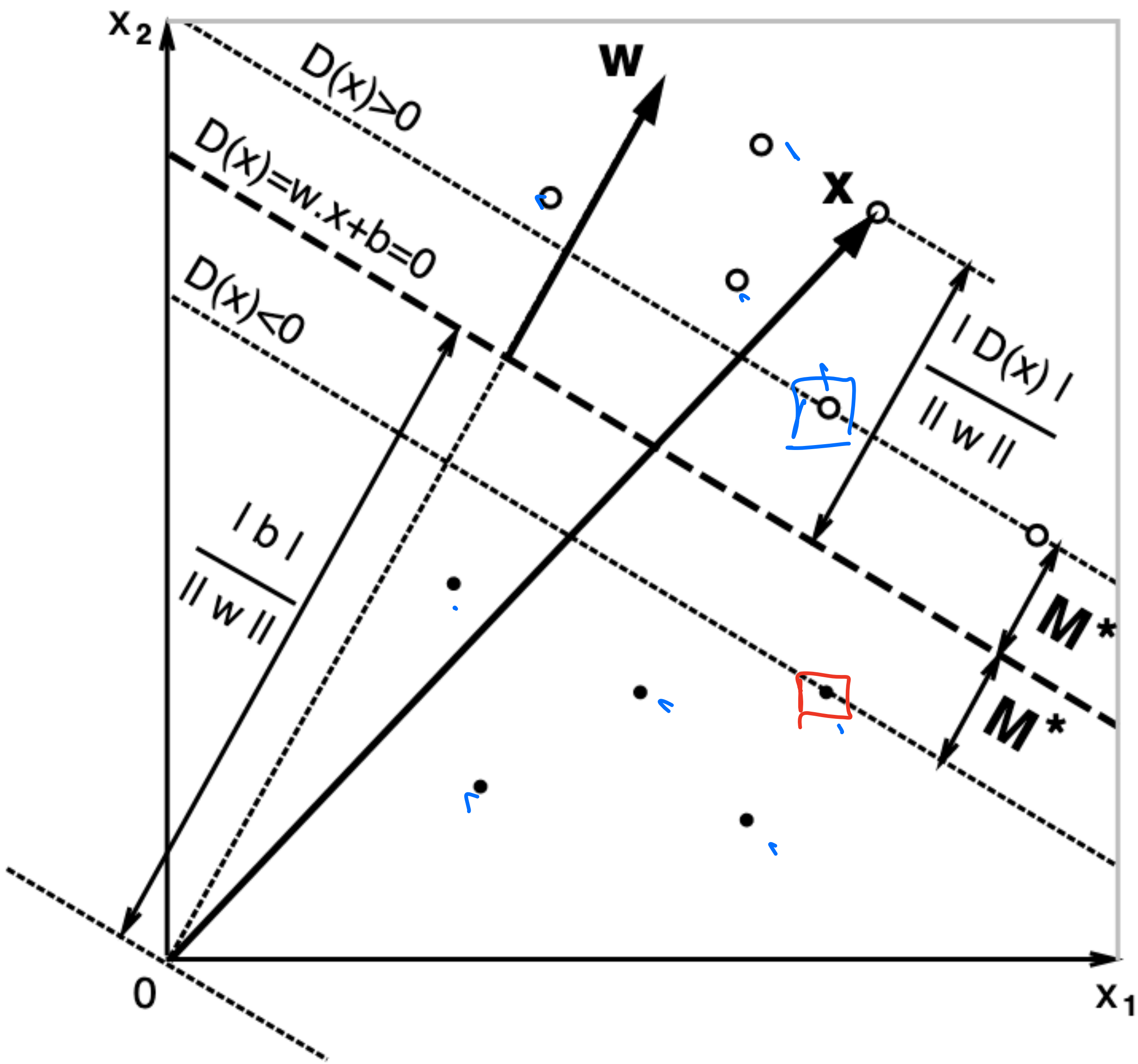
$$D_n(\theta) = \frac{y_n g_\theta(\mathbf{x}_n)}{\|\mathbf{w}\|_2}$$

- The value of the margin is then the minimum over the data set of the signed distances of the data points to the separating hyper-plane:



$$M(\theta) = \min_n D_n(\theta)$$

Margin Example



Margin Maximization

The original SVC optimization problem was to maximize the value of the margin as a function of the model parameters. This gives us:

Margin Maximization

The original SVC optimization problem was to maximize the value of the margin as a function of the model parameters. This gives us:

$$\hat{\theta} = \arg\max_{\theta}\ \min_{n} D_n(\theta)$$

Margin Maximization

The original SVC optimization problem was to maximize the value of the margin as a function of the model parameters. This gives us:

$$\hat{\theta} = \arg\max_{\theta} \min_{n} D_n(\theta)$$

$$\hat{\theta} = \arg\max_{\theta} \underbrace{\min_{n} \frac{y_n g_{\theta}(\mathbf{x}_n)}{\|\mathbf{w}\|_2}}_{\text{Worst-case prediction quality over } n \text{ datapoints}}$$

Quadratic Program Formulation

The direct margin maximization problem is not easy to solve, however, we can re-formulate it into a quadratic program for which there are existing solvers.

Quadratic Program Formulation

The direct margin maximization problem is not easy to solve, however, we can re-formulate it into a quadratic program for which there are existing solvers.

$$\hat{\theta} = \arg\min_{\theta, \epsilon_{1:N}} \|\mathbf{w}\|_2^2 + C \sum_{n=1}^{N} \epsilon_n$$

$$\text{s.t. } \forall n \; y_n g_\theta(\mathbf{x}_n) \geq 1 - \epsilon_n$$

$$\forall n \; \epsilon_n \geq 0$$

Quadratic Program Formulation

The direct margin maximization problem is not easy to solve, however, we can re-formulate it into a quadratic program for which there are existing solvers.

$$\hat{\theta} = \underset{\theta, \epsilon_{1:N}}{\arg\min} \|\mathbf{w}\|_2^2 + C \sum_{n=1}^{N} \epsilon_n$$

$$\text{s.t. } \forall n \; y_n g_\theta(\mathbf{x}_n) \geq 1 - \epsilon_n$$

$$\forall n \; \epsilon_n \geq 0$$

This version of the problem can be further manipulated into the hinge loss formulation.

Dual Quadratic Program Formulation

The constrained quadratic program can also be re-written in a form where the parameters are weights applied to the data cases. This is called the SVM dual formulation.

