COMPSCI 689
Lecture 9: Empirical Risk Minimization and Classification

Benjamin M. Marlin

College of Information and Computer Sciences
University of Massachusetts Amherst

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

Expected Risk Minimization

The ERM framework for parametric prediction models corresponds to finding the optimal prediction function within a set of functions by minimizing it's expected loss under the empirical distribution. The expected loss of the function is called it's risk.

Expected Risk Minimization

$$f_* = \min_{f \in F(\Theta)} \frac{1}{N} \sum_{n=1}^{N} L(\mathbf{y}_n, f(\mathbf{x}_n))$$

$$\theta_* = \min_{\theta \in \Theta} \frac{1}{N} \sum_{n=1}^{N} L(\mathbf{y}_n, f(\mathbf{x}_n, \theta))$$

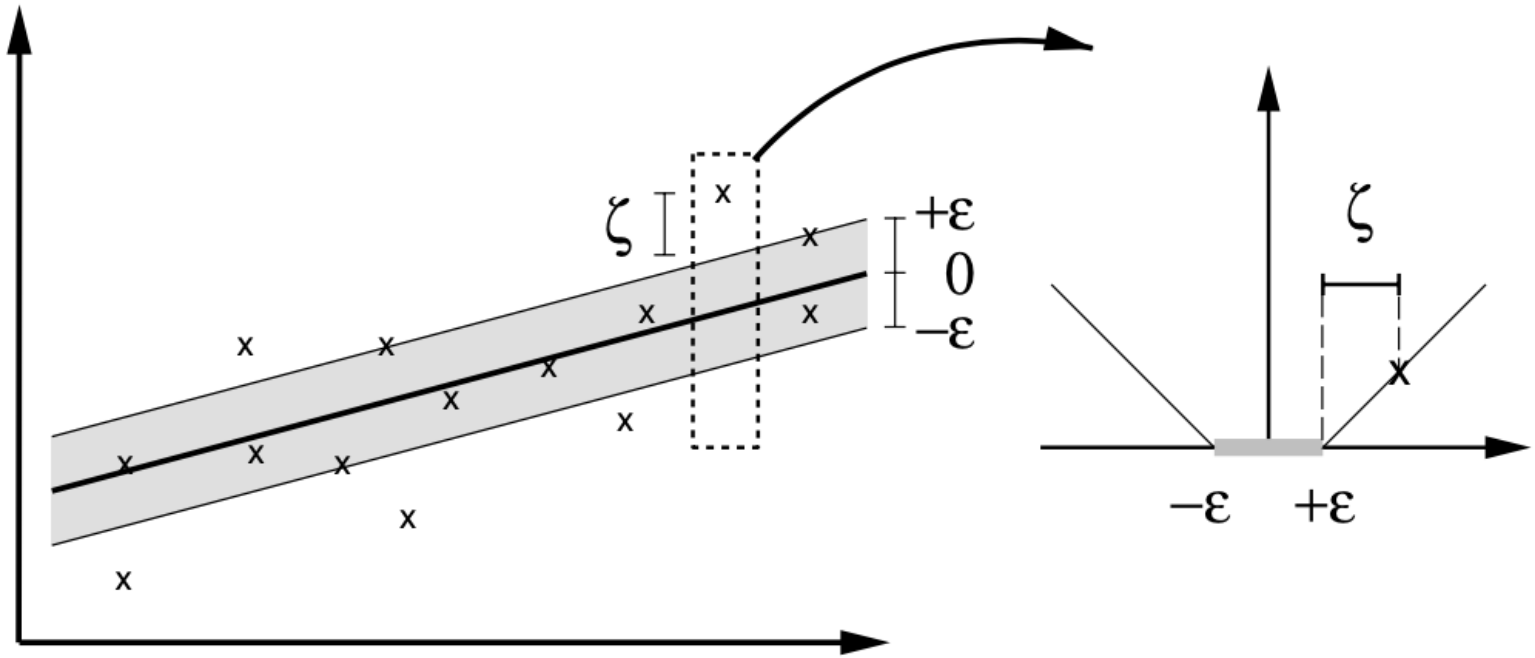
Support Vector Regression

$$f_{SVR}(\mathbf{x}, \mathbf{w}) = \left(\sum_{d=1}^{D} w_d x_d \right) = \mathbf{w}\mathbf{x}^T$$

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

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

Support Vector Regression



This model is called *support* vector regression because only the data cases that fall outside of the epsilon insensitive region determine the model parameters. The optimal parameters are thus *supported* by a subset of all data instances.

