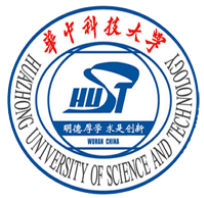
# Empirical Risk Minimization (ERM)

Kun He (何琨)

Data Mining and Machine Learning Lab
(John Hopcroft Lab)
Huazhong University of Science & Technology

*brooklet60@hust.edu.cn*

2022年5月



# Table of Contents

# Table of Contents

# Recap

Remember the unconstrained SVM Formulation

$$\min_{\mathbf{w}} \underbrace{C\sum_{i=1}^{n} \max[1 - y_i\underbrace{(w^\top \mathbf{x}_i + b)}_{h(\mathbf{x}_i)}, 0]}_{Hinge-Loss} + \underbrace{\|w\|_z^2}_{l_2-Regularizer}$$

The hinge loss is the SVM's error function of choice, whereas the $\ell 2$-regularizer reflects the complexity of the solution, and penalizes complex solutions. This is an example of empirical risk minimization with a loss function $\ell$ and a regularizer $r$,

$$\min_{\mathbf{w}} \frac{1}{n}\sum_{i=1}^{n} \underbrace{l(h_{\mathbf{w}}(\mathbf{x}_i), y_i)}_{Loss} + \underbrace{\lambda r(w)}_{Regularizer},$$

where the loss function is a continuous function which penalizes training error, and the regularizer is a continuous function which penalizes classifier complexity. Here, we define $\lambda$ as $\frac{1}{C}$ from the previous lecture.

# Background

- Consider the following situation, a general setting of many supervised learning problems.
- We have two spaces of objects $X$ and $Y$ and would like to learn a function $h : X \to Y$ which outputs an object $y \in Y$ , given $x \in X$.
- To do so, we have at our disposal a training set of $n$ examples $(x_1, y_1), ..., (x_n, y_n)$ where $x_i \in X$ is an input and $y_i \in Y$ is the corresponding response that we wish to get from $h(x_i)$.
- To put it more formally, we assume that there is a **joint probability distribution** $p(x, y)$ over $X$ and $Y$, and that the training set consists of $n$ instances $(x_1, y_1), ..., (x_n, y_n)$ drawn i.i.d. from $p(x, y)$.
- Note that the assumption of a joint probability distribution allows us to model uncertainty in predictions because $y$ is not a deterministic function of $x$, but rather a random variable with conditional distribution $p(x|y)$ for a fixed $x$.

# Background

We also assume that we are given a **non-negative real-valued loss function** $L(\widehat{y}, y)$ which measures how different the prediction $\widehat{y}$ of a hypothesis is from the true outcome $y$. The risk associated with hypothesis $h(x)$ is then defined as the expectation of the loss function:

$$R(h) = E[L(h(x), y)] = \int L(h(x), y) \, \mathrm{d} P(x, y)$$

A loss function commonly used in theory is the 0-1 loss function: $L(\widehat{y}, y) = I(\widehat{y} \neq y)$, where $I(\cdot)$ is the indicator notation.

The ultimate goal of a learning algorithm is to find a hypothesis $h^*$ among a fixed class of functions $H$ for which the risk $R(h)$ is minimal:

$$h^* = arg\min_{h \in H} R(h).$$

# Empirical Risk Minimization

In general, the risk $R(h)$ cannot be computed because the distribution $P(x, y)$ is unknown to the learning algorithm (this situation is referred to as agnostic learning). However, we can compute an approximation, called empirical risk, by averaging the loss function on the training set:

$$R_{emp}(h) = \frac{1}{n} \sum_{i=1}^{n} L(h(x_i), y_i).$$

The empirical risk minimization principle states that the learning algorithm should choose a hypothesis $\widehat{h}$ which minimizes the empirical risk:

$$h^* = arg\min_{h \in H} R(h).$$

Thus **Empirical Risk Minimization (ERM)** is a principle in statistical learning theory which defines a family of learning algorithms and is used to give theoretical bounds on their performance.