Dual Quadratic Program Formulation

The constrained quadratic program can also be re-written in a form where the parameters are weights applied to the data cases. This is called the SVM dual formulation.

$$\arg\max_{\boldsymbol{\alpha}} \sum_{n=1}^{N} \alpha_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} y_n y_m \alpha_n \alpha_m \mathbf{K}_{nm}$$

$$\text{s.t. } \forall n \;\; 0 \leq \alpha_n \leq C$$

$$\sum_{n=1}^{N} \alpha_n y_n = 0$$

Dual Quadratic Program Formulation

The constrained quadratic program can also be re-written in a form where the parameters are weights applied to the data cases. This is called the SVM dual formulation.

$$\arg\max_{\boldsymbol{\alpha}} \sum_{n=1}^{N} \alpha_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} y_n y_m \alpha_n \alpha_m \mathbf{K}_{nm}$$

$$\text{s.t. } \forall n \;\; 0 \leq \alpha_n \leq C$$

$$\sum_{n=1}^{N} \alpha_n y_n = 0$$

The matrix $\mathbf{K}$ that appears in the objective contains the inner products between all pairs of training data vectors: $\mathbf{K}_{nm} = \mathbf{x}_n^T \mathbf{x}_m$. The matrix $\mathbf{K}$ is thus positive semi-definite and the optimization problem is maximizing a concave function.

Dual Prediction

Given the value of α, we make predictions using the dual formulation as follows:

$$f_{svm}(\mathbf{x}) = \text{sign}(\sum_{n=1}^{N} y_n \alpha_n \mathbf{x}_n^T \mathbf{x} + b)$$

Dual Prediction

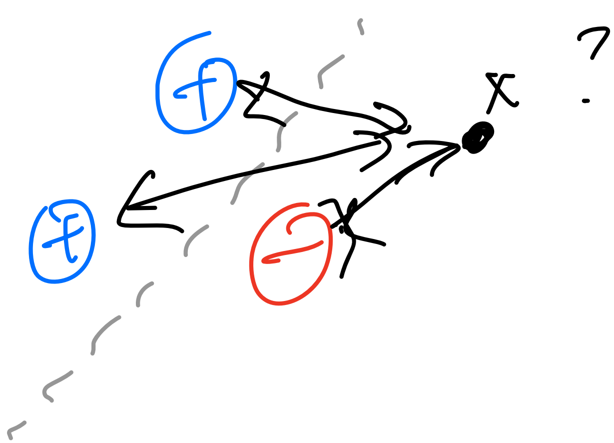
Given the value of α, we make predictions using the dual formulation as follows:

+1 or −1 vote

$K(x_n, x)$

$$f_{svm}(\mathbf{x}) = \text{sign}\left(\sum_{n=1}^{N} y_n \alpha_n \mathbf{x}_n^T \mathbf{x} + b\right)$$

Note that this requires having access to the training data at prediction time. Also note that both the learning problem and the prediction problem are both based on computing inner products between data cases.



Stopped here

3/7/23

Basis Expansion and Kernels

We can apply basis expansion to any formulation of the learning problem, but it's particularly interesting in the dual formulation:

$$
\begin{aligned}
f_{svm}(\mathbf{x}) &= \text{sign}\left(\sum_{n=1}^{N} y_n \alpha_n \phi(\mathbf{x}_n)^T \phi(\mathbf{x}) + b\right) \\
&= \text{sign}\left(\sum_{n=1}^{N} y_n \alpha_n \mathcal{K}(\mathbf{x}_n, \mathbf{x}) + b\right)
\end{aligned}
$$

Basis Expansion and Kernels

We can apply basis expansion to any formulation of the learning problem, but it's particularly interesting in the dual formulation:

$$
\begin{aligned}
f_{svm}(\mathbf{x}) &= \mathrm{sign}\Big(\sum_{n=1}^{N} y_n \alpha_n \phi(\mathbf{x}_n)^T \phi(\mathbf{x}) + b\Big) \\
&= \mathrm{sign}\Big(\sum_{n=1}^{N} y_n \alpha_n \mathcal{K}(\mathbf{x}_n, \mathbf{x}) + b\Big)
\end{aligned}
$$

The function $\mathcal{K}$ is referred to as a *kernel function* and must satisfy a property known as *Mercer*'s condition.

Example: Kernels

- Linear Kernel: $\mathcal{K}(\mathbf{x}', \mathbf{x}) = \mathbf{x}^T\mathbf{x}'$
- Polynomial Kernel: $\mathcal{K}(\mathbf{x}', \mathbf{x}) = (1 + \mathbf{x}^T\mathbf{x}')^B$
- Exponential (RBF) Kernel: $\mathcal{K}(\mathbf{x}', \mathbf{x}) = \exp\Big(-\gamma\|\mathbf{x}' - \mathbf{x}\|_2^2\Big)$
- ... and many more kernel functions, including for inputs that are not real-valued.

Outline

Back to the Hinge Loss Formulation

$$f_\theta(\mathbf{x}) = \text{sign}(g_\theta(\mathbf{x}))$$

$$g_\theta(\mathbf{x}) = \mathbf{x}\theta = \mathbf{x}\mathbf{w} + b$$

$$\hat{\theta} = \arg\min_\theta C \sum_{n=1}^{N} \max(0, 1 - y_n g_\theta(\mathbf{x}_n)) + ||\mathbf{w}||_2^2$$

Global Optimality of Convex Non-Differentiable Functions

- Our existing optimization theory only holds for the case of differentiable functions.

[1] This section mostly follows the presentation in Boyd's EE364b - Convex Optimization II course.

Global Optimality of Convex Non-Differentiable Functions

- Our existing optimization theory only holds for the case of differentiable functions.
- However, it turns out that many results generalize to the case of non-differentiable functions that are convex and we can use them to directly minimize the hinge loss.

[1] This section mostly follows the presentation in Boyd's EE364b - Convex Optimization II course.

Global Optimality of Convex Non-Differentiable Functions

- Our existing optimization theory only holds for the case of differentiable functions.
- However, it turns out that many results generalize to the case of non-differentiable functions that are convex and we can use them to directly minimize the hinge loss.
- To begin, strongly convex non-differentiable functions have a unique global minima, exactly as with convex differentiable functions.

[1] This section mostly follows the presentation in Boyd's EE364b - Convex Optimization II course.

Global Optimality of Convex Non-Differentiable Functions

- Our existing optimization theory only holds for the case of differentiable functions.
- However, it turns out that many results generalize to the case of non-differentiable functions that are convex and we can use them to directly minimize the hinge loss.
- To begin, strongly convex non-differentiable functions have a unique global minima, exactly as with convex differentiable functions.
- We will begin with the characterization of the minimizer of a non-differentiable convex function. [1]

[1]This section mostly follows the presentation in Boyd's EE364b - Convex Optimization II course.