ERM For Classification

- In the classification setting, the loss we are most often interested is the zero-one loss, or classification error.
- This leads to the following optimization problem:

$$f_* = \arg\min_f \sum_{n=1}^{N} [y_n \neq f(\mathbf{x}_n)]$$

Losses For Classification

- In general, optimizing the zero-one loss directly is computationally intractable since this function is not continuous, differentiable, or convex.
- In addition, a model $f(\mathbf{x}, \theta)$ that directly predicts discrete outputs from $\mathcal{Y}$ will generally include a non-differentiable, non-convex component.
- To make ERM easier in the classification setting, it is more common to optimize differentiable (or at least continuous convex losses) that upper-bound the zero-one loss.
- In binary classification, it is also common to learn a discrimination function $f(\mathbf{x}, \theta)$ with real-valued outputs where the positive class is predicted if $f(\mathbf{x}, \theta) \geq 0$ and the negative class is predicted otherwise.

Logistic Regression as ERM

- In the case of binary logistic regression, we select the model parameters by maximizing the conditional log likelihood function:

$$\sum_{i=1}^{n} \log P(Y = y_i | \mathbf{X} = \mathbf{x}_i, \mathbf{w})$$

- Under the assumption that the labels take the values $\{-1, 1\}$, we have shown that this is equivalent to minimizing the function:

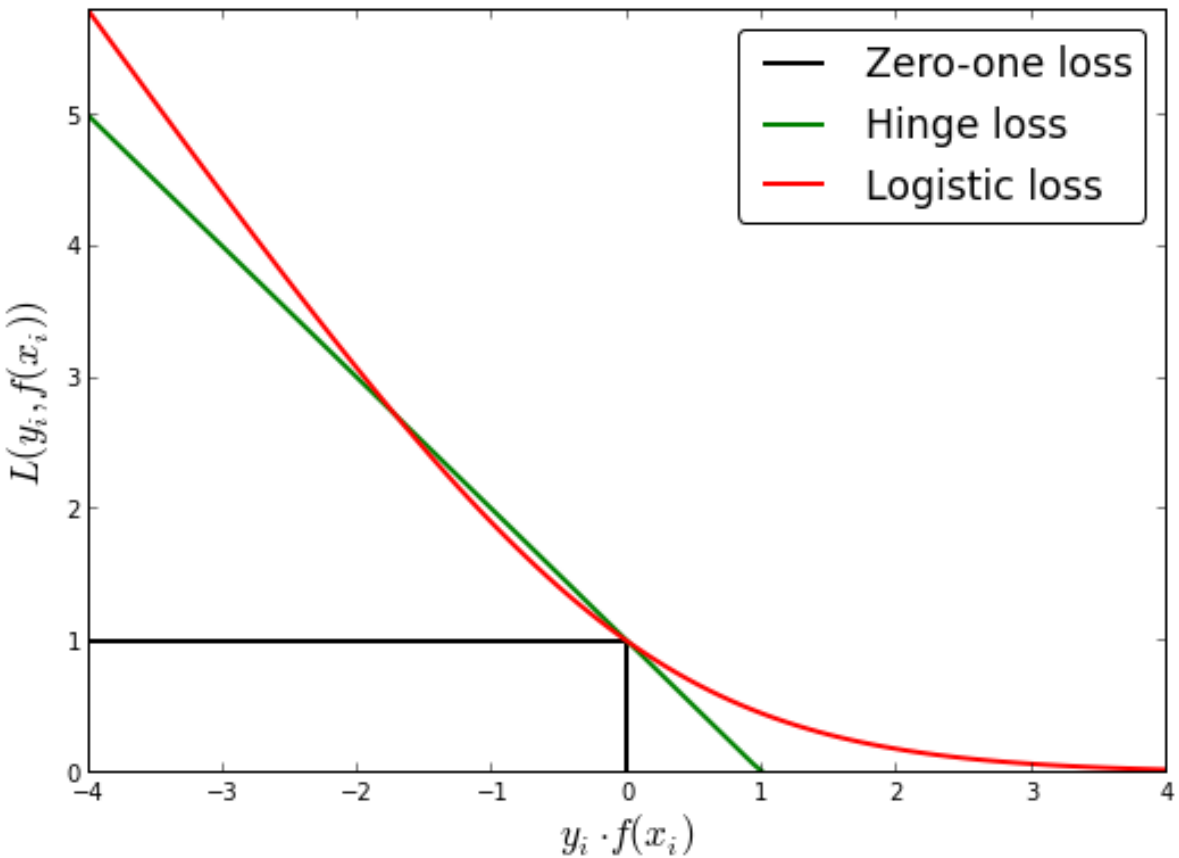
$$\sum_{i=1}^{n} \log(1 + \exp(-y_i \mathbf{w}^T \mathbf{x}))$$