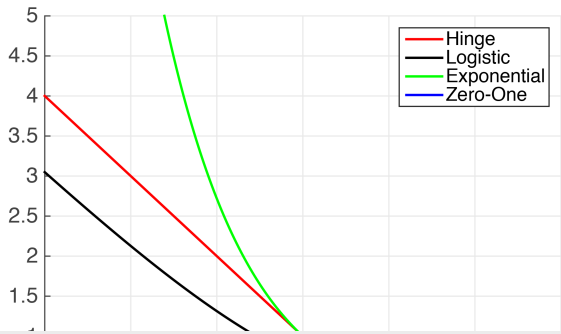
# Table of Contents

# Binary Classification Loss Functions

Different Machine Learning algorithms use different loss functions

Table 1: Binary Classification Loss Functions, $y \in \{-1, +1\}$

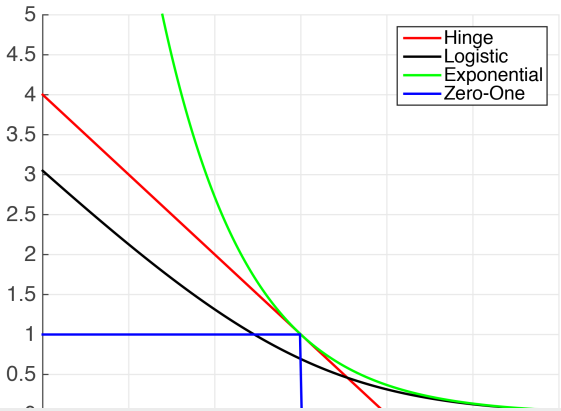
| **Loss** $\ell(h_{\mathbf{w}}(\mathbf{x}_i, y_i))$ | **Usage** | **Comments** |
|---|---|---|
| **Hinge-Loss:** $\max[1 - h_{\mathbf{w}}(\mathbf{x}_i)y_i, 0]^p$ | • Standard SVM( $p = 1$ )<br>• (Differentiable) Squared Hingeless SVM ( $p = 2$ ) | When used for Standard SVM, the loss function denotes the size of the margin between linear separator and its closest points in either class. Only differentiable everywhere with $p = 2$. |
| **Log-Loss:** $\log(1 + e^{-h_{\mathbf{w}}(\mathbf{x}_i)y_i})$ | Logistic Regression | One of the most popular loss functions in Machine Learning, since its outputs are well-calibrated probabilities. |



# Binary Classification Loss Functions

Table 1: Binary Classification Loss Functions, $y \in \{-1, +1\}$

| **Loss** $\ell(h_{\mathbf{w}}(\mathbf{x}_i, y_i))$ | **Usage** | **Comments** |
|---|---|---|
| **Exponential Loss:** $e^{-h_{\mathbf{w}}(\mathbf{x}_i)y_i}$ | AdaBoost | This function is very aggressive. The loss of a mis-prediction increases *exponentially* with the value of $-h_{\mathbf{w}}(\mathbf{x}_i)y_i$. This can lead to nice convergence results, for example in the case of Adaboost, but it can also cause problems with noisy data. |
| **Zero-One Loss:** $\delta(\text{sign}(h_{\mathbf{w}}(\mathbf{x}_i)) \neq y_i)$ | Actual Classification Loss | Non-continuous and thus impractical to optimize. |