Subgradient

- Let $f : \mathbb{R}^D \to \mathbb{R}$. A vector $\mathbf{g} \in \mathbb{R}^D$ is said to be a sub-gradient of f at a point $\mathbf{x}_o \in \mathbb{R}^D$ if for all $\mathbf{x} \in \mathbb{R}^D$:

$$f(\mathbf{x}) \geq f(\mathbf{x}_o) + \mathbf{g}^T \cdot (\mathbf{x} - \mathbf{x}_o)$$

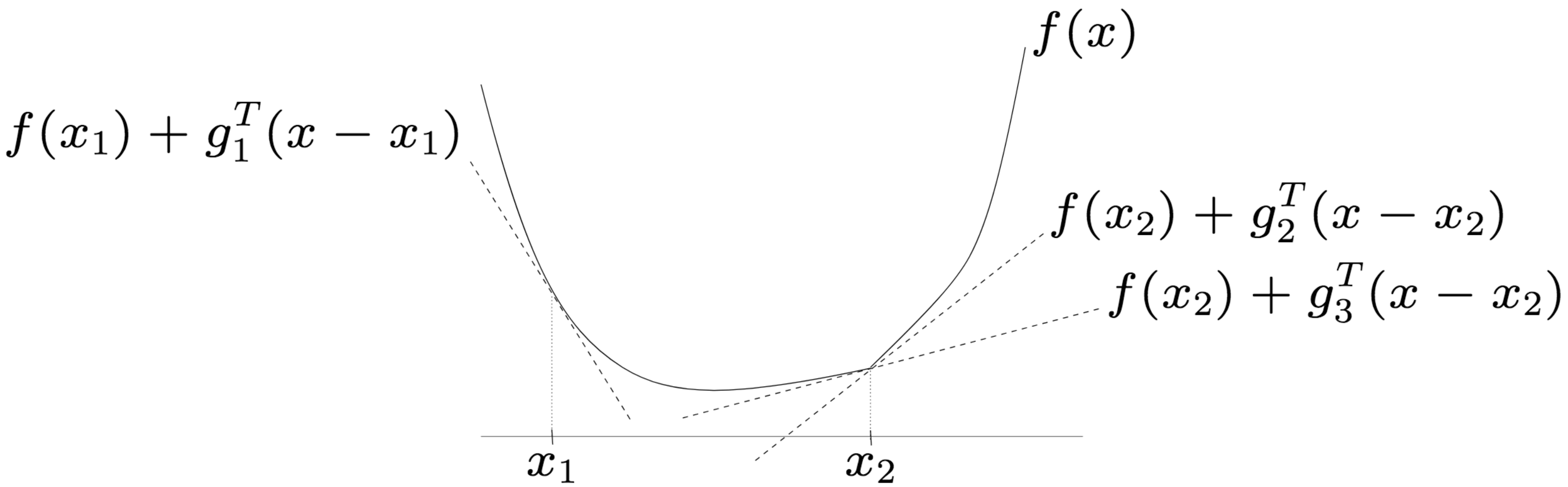
Subgradient

- Let $f : \mathbb{R}^D \rightarrow \mathbb{R}$. A vector $\mathbf{g} \in \mathbb{R}^D$ is said to be a sub-gradient of f at a point $\mathbf{x}_o \in \mathbb{R}^D$ if for all $\mathbf{x} \in \mathbb{R}^D$:

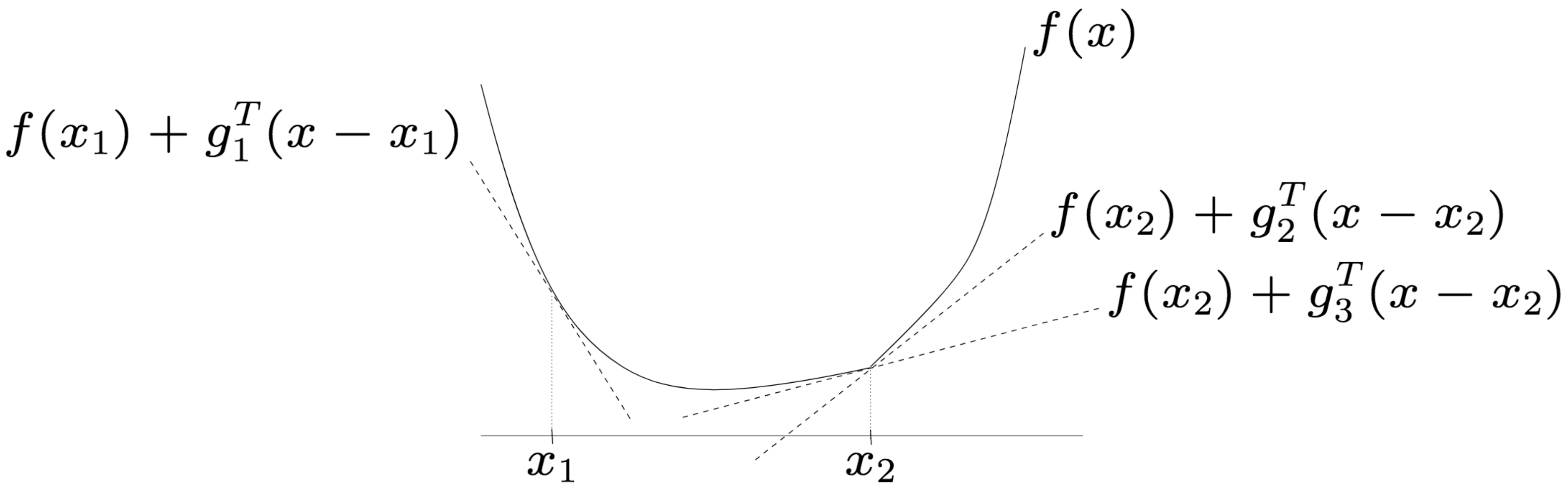
$$f(\mathbf{x}) \geq f(\mathbf{x}_o) + \mathbf{g}^T \cdot (\mathbf{x} - \mathbf{x}_o)$$

- That is to say, the hyperplane defined by $h(\mathbf{x}) = f(\mathbf{x}_o) + \mathbf{g}^T \cdot (\mathbf{x} - \mathbf{x}_o)$ lies at or below below f everywhere and touches f at $\mathbf{x}_o$.

Example: Subgradients



Example: Subgradients



In this example, $\mathbf{g}_1$ is the unique subgradient of f at $\mathbf{x}_1$. Due to f being non-differentiable at $\mathbf{x}_2$, both $\mathbf{g}_2$ and $\mathbf{g}_3$ are subgradients of f at $\mathbf{x}_2$.

Subdifferentials

- If f is convex and is differentiable at $\mathbf{x}_o$, then $\nabla f(\mathbf{x}_o)$ is its unique subgradient at $\mathbf{x}_o$.

Subdifferentials

- If f is convex and is differentiable at $\mathbf{x}_o$, then $\nabla f(\mathbf{x}_o)$ is its unique subgradient at $\mathbf{x}_o$.
- If f is convex and non-differentiable at $\mathbf{x}_o$, it will generally have more that one vector $\mathbf{g}$ satisfying the subgradient property.