- The function $L_{\log}(y, z) = \log(1 + \exp(-yz))$ is known as the logistic loss or log-loss. We can thus view the MLE for logistic regression as an instance of ERM, and regularized logistic regression as an instance of RRM.

Hinge Loss

- As in the case of linear regression, we are free to choose any loss we like in the ERM-based classification framework.
- The logistic loss is convex, differentiable, and provides an upper bound on the zero-one loss.
- The hinge loss is an alternate loss that is convex and non-differentiable, but provides a similar upper bound on the zero-one loss: $L_h(y, z) = \max(0, 1 - yz)$

Classification Losses



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

Support Vector Classifier

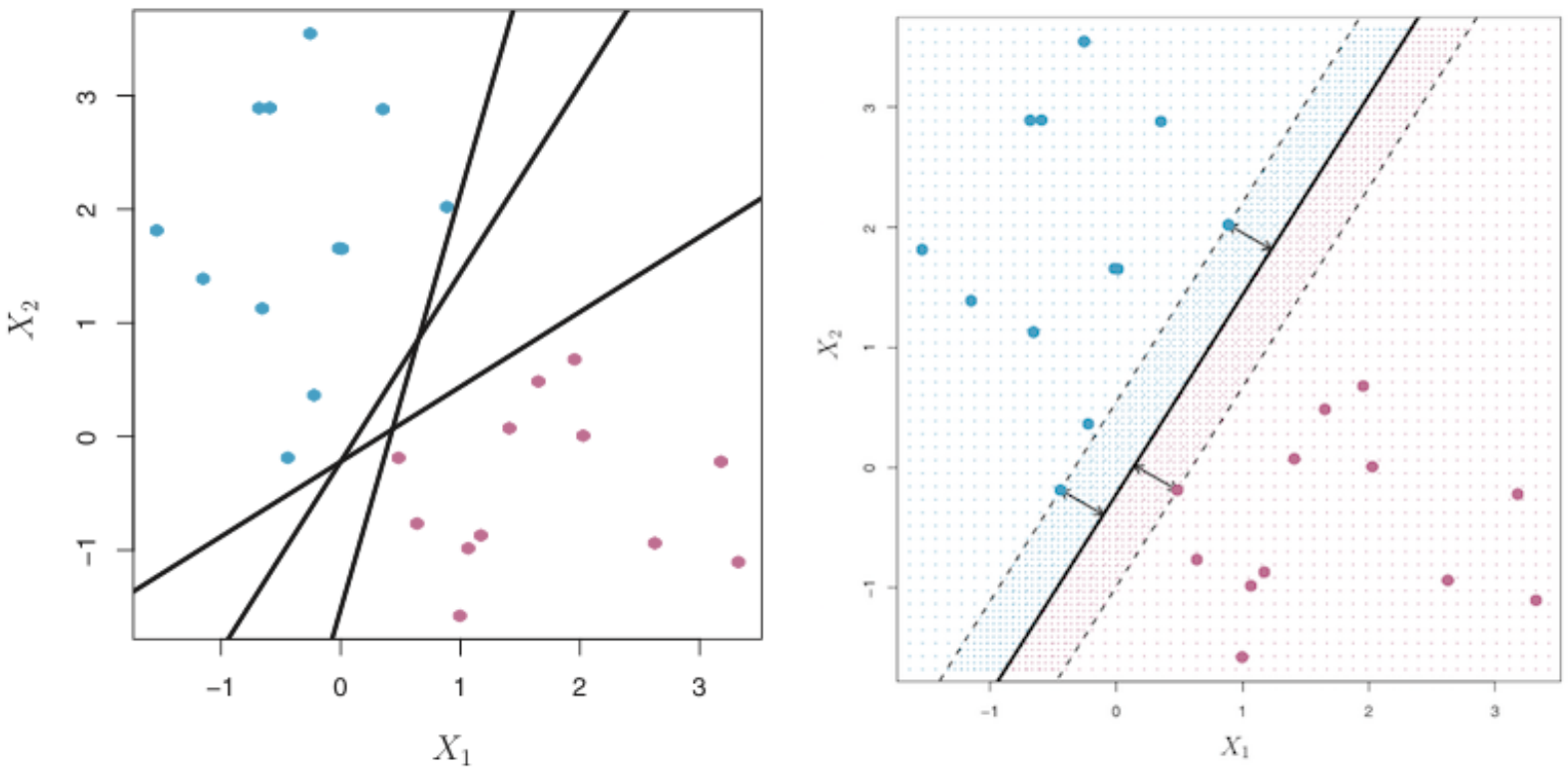
$$f_{SVC}(\mathbf{x}, \mathbf{w}) = \left(\sum_{d=1}^{D} w_d x_d \right) = \mathbf{w}\mathbf{x}^T$$

$$\mathbf{w}_* = \arg\min_{\mathbf{w}} C \sum_{n=1}^{N} L_h(y_n, \mathbf{w}\mathbf{x}_n^T) + ||\mathbf{w}||_2^2$$