# Binary Classification Loss Functions

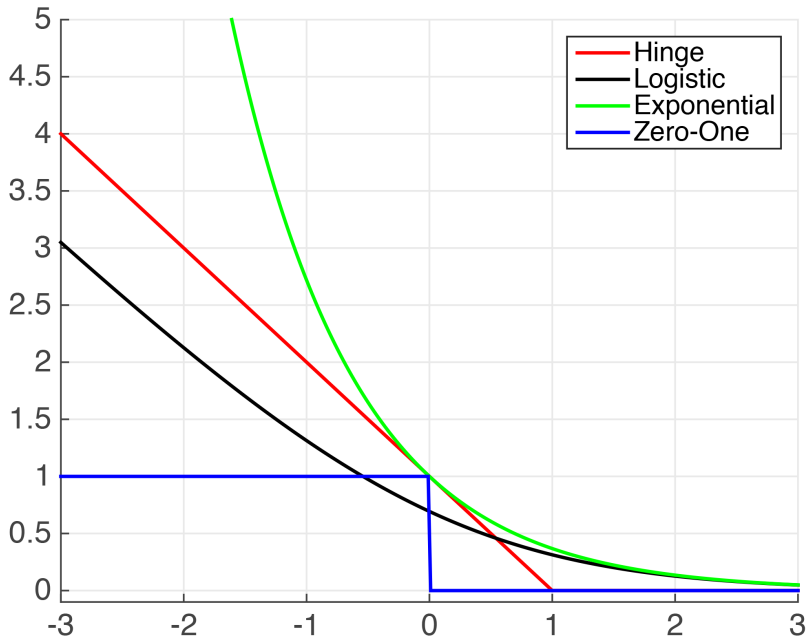



Figure 1: Plots of Common Classification Loss Functions. x-axis: $h(\mathbf{x}_i)y_i$ , or "correctness" of prediction; y-axis: loss value

**Quiz:** Which functions are strict upper bounds on the 0/1-loss?

**Quiz:** What can you say about the hinge-loss and the log-loss as $z \to -\infty$ ?

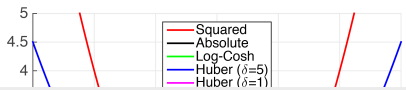
# Table of Contents

# Regression Loss Functions

Loss for regression algorithms (a prediction can lie anywhere on the real-number line):

Table 2: Regression Loss Functions, $y \in \mathbb{R}$

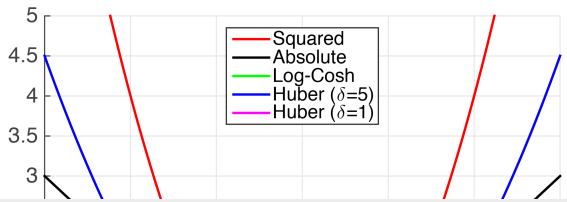
| **Loss** $\ell(h_{\mathbf{w}}(\mathbf{x}_i, y_i))$ | **Comments** |
|---|---|
| **Squared Loss:** $(h(\mathbf{x}_i) - y_i)^2$ | • Most popular regression loss function<br>• Estimates Mean Label<br>• Also known as Ordinary Least Squares (OLS)<br>• DISADVANTAGE: Somewhat sensitive to outliers/noise |
| **Absolute Loss:** $\lvert h(\mathbf{x}_i) - y_i \rvert$ | • Also a very popular loss function<br>• Estimates Median Label<br>• ADVANTAGE: Less sensitive to noise<br>• DISADVANTAGE: Not differentiable at 0 |



# Regression Loss Functions

Table 2:Regression Loss Functions, $y \in \mathbb{R}$

| **Loss** $\ell(h_{\mathbf{w}}(\mathbf{x}_i, y_i))$ | **Comments** |
|---|---|
| **Huber Loss:**<br>• $\frac{1}{2}\left(h(\mathbf{x}_i) - y_i\right)^2$ if $\lvert h(\mathbf{x}_i) - y_i \rvert < \delta$,<br>• otherwise $\delta(\lvert h(\mathbf{x}_i) - y_i \rvert - \frac{\delta}{2})$ | • Also known as Smooth Absolute Loss<br>• Once-differentiable<br>• Takes on behavior of Squared-Loss when loss is small, and Absolute Loss when loss is large.<br>• ADVANTAGE: "Best of Both Worlds" of Squared and Absolute Loss |
| **Log-Cosh Loss:**<br>$log(cosh(h(\mathbf{x}_i) - y_i))$,<br>$cosh(x) = \frac{e^x + e^{-x}}{2}$ | ADVANTAGE: Similar to Huber Loss, but twice differentiable everywhere |