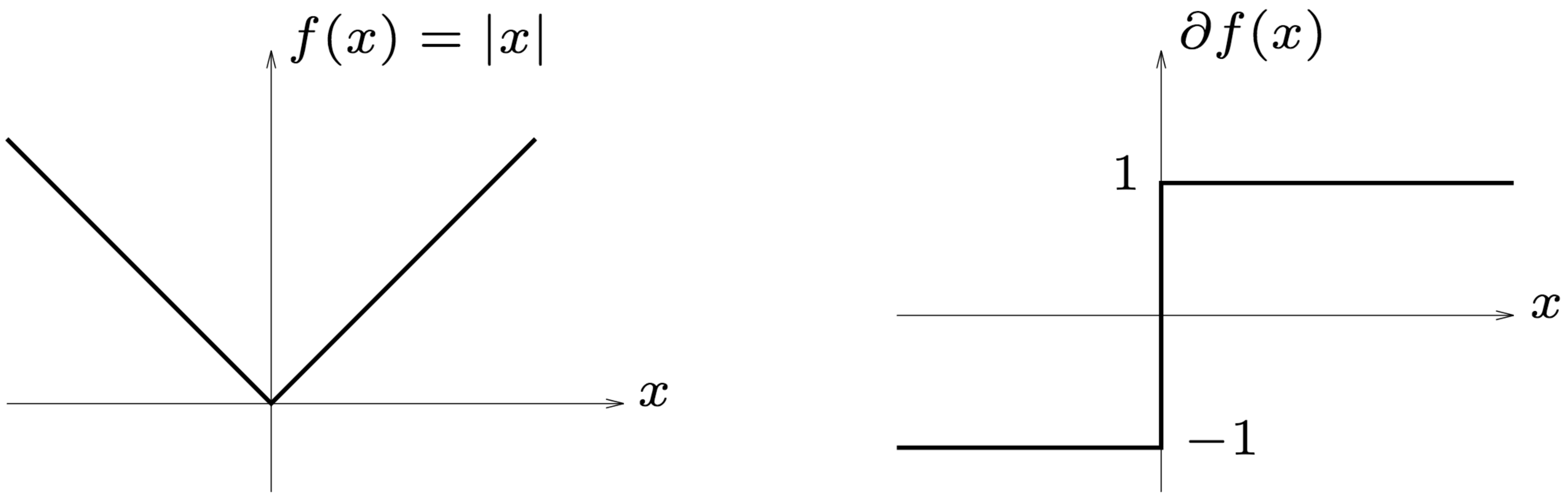
Subdifferentials

- If f is convex and is differentiable at $\mathbf{x}_o$, then $\nabla f(\mathbf{x}_o)$ is its unique subgradient at $\mathbf{x}_o$.
- If f is convex and non-differentiable at $\mathbf{x}_o$, it will generally have more that one vector $\mathbf{g}$ satisfying the subgradient property.
- The set of all subgradients of f at $\mathbf{x}_o$ is called the subdifferential of f at $\mathbf{x}_o$ denoted by $\partial f(\mathbf{x}_o)$.

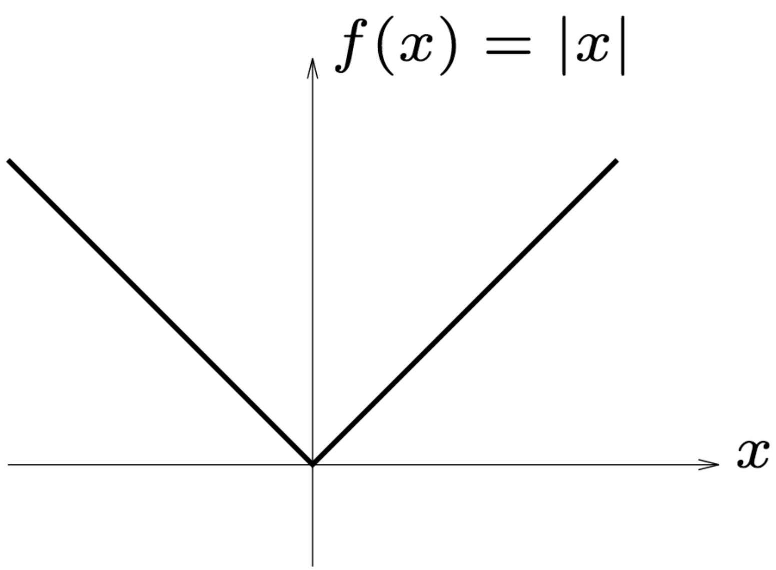
Subdifferentials

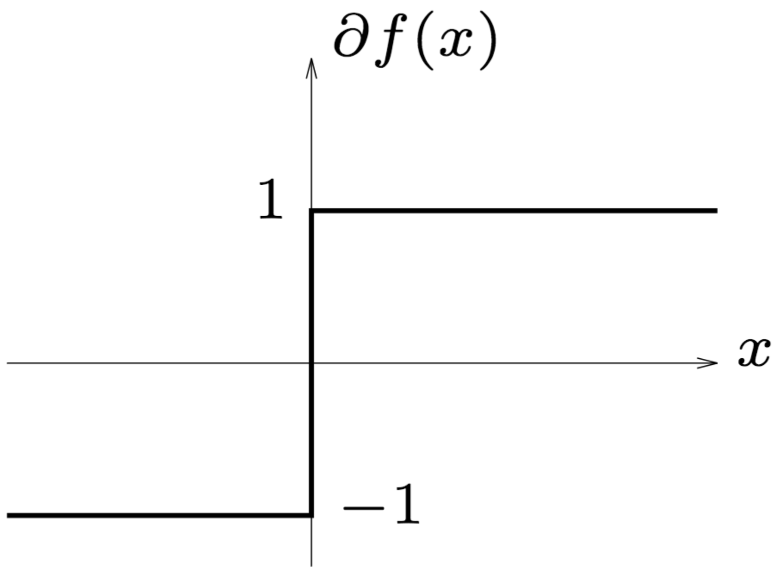
- If f is convex and is differentiable at $\mathbf{x}_o$, then $\nabla f(\mathbf{x}_o)$ is its unique subgradient at $\mathbf{x}_o$.
- If f is convex and non-differentiable at $\mathbf{x}_o$, it will generally have more that one vector $\mathbf{g}$ satisfying the subgradient property.
- The set of all subgradients of f at $\mathbf{x}_o$ is called the subdifferential of f at $\mathbf{x}_o$ denoted by $\partial f(\mathbf{x}_o)$.
- $\partial f(\mathbf{x}_o)$ is a closed, convex set in $\mathbb{R}^D$. If f is convex, $\partial f(\mathbf{x}_o)$ is always non-empty.