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

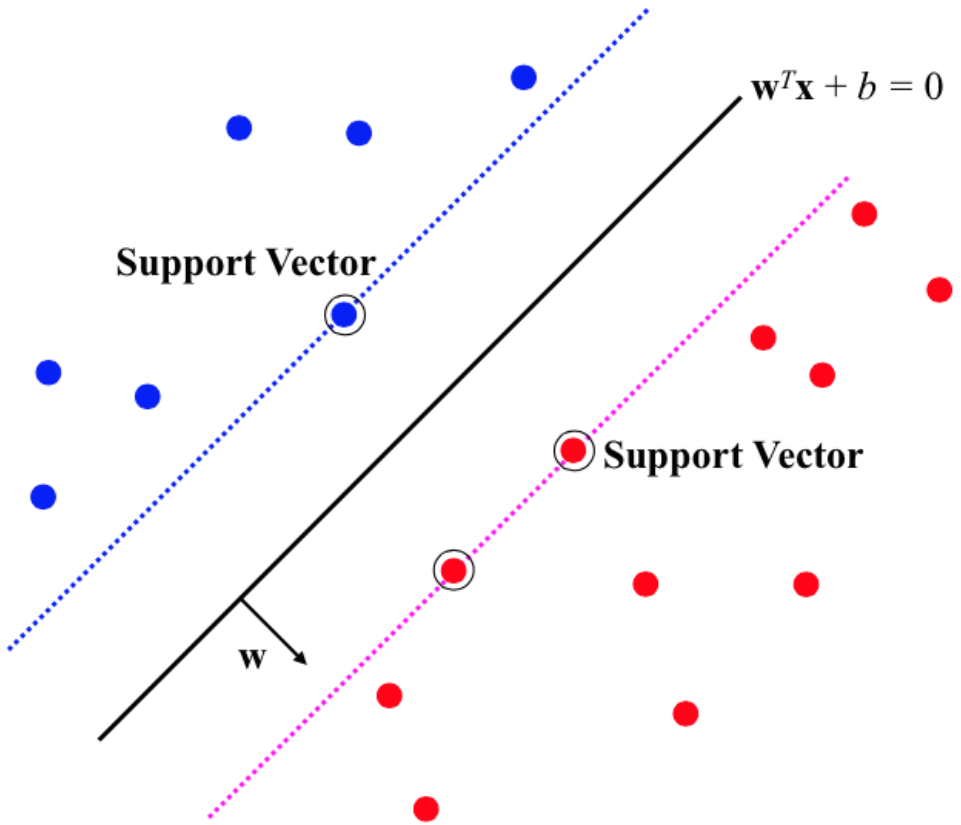
Maximum Margin Property

Part of popularity 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.



A Problem

The losses that correspond to likelihoods are convex and differentiable. The epsilon insensitive loss and the hinge loss are convex, but not differentiable. This breaks our existing optimization framework.