# Regression Loss Functions

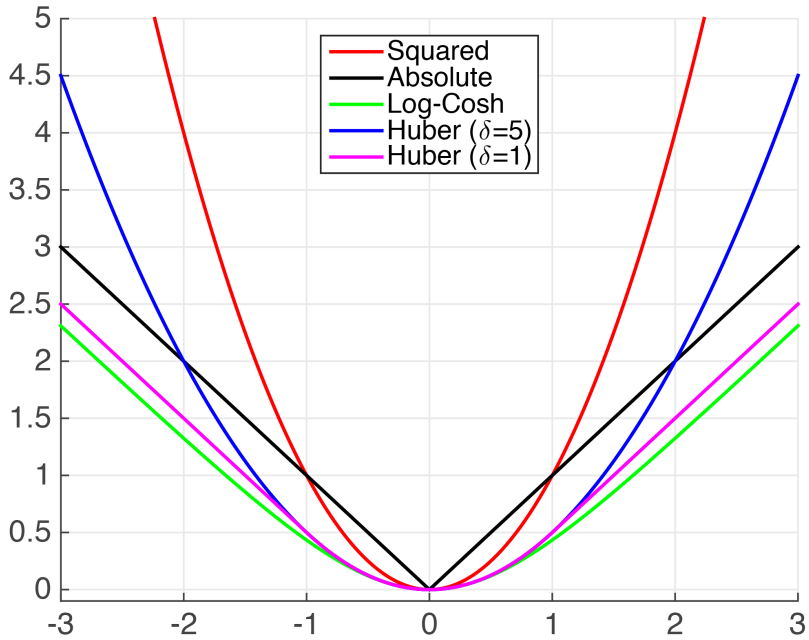



Figure 2: Plots of Common Regression Loss Functions - x-axis: $h(\mathbf{x}_i)y_i$ , or "error" of prediction; y-axis: loss value

**Quiz:** What do the loss functions in Table 2 look like with respect to $z = h(\mathbf{x}_i) - y_i$ ?

# Table of Contents

# Regularizers

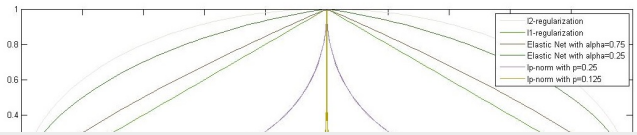
- In mathematics, statistics, and computer science, particularly in machine learning problems, regularization is the process of adding information in order to solve an ill-posed problem or to prevent overfitting.
- When we look at regularizers it helps to change the formulation of the optimization problem to obtain a better geometric intuition.
- In previous sections, $l_2$-regularizer has been introduced as the component in SVM that reflects the complexity of solutions.

Besides the $l_2$-regularizer, other types of useful regularizers and their properties are listed in the following Table.

# Regularizers

Table 3:Types of Regularizers

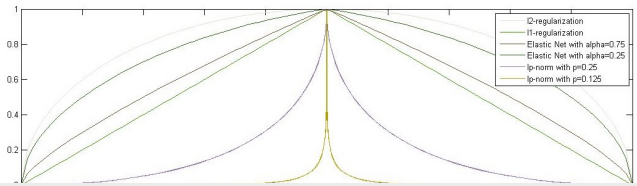
| **Regularizer** $r(\mathbf{w})$ | **Properties** |
|---|---|
| $l_2$-Regularization<br>$r(\mathbf{w}) = \mathbf{w}^\top \mathbf{w} = \|\mathbf{w}\|_2^2$ | • ADVANTAGE: Strictly Convex;Differentiable<br>• DISADVANTAGE: Uses weights on all features, i.e. relies on all features to some degree (ideally we would like to avoid this) - these are known as Dense Solutions. |
| $l_1$-Regularization<br>$r(\mathbf{w}) = \|\mathbf{w}\|_1$ | • Convex (but not strictly)<br>• DISADVANTAGE: Not differentiable at 0 (the point which minimization is intended to bring us to<br>• Effect: Sparse (i.e. not Dense) Solutions |



# Regularizers

Table 3:Types of Regularizers

| **Regularizer** $r(\mathbf{w})$ | **Properties** |
|---|---|
| $l_p$-Norm $\|\mathbf{w}\|_p = (\sum_{i=1}^{d} v_i^p)^{1/p}$ | • (often $0 < p \leq 1$)<br>• Initialization dependent<br>• ADVANTAGE: Very sparse solutions<br>• DISADVANTAGE: Non-convex;Not differentiable |
| **Elastic Net:** $\alpha\|\mathbf{w}\|_1 + (1-\alpha)\|\mathbf{w}\|_2^2$<br>$\alpha \in [0, 1)$ | • ADVANTAGE: Strictly convex (i.e. unique solution)<br>• ADVANTAGE: Sparsity inducing (good for feature selection);Dual of squared-loss SVM, see SVEN<br>• DISADVANTAGE: Non-differentiable |