Example: Subdifferentials



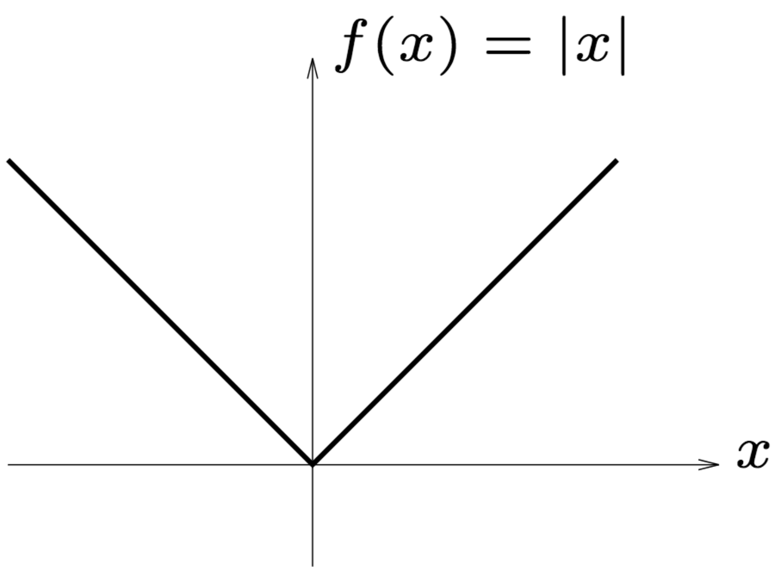
Example: Subdifferentials

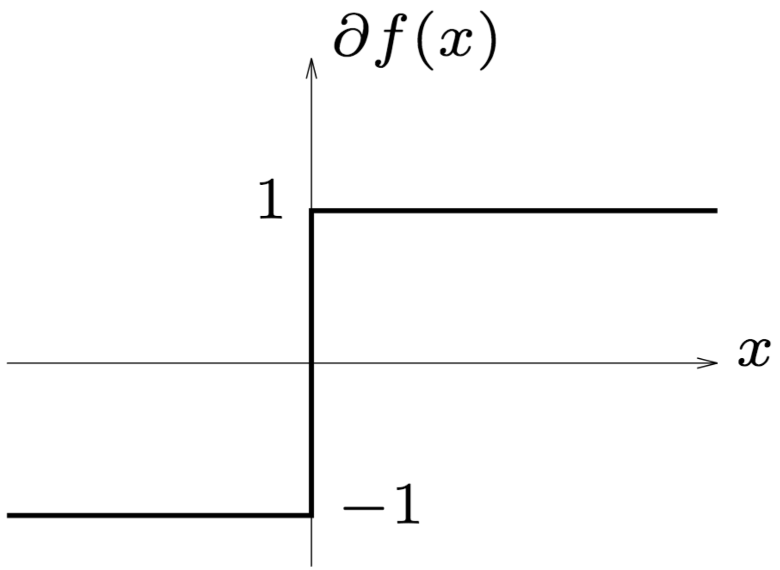




The righthand plot shows $\partial f(x)$ for $f(x) = |x|$.

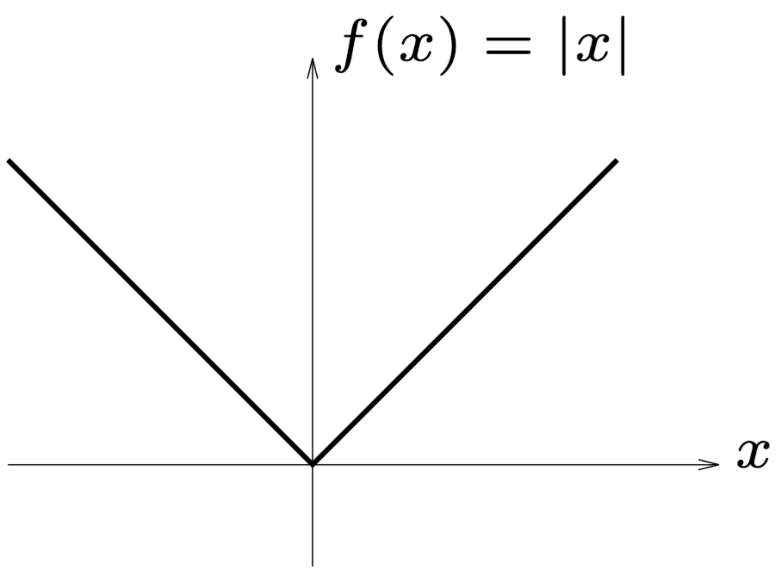
Example: Subdifferentials

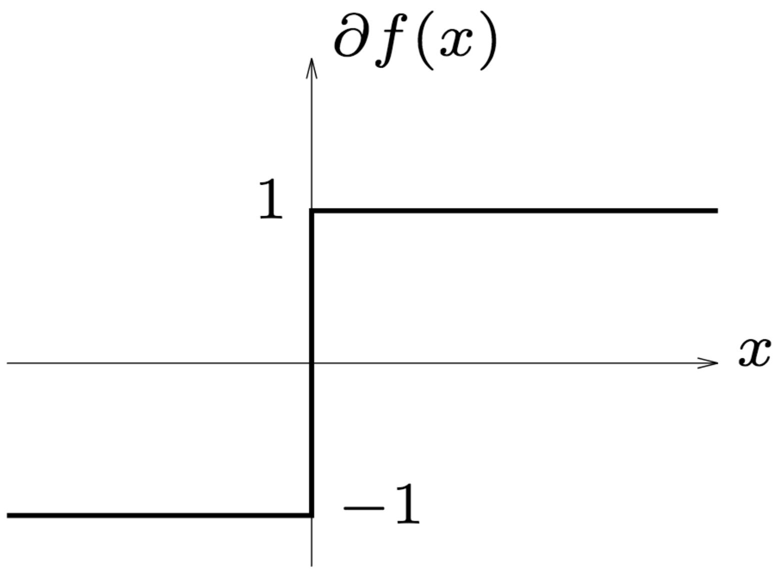




The righthand plot shows $\partial f(x)$ for $f(x) = |x|$.
We have $\partial f(\mathbf{x}) = \{\text{sign}(x)\}$ for $x \neq 0$.

Example: Subdifferentials





The righthand plot shows $\partial f(x)$ for $f(x) = |x|$.
We have $\partial f(\mathbf{x}) = \{\text{sign}(x)\}$ for $x \neq 0$.
When $x = 0$, the line $|0| + g \cdot (x - 0) = g \cdot x$ will lie below f everywhere only if $g \in [-1, 1]$.

Characterizing the Global Minimum

- Let $f : \mathbb{R}^D \to \mathbb{R}$ be a convex function.

Characterizing the Global Minimum

- Let $f : \mathbb{R}^D \to \mathbb{R}$ be a convex function.
- $\mathbf{x}_*$ is the global minimizer of f if and only if $\mathbf{0} \in \partial f(\mathbf{x}_*)$.

Characterizing the Global Minimum

- Let $f : \mathbb{R}^D \to \mathbb{R}$ be a convex function.
- $\mathbf{x}_*$ is the global minimizer of f if and only if $\mathbf{0} \in \partial f(\mathbf{x}_*)$.
- This is a generalization of the idea of a stationary point to include the case of non-differentiable functions.

Finding Subdifferentials

- Suppose that x_0 is a point of non-differentiability for a 1-dimensional convex function $f(x)$.

Finding Subdifferentials

- Suppose that x_0 is a point of non-differentiability for a 1-dimensional convex function $f(x)$.
- Suppose that in the neighborhood $[a, x_0]$, for some $a < x_0$, the value of $f(x)$ is given by a differentiable function $g(x)$ and in the neighborhood $[x_0, b]$ for some $b > x_0$, the value of $f(x)$ is given by a differentiable function $h(x)$.