We need tools for optimizing convex, non-differentiable functions to learn these models.

Global Optimality

- 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.
- 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.

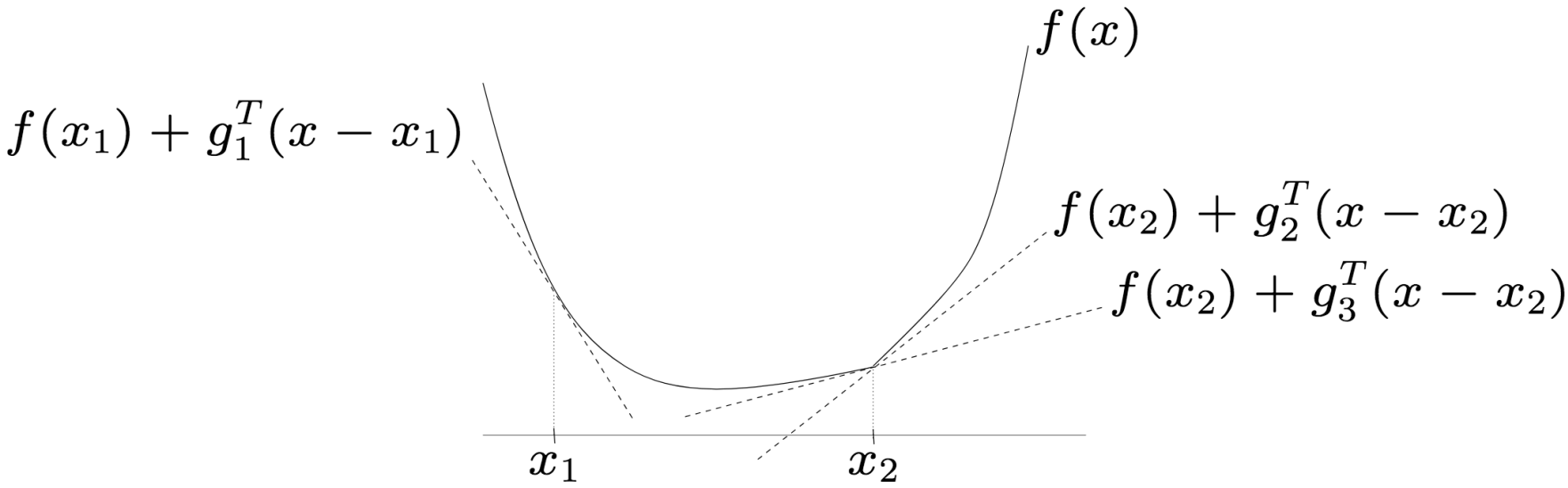
Subgradient

- 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 $\mathbf{x} \in \mathbb{R}^D$:

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

- 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

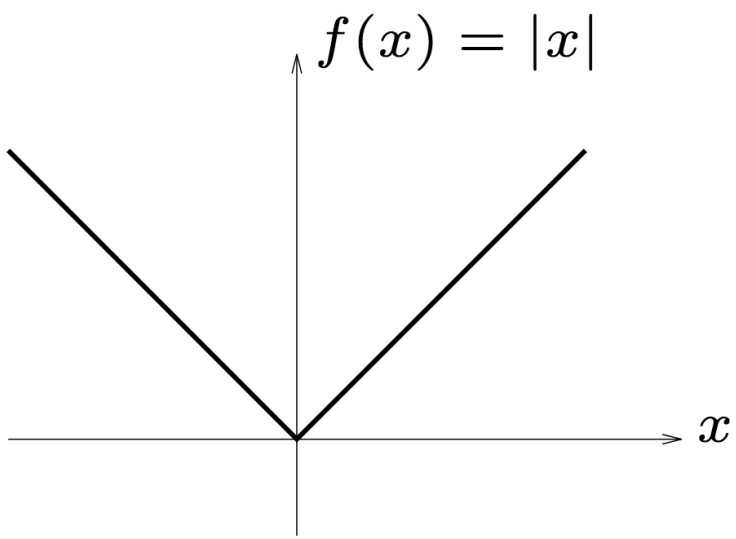


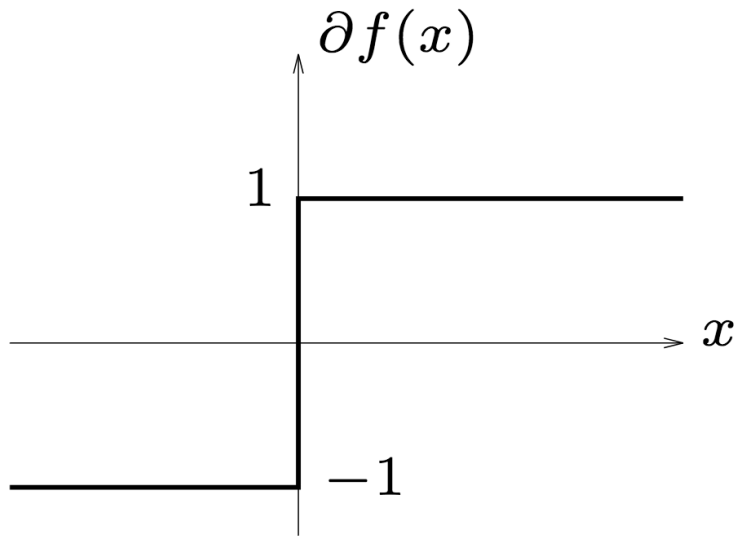
In this example, $\mathbf{g}_1$ is the unique subgradient of f at $\mathbf{x}_1$. Due to f being non-differentiable at $\mathbf{x}_2$, both $\mathbf{g}_2$ and $\mathbf{g}_3$ are subgradients of f at $\mathbf{x}_2$.

Subdifferentials

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





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 $g \in [-1, 1]$.

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 of Subdifferentials

The subdifferential operator satisfies a partial linearity property:

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

These properties can be used to determine the subdifferentials of more complex functions by reducing them to linear combinations of simpler functions.

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 the Global Minimum

- If f is differentiable at $\mathbf{x}$ and $\mathbf{0} \notin \partial f(\mathbf{x})$, then for all $\mathbf{g} \in \partial f(\mathbf{x})$, $-\mathbf{g}$ is a descent direction.
- 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.

Subgradient 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 $\mathbf{x}_* = \mathbf{x}_k$

Subgradient Descent Convergence

- Line search is typically not used for sub-gradient descent procedures. It is more common to used 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 stepsize rules, we have that:

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

Example: Sub-GD for ABS

- Consider the optimization problem shown below. Let us derive a subgradient descent method for this objective.

$$\mu^* = \arg\min_{\mu} \sum_{n=1}^{N} \|\mu - \mathbf{x}_n\|_1$$