# Regularizers

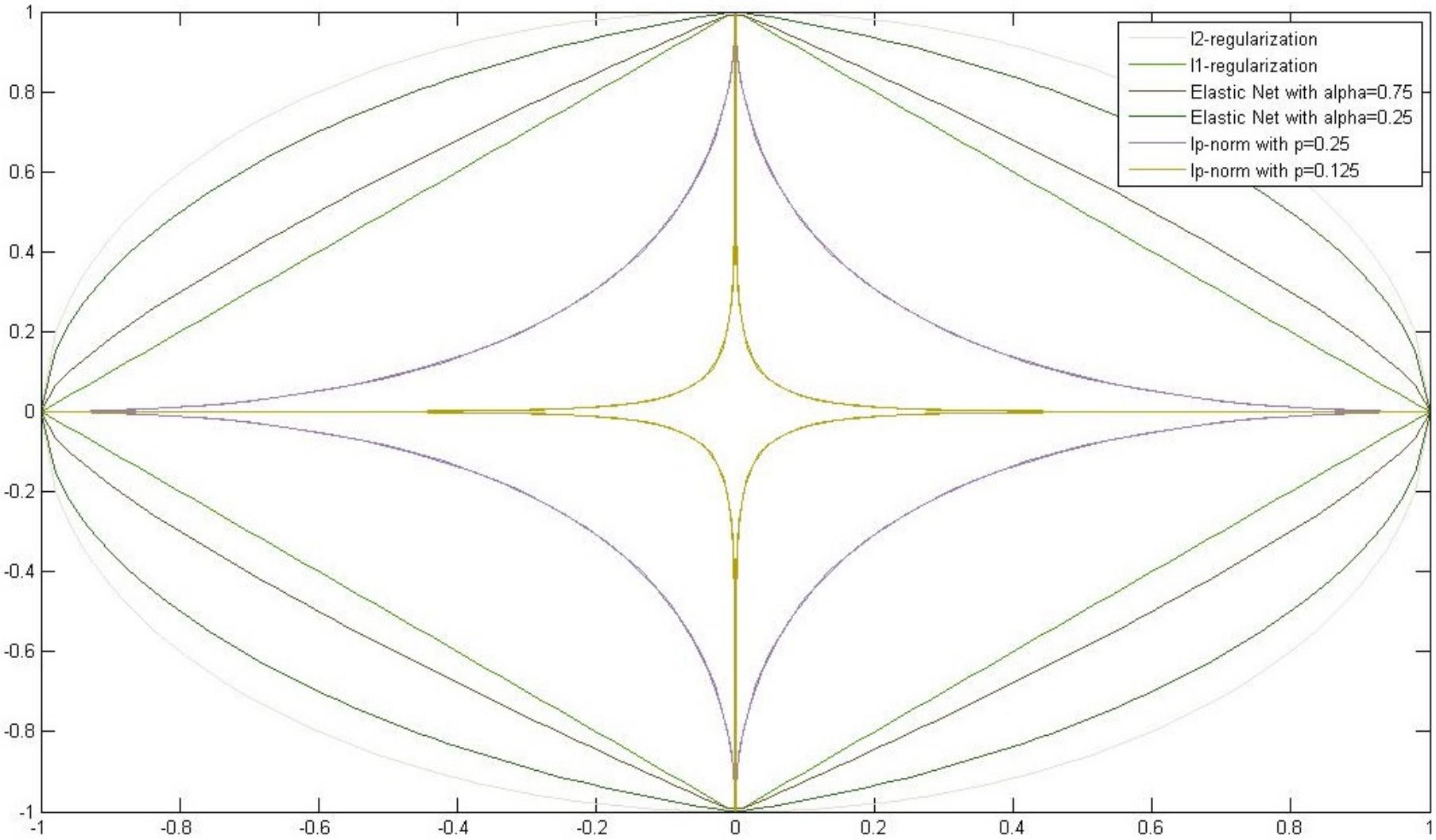



Figure 3: Plots of Common Regularizers

# Table of Contents

# Famous Special Cases

This section includes several special cases that deal with risk minimization, such as Ordinary Least Squares, Ridge Regression, Lasso, and Logistic Regression. The following blocks provide information on their loss functions, regularizers, as well as solutions.