Finding Subdifferentials

- Suppose that x_0 is a point of non-differentiability for a 1-dimensional convex function $f(x)$.
- Suppose that in the neighborhood $[a, x_0]$, for some $a < x_0$, the value of $f(x)$ is given by a differentiable function $g(x)$ and in the neighborhood $[x_0, b]$ for some $b > x_0$, the value of $f(x)$ is given by a differentiable function $h(x)$.
- Then, the subdifferential of $f(x)$ at x_0 is:

$$\partial f(x_0) = \left[\frac{dg(x)}{dx}|_{x_0}, \frac{dh(x)}{dx}|_{x_0}\right]$$

Partial Linearity and Chain Rule

The subdifferential operator satisfies partial linearity and chain rule properties:

- **Scaling:** If $f(\mathbf{x})$ is a convex function and $\alpha > 0$, then if $\mathbf{g} \in \partial f(\mathbf{x})$, $\alpha \mathbf{g} \in \partial(\alpha f(\mathbf{x}))$

Partial Linearity and Chain Rule

The subdifferential operator satisfies partial linearity and chain rule properties:

- **Scaling:** If $f(\mathbf{x})$ is a convex function and $\alpha > 0$, then if $\mathbf{g} \in \partial f(\mathbf{x})$, $\alpha\mathbf{g} \in \partial(\alpha f(\mathbf{x}))$
- **Addition:** If $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$ are convex functions and $\mathbf{g}_1 \in \partial f_1(\mathbf{x})$ and $\mathbf{g}_2 \in \partial f_2(\mathbf{x})$, then $\mathbf{g}_1 + \mathbf{g}_2 \in \partial(f_1(\mathbf{x}) + f_2(\mathbf{x}))$

Partial Linearity and Chain Rule

The subdifferential operator satisfies partial linearity and chain rule properties:

- **Scaling:** If $f(\mathbf{x})$ is a convex function and $\alpha > 0$, then if $\mathbf{g} \in \partial f(\mathbf{x})$, $\alpha\mathbf{g} \in \partial(\alpha f(\mathbf{x}))$
- **Addition:** If $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$ are convex functions and $\mathbf{g}_1 \in \partial f_1(\mathbf{x})$ and $\mathbf{g}_2 \in \partial f_2(\mathbf{x})$, then $\mathbf{g}_1 + \mathbf{g}_2 \in \partial(f_1(\mathbf{x}) + f_2(\mathbf{x}))$
- **Chain Rule**: If $f : \mathbb{R} \to \mathbb{R}$ is a convex function and $h : \mathbb{R}^D \to \mathbb{R}$ is a linear function $h(\mathbf{x}) = \mathbf{x}\mathbf{a} + b$, then the sub-differential of $f(h(\mathbf{x}))$ is $\{g\mathbf{a}^T | g \in \partial f(y), y = h(\mathbf{x})\}$.

Partial Linearity and Chain Rule

The subdifferential operator satisfies partial linearity and chain rule properties:

- **Scaling:** If $f(\mathbf{x})$ is a convex function and $\alpha > 0$, then if $\mathbf{g} \in \partial f(\mathbf{x})$, $\alpha\mathbf{g} \in \partial(\alpha f(\mathbf{x}))$
- **Addition:** If $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$ are convex functions and $\mathbf{g}_1 \in \partial f_1(\mathbf{x})$ and $\mathbf{g}_2 \in \partial f_2(\mathbf{x})$, then $\mathbf{g}_1 + \mathbf{g}_2 \in \partial(f_1(\mathbf{x}) + f_2(\mathbf{x}))$
- **Chain Rule**: If $f : \mathbb{R} \to \mathbb{R}$ is a convex function and $h : \mathbb{R}^D \to \mathbb{R}$ is a linear function $h(\mathbf{x}) = \mathbf{x}\mathbf{a} + b$, then the sub-differential of $f(h(\mathbf{x}))$ is $\{g\mathbf{a}^T | g \in \partial f(y), y = h(\mathbf{x})\}$.

Partial Linearity and Chain Rule

The subdifferential operator satisfies partial linearity and chain rule properties:

- **Scaling:** If $f(\mathbf{x})$ is a convex function and $\alpha > 0$, then if $\mathbf{g} \in \partial f(\mathbf{x})$, $\alpha\mathbf{g} \in \partial(\alpha f(\mathbf{x}))$
- **Addition:** If $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$ are convex functions and $\mathbf{g}_1 \in \partial f_1(\mathbf{x})$ and $\mathbf{g}_2 \in \partial f_2(\mathbf{x})$, then $\mathbf{g}_1 + \mathbf{g}_2 \in \partial(f_1(\mathbf{x}) + f_2(\mathbf{x}))$
- **Chain Rule**: If $f : \mathbb{R} \to \mathbb{R}$ is a convex function and $h : \mathbb{R}^D \to \mathbb{R}$ is a linear function $h(\mathbf{x}) = \mathbf{x}\mathbf{a} + b$, then the sub-differential of $f(h(\mathbf{x}))$ is $\{g\mathbf{a}^T | g \in \partial f(y), y = h(\mathbf{x})\}$.

These properties can be used to determine the subdifferentials of more some complex functions if they reduce to certain combinations or compositions of simpler functions.

Outline

Sub-gradient Descent

- If f is differentiable at $\mathbf{x}$ then $\partial f(\mathbf{x}) = \{\nabla f(\mathbf{x})\}$ and $-\mathbf{g} = -\nabla f(\mathbf{x})$ is a descent direction unless $\nabla f(\mathbf{x}) = 0$.

Sub-gradient Descent

- If f is differentiable at $\mathbf{x}$ then $\partial f(\mathbf{x}) = \{\nabla f(\mathbf{x})\}$ and $-\mathbf{g} = -\nabla f(\mathbf{x})$ is a descent direction unless $\nabla f(\mathbf{x}) = 0$.
- If f is not differentiable at $\mathbf{x}$, and $\mathbf{0} \notin \partial f(\mathbf{x})$, then there exists at least one $\mathbf{g} \in \partial f(\mathbf{x})$ where $-\mathbf{g}$ is a descent direction.

Sub-gradient Descent

- If f is differentiable at $\mathbf{x}$ then $\partial f(\mathbf{x}) = \{\nabla f(\mathbf{x})\}$ and $-\mathbf{g} = -\nabla f(\mathbf{x})$ is a descent direction unless $\nabla f(\mathbf{x}) = 0$.
- If f is not differentiable at $\mathbf{x}$, and $\mathbf{0} \notin \partial f(\mathbf{x})$, then there exists at least one $\mathbf{g} \in \partial f(\mathbf{x})$ where $-\mathbf{g}$ is a descent direction.
- If f is not differentiable at $\mathbf{x}$, then there may exist some $\mathbf{g} \in \partial f(\mathbf{x})$ where $-\mathbf{g}$ is a **not** descent direction.

Sub-gradient Descent

Despite the fact that not all sub-gradients yield descent directions, it is still possible to minimize a convex non-differentiable function using a fairly basic sub-gradient descent procedure:

Sub-gradient Descent

Despite the fact that not all sub-gradients yield descent directions, it is still possible to minimize a convex non-differentiable function using a fairly basic sub-gradient descent procedure:

- Initialize $\mathbf{x}_0 \in \mathbb{R}^D, f_{min} = \infty, \mathbf{x}_* = 0$

Sub-gradient Descent

Despite the fact that not all sub-gradients yield descent directions, it is still possible to minimize a convex non-differentiable function using a fairly basic sub-gradient descent procedure:

- Initialize $\mathbf{x}_0 \in \mathbb{R}^D, f_{min} = \infty, \mathbf{x}_* = 0$
- For k from 1 to K:

Sub-gradient Descent

Despite the fact that not all sub-gradients yield descent directions, it is still possible to minimize a convex non-differentiable function using a fairly basic sub-gradient descent procedure:

- Initialize $\mathbf{x}_0 \in \mathbb{R}^D, f_{min} = \infty, \mathbf{x}_* = 0$
- For k from 1 to K:
  - Let $\mathbf{g}_k \in \partial f(\mathbf{x}_{k-1})$

Sub-gradient Descent

Despite the fact that not all sub-gradients yield descent directions, it is still possible to minimize a convex non-differentiable function using a fairly basic sub-gradient descent procedure:

- Initialize $\mathbf{x}_0 \in \mathbb{R}^D, f_{min} = \infty, \mathbf{x}_* = 0$
- For k from 1 to K:
  - Let $\mathbf{g}_k \in \partial f(\mathbf{x}_{k-1})$
  - Set $\mathbf{x}_k = \mathbf{x}_{k-1} - \alpha_k \mathbf{g}_k$

Sub-gradient Descent

Despite the fact that not all sub-gradients yield descent directions, it is still possible to minimize a convex non-differentiable function using a fairly basic sub-gradient descent procedure:

- Initialize $\mathbf{x}_0 \in \mathbb{R}^D$, $f_{min} = \infty$, $\mathbf{x}_* = 0$
- For k from 1 to K:
  - Let $\mathbf{g}_k \in \partial f(\mathbf{x}_{k-1})$
  - Set $\mathbf{x}_k = \mathbf{x}_{k-1} - \alpha_k \mathbf{g}_k$
  - If $f(\mathbf{x}_k) < f_{min}$ then set $f_{min} = f(\mathbf{x}_k)$ and $\mathbf{x}_* = \mathbf{x}_k$

Sub-gradient Descent

Despite the fact that not all sub-gradients yield descent directions, it is still possible to minimize a convex non-differentiable function using a fairly basic sub-gradient descent procedure:

- Initialize $\mathbf{x}_0 \in \mathbb{R}^D$, $f_{min} = \infty$, $\mathbf{x}_* = 0$
- For k from 1 to K:
  - Let $\mathbf{g}_k \in \partial f(\mathbf{x}_{k-1})$
  - Set $\mathbf{x}_k = \mathbf{x}_{k-1} - \alpha_k \mathbf{g}_k$
  - If $f(\mathbf{x}_k) < f_{min}$ then set $f_{min} = f(\mathbf{x}_k)$ and $\mathbf{x}_* = \mathbf{x}_k$
- Return $\mathbf{x}_*$

Sub-gradient Descent

Despite the fact that not all sub-gradients yield descent directions, it is still possible to minimize a convex non-differentiable function using a fairly basic sub-gradient descent procedure:

- Initialize $\mathbf{x}_0 \in \mathbb{R}^D$, $f_{min} = \infty$, $\mathbf{x}_* = 0$
- For k from 1 to K:
  - Let $\mathbf{g}_k \in \partial f(\mathbf{x}_{k-1})$
  - Set $\mathbf{x}_k = \mathbf{x}_{k-1} - \alpha_k \mathbf{g}_k$
  - If $f(\mathbf{x}_k) < f_{min}$ then set $f_{min} = f(\mathbf{x}_k)$ and $\mathbf{x}_* = \mathbf{x}_k$
- Return $\mathbf{x}_*$

Questions: How to choose α_k? How to choose K?

Subgradient Descent Convergence

- Line search is typically not used for sub-gradient descent procedures. It is more common to use a fixed sequence of step sizes.

Subgradient Descent Convergence

- Line search is typically not used for sub-gradient descent procedures. It is more common to use a fixed sequence of step sizes.
- Common step size rules include $\alpha_k = \alpha/(\beta + k)$ or $\alpha_k = \alpha/\sqrt{k}$.

Subgradient Descent Convergence

- Line search is typically not used for sub-gradient descent procedures. It is more common to use a fixed sequence of step sizes.
- Common step size rules include $\alpha_k = \alpha/(\beta + k)$ or $\alpha_k = \alpha/\sqrt{k}$.
- Under either of these step size rules, we have that the sequence of subgradient descent iterates $\mathbf{x}_k$ satisfies:

$$\lim_{k\to\infty} f(\mathbf{x}_k) = \min_{\mathbf{x}} f(\mathbf{x})$$

Subgradient Descent with Momentum

- Initialize $\mathbf{x}_0 \in \mathbb{R}^D$, $f_{min} = \infty$, $\mathbf{x}_* = 0$
- For k from 1 to K:
  - Let $\mathbf{g}_k \in \partial f(\mathbf{x}_{k-1})$
  - Let $\mathbf{d}_k = \gamma \mathbf{d}_{k-1} + \alpha_k \mathbf{g}_k$
  - Set $\mathbf{x}_k = \mathbf{x}_{k-1} - \mathbf{d}_k$
  - If $f(\mathbf{x}_k) < f_{min}$ then set $f_{min} = f(\mathbf{x}_k)$ and $\mathbf{x}_* = \mathbf{x}_k$
- Return $\mathbf{x}_*$

Typically use with $0 < \gamma < 1$. $\gamma = 0.9$ is a common choice.

Nesterov Accelerated Subgradient Descent

- Initialize $\mathbf{x}_0 \in \mathbb{R}^D$, $f_{min} = \infty$, $\mathbf{x}_* = 0$
- For k from 1 to K:
  - Let $\mathbf{g}_k \in \partial f(\mathbf{x}_{k-1} - \gamma \mathbf{d}_{k-1})$
  - Let $\mathbf{d}_k = \gamma \mathbf{d}_{k-1} + \alpha_k \mathbf{g}_k$
  - Set $\mathbf{x}_k = \mathbf{x}_{k-1} - \mathbf{d}_k$
  - If $f(\mathbf{x}_k) < f_{min}$ then set $f_{min} = f(\mathbf{x}_k)$ and $\mathbf{x}_* = \mathbf{x}_k$
- Return $\mathbf{x}_*$

Typically use with $0 < \gamma < 1$. $\gamma = 0.9$ is a common choice.