## Ordinary Least Squares

In statistics, ordinary least squares (OLS) is a type of linear least squares method for estimating the unknown parameters in a linear regression model.

**goal:**

$$\min_{\mathbf{w}} \frac{1}{n} \sum_{i=1}^{n} (\mathbf{w}^\top x_i - y_i)^2$$

**Comments:**

- Squared Loss
- No Regularization
- Closed form solution: $\mathbf{w} = (\mathbf{X}\mathbf{X}^\top)^{-1}\mathbf{X}\mathbf{y}^\top$ , $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_n]$ , $\mathbf{y} = [y_1, ..., y_n]$
- There is an interesting connection between OLS and the first principal component of PCA (Principal Component Analysis). PCA also minimizes square loss, but looks at perpendicular loss (the horizontal distance between each point and the regression

# Famous Special Cases

## Ridge Regression

Tikhonov regularization, is the most commonly used method of regularization of ill-posed problems. In statistics, the method is known as **ridge regression**.

**goal:**

$$\min_{\mathbf{w}} \frac{1}{n} \sum_{i=1}^{n} (\mathbf{w}^\top x_i - y_i)^2 + \lambda \|w\|_2^2$$

**Comments:**

- Squared Loss
- $l_2$-Regularization
- $\mathbf{w} = (\mathbf{X}\mathbf{X}^\top + \lambda \mathbb{I})^{-1} \mathbf{X}\mathbf{y}^\top$
- Fast if data isn't too high dimensional
- Just 1 line of Julia / Python

# Famous Special Cases

## Lasso

**LASSO** is a regression analysis method that performs both variable selection and regularization in order to enhance the prediction accuracy and interpretability of the statistical model it produces.

**Goal:**

$$\min_{\mathbf{w}} \frac{1}{n} \sum_{i=1}^{n} (\mathbf{w}^\top \mathbf{x}_i - y_i)^2 + \lambda \|\mathbf{w}\|_1$$

**Comments:**

- ADVANTAGE sparsity inducing (good for feature selection);Convex
- DISADVANTAGE Not strictly convex (no unique solution); Not differentiable (at 0)
- Solve with (sub)-gradient descent or SVEN

# Famous Special Cases

## Elastic Net

**Goal**

$$\min_{\mathbf{w}} \frac{1}{n} \sum_{i=1}^{n} (\mathbf{w}^\top \mathbf{x}_i - y_i)^2 + \alpha \|\mathbf{w}\|_1 + (1-\alpha)\|\mathbf{w}\|_2^2$$

$$\alpha \in [0, 1)$$

**Comments:**

- ADVANTAGE: Strictly convex (i.e. unique solution)
- +Sparsity inducing (good for feature selection)
- +Dual of squared-loss SVM, see SVEN
- DISADVANTAGE - Non-differentiable
- Solve with (sub)-gradient descent or SVEN

# Famous Special Cases

## Logistic Regression

In regression analysis, **logistic regression** is estimating the parameters of a logistic model; it is a form of binomial regression.

**goal:**

$$\min_{\mathbf{w},b} \frac{1}{n} \sum_{i=1}^{n} \log\left(1 + e^{-y_i(\mathbf{w}^\top \mathbf{x}_i + b)}\right)$$

**Comments:**

- Often $l_1$ or $l_2$ regularized
- Solve with gradient descent.
- $\Pr(y|x) = \frac{1}{1+e^{-y(\mathbf{w}^\top x + b)}}$

# Famous Special Cases

## Linear Support Vector Machine

**goal:**

$$\min_{\mathbf{w},b} C \sum_{i=1}^{n} \max[1 - y_i(\mathbf{w}^\top \mathbf{x}_i + b), 0] + \|\mathbf{w}\|_2^2$$

**Comments:**

- Typically $l_2$ regularized (sometimes $l_1$).
- Quadratic program.
- When kernelized leads to **sparse** solutions.
- Kernelized version can be solved very efficiently with specialized algorithms (e.g. SMO)

# The End