Outline

Example: Finding a Subgradient for SVC

- We begin the risk function:

$$
\begin{aligned}
R(\theta, \mathcal{D}) &= C\sum_{n=1}^{N} \max(0, 1 - y_n g_\theta(\mathbf{x}_n)) + ||\mathbf{w}||_2^2 \\
&= C\sum_{n=1}^{N} L(y_n g_\theta(\mathbf{x}_n)) + ||\mathbf{w}||_2^2
\end{aligned}
$$

Example: Finding a Subgradient for SVC

- We begin the risk function:

$$R(\theta, \mathcal{D}) = C\sum_{n=1}^{N}\max(0, 1 - y_n g_\theta(\mathbf{x}_n)) + ||\mathbf{w}||_2^2$$
$$= C\sum_{n=1}^{N} L(y_n g_\theta(\mathbf{x}_n)) + ||\mathbf{w}||_2^2$$

- We are looking for a vector **h** such that:

$$\mathbf{h} \in \partial R(\theta, \mathcal{D})$$

Example: Finding a Subgradient for SVC

- By the additivity and scaling properties, we have that if $\mathbf{h}_n \in \partial L(y_n g_\theta(\mathbf{x}_n))$ and $\mathbf{h}_w \in \partial||\mathbf{w}||_2^2$, then:

$$C \sum_{n=1}^{N} \mathbf{h}_n + \mathbf{h}_w \in \partial R(\theta, \mathcal{D})$$

Example: Finding a Subgradient for SVC

- By the additivity and scaling properties, we have that if $\mathbf{h}_n \in \partial L(y_n g_\theta(\mathbf{x}_n))$ and $\mathbf{h}_w \in \partial||\mathbf{w}||_2^2$, then:

$$C\sum_{n=1}^{N}\mathbf{h}_n + \mathbf{h}_w \in \partial R(\theta, \mathcal{D})$$

- We will proceed by finding suitable vectors $\mathbf{h}_n$ and $\mathbf{h}_w$ using properties of sub-gradients.

Example: Finding a Subgradient for SVC

- Consider $h_w \in \partial||\mathbf{w}||_2^2$.

Example: Finding a Subgradient for SVC

- Consider $h_w \in \partial ||\mathbf{w}||_2^2$.
- Since $||\mathbf{w}||_2^2$ is a differentiable function with gradient $2\mathbf{w}$, we have that $h_w = 2[\mathbf{w}; 0]$.

Example: Finding a Subgradient for SVC

- Consider $\mathbf{h}_n \in \partial L(y_n g_\theta(\mathbf{x}_n))$. Recall $L(z) = \max(0, 1 - z)$ and $g_\theta(\mathbf{x}_n) = \mathbf{x}_n \theta$.

Example: Finding a Subgradient for SVC

- Consider $\mathbf{h}_n \in \partial L(y_n g_\theta(\mathbf{x}_n))$. Recall $L(z) = \max(0, 1 - z)$ and $g_\theta(\mathbf{x}_n) = \mathbf{x}_n\theta$.
- By the chain rule, we have that $\mathbf{h}_n \in \partial L(y_n \mathbf{x}_n \theta)$ if and only if:

$$\mathbf{h}_n \in \{k y_n \mathbf{x}_n^T \mid k \in \partial L(z), z = y_n \mathbf{x}_n \theta\}$$

Example: Finding a Subgradient for SVC

- Consider $\mathbf{h}_n \in \partial L(y_n g_\theta(\mathbf{x}_n))$. Recall $L(z) = \max(0, 1 - z)$ and $g_\theta(\mathbf{x}_n) = \mathbf{x}_n\theta$.
- By the chain rule, we have that $\mathbf{h}_n \in \partial L(y_n\mathbf{x}_n\theta)$ if and only if:

$$\mathbf{h}_n \in \{k y_n \mathbf{x}_n^T \mid k \in \partial L(z), z = y_n\mathbf{x}_n\theta\}$$

- We next need to figure out what $\partial L(z)$ is.

Example: Finding a Subgradient for SVC

- Consider $\partial L(z) = \partial \max(0, 1 - z)$. This is a non-differentiable convex function with one point of non-differentiability at $z = 1$.

Example: Finding a Subgradient for SVC

- Consider $\partial L(z) = \partial \max(0, 1 - z)$. This is a non-differentiable convex function with one point of non-differentiability at $z = 1$.
- For $z > 1$ the functions is constant so $\partial L(z) = \{0\}$ for $z > 1$

Example: Finding a Subgradient for SVC

- Consider $\partial L(z) = \partial \max(0, 1 - z)$. This is a non-differentiable convex function with one point of non-differentiability at $z = 1$.
- For $z > 1$ the functions is constant so $\partial L(z) = \{0\}$ for $z > 1$
- For $z < 1$ the functions is $1 - z$ so $\partial L(z) = \{-1\}$ for $z < 1$.

Example: Finding a Subgradient for SVC

- Consider $\partial L(z) = \partial \max(0, 1-z)$. This is a non-differentiable convex function with one point of non-differentiability at $z = 1$.
- For $z > 1$ the functions is constant so $\partial L(z) = \{0\}$ for $z > 1$
- For $z < 1$ the functions is $1 - z$ so $\partial L(z) = \{-1\}$ for $z < 1$.
- At $z = 0$ we have $\partial L(z) = [-1, 0]$ following subdifferential rules for 1-D convex functions.

Example: Finding a Subgradient for SVC

- Consider $\partial L(z) = \partial \max(0, 1-z)$. This is a non-differentiable convex function with one point of non-differentiability at $z = 1$.
- For $z > 1$ the functions is constant so $\partial L(z) = \{0\}$ for $z > 1$
- For $z < 1$ the functions is $1 - z$ so $\partial L(z) = \{-1\}$ for $z < 1$.
- At $z = 0$ we have $\partial L(z) = [-1, 0]$ following subdifferential rules for 1-D convex functions.
- We choose the following sub-gradient function for the hinge loss:

$$L'(z) = \begin{cases} 0 & \ldots\ z \geq 1 \\ -1 & \ldots\ z < 1 \end{cases}$$

Example: Finding a Subgradient for SVC

- Consider again, $\mathbf{h}_n \in \partial L(y_n \mathbf{x}_n \theta)$ if and only if:

$$\mathbf{h}_n \in \{k y_n \mathbf{x}_n^T \mid k \in \partial L(z), z = y_n \mathbf{x}_n \theta\}$$

Example: Finding a Subgradient for SVC

- Consider again, $\mathbf{h}_n \in \partial L(y_n \mathbf{x}_n \theta)$ if and only if:

$$\mathbf{h}_n \in \{k y_n \mathbf{x}_n^T \mid k \in \partial L(z), z = y_n \mathbf{x}_n \theta\}$$

- We have shown that $L'(z) \in \partial L(z)$, so we have that:

$$\mathbf{h}_n = L'(y_n \mathbf{x}_n \theta) y_n \mathbf{x}_n^T \in \partial L(y_n \mathbf{x}_n \theta)$$

Example: Finding a Subgradient for SVC

- Consider again, $\mathbf{h}_n \in \partial L(y_n\mathbf{x}_n\theta)$ if and only if:

$$\mathbf{h}_n \in \{k y_n \mathbf{x}_n^T \mid k \in \partial L(z), z = y_n\mathbf{x}_n\theta\}$$

- We have shown that $L'(z) \in \partial L(z)$, so we have that:

$$\mathbf{h}_n = L'(y_n\mathbf{x}_n\theta) y_n \mathbf{x}_n^T \in \partial L(y_n\mathbf{x}_n\theta)$$

- This gives us a final answer for a subgradient of the risk:

$$\mathbf{h} = C \sum_{n=1^N} L'(y_n\mathbf{x}_n\theta) y_n \mathbf{x}_n^T + 2[\mathbf{w}; 0]$$