CS722/822: Machine Learning

Instructor: Jiangwen Sun
Computer Science Department

Where we are?

- Last lecture
  - Gradient descent
  - Overfitting VS underfitting
  - Overcome overfitting
  - Reduce hypothesis space complexity
  - Increase sample size
  - Apply regularization
- Today's lecture
  - Statistical model of regularized regression
  - How to solve regularized regression

Statistical Model of Regularized Regression

- What is the statistical model for a regularized regression?
- Recall the least squares method is a maximum likelihood method (ML)
- We prove that the regularized regression is a maximum a posterior method (MAP)

Posterior Probability

Likelihood

Prior Probability

Bayes' rule $$p(Y|X) = \frac{p(X|Y)p(Y)}{p(X)}$$

Marginal Probability

where $$p(X) = \sum_Y p(X|Y)p(Y)$$

Statistical Model of Regularized Regression

- Again, assume that there is a white noise in each observation

$$y_i = f(\mathbf{x}_i; \mathbf{w}) + \varepsilon_i, \ \varepsilon_i \sim N(0,1) \quad \Longrightarrow \quad y_i \sim N(f(\mathbf{x}_i; \mathbf{w}), 1)$$

- We add another assumption on $\mathbf{w}$, so the prior of $\mathbf{w}$ is $p(\mathbf{w})$

- Now, what is the posterior of $\mathbf{w}$ given the observation of training data of pairs $(\mathbf{x}_i, y_i), \ i = 1, \cdots, N$

$$p(\mathbf{w}|\mathbf{x}, y) = \frac{p(y|\mathbf{x}, \mathbf{w})p(\mathbf{w})}{p(y|\mathbf{x})}$$

Statistical Model of Regularized Regression

$$p(\mathbf{w}|\mathbf{x}, y) \propto p(y|\mathbf{x}, \mathbf{w})p(\mathbf{w})$$

Likelihood Prior

- Likelihood

$$p(y|\mathbf{x}, \mathbf{w}) = \prod_{i=1}^{N} p(y_i|\mathbf{x}_i, \mathbf{w}) = \prod_{i=1}^{N} C \exp\left(-\frac{(y_i - f(\mathbf{x}_i; \mathbf{w}))^2}{2}\right)$$
$$= C^N \exp\left(-\frac{1}{2}\sum_{i=1}^{N}(y_i - f(\mathbf{x}_i; \mathbf{w}))^2\right)$$

- Different priors on $\mathbf{w}$ leads to different regularizers

Gaussian Prior

- Prior (Gaussian distribution, $N(0, \gamma^2 \mathbf{I})$):

$$p(\mathbf{w}) = \frac{1}{\sqrt{(2\pi\gamma^2)^{\boldsymbol{d}}}} \exp(-\frac{\|\mathbf{w}\|^2}{2\gamma^2})$$

$$p(\mathbf{w}|y, \mathbf{x}) \propto \quad C^N \tilde{C} \exp\left(-\frac{1}{2}\sum_{i=1}^{N}(y_i - f(\mathbf{x}_{\boldsymbol{i}}; \mathbf{w}))^2 - \frac{\|\mathbf{w}\|^2}{2\gamma^2}\right)$$

Maximizing the posterior is equivalent to minimizing

$$\sum_{i=1}^{N}(y_i - f(\mathbf{x}_{\boldsymbol{i}}; \mathbf{w}))^2 + \frac{\|\mathbf{w}\|^2}{\gamma^2}$$

$$\sum_{i=1}^{N}(y_i - f(\mathbf{x}_{\boldsymbol{i}}; \mathbf{w}))^2 + \lambda\|w\|^2$$ Ridge Regression

Laplace Prior

- Prior (Laplace distribution , Laplace$(0, b)$):

$$p(\mathbf{w}) = \frac{1}{(2b)^d} \exp\left(-\frac{\|\mathbf{w}\|_1}{b}\right)$$

$$p(\mathbf{w}|y, \mathbf{x}) \propto \quad C^N \tilde{C} \exp\left(-\frac{1}{2}\sum_{i=1}^{N}(y_i - f(\mathbf{x}_i; \mathbf{w}))^2 - \frac{\|\mathbf{w}\|_1}{b}\right)$$

- Maximizing the posterior is equivalent to minimizing

$$\sum_{i=1}^{N}(y_i - f(\mathbf{x}_i; \mathbf{w}))^2 + 2\frac{\|\mathbf{w}\|_1}{b}$$

$$\sum_{i=1}^{N}(y_i - f(\mathbf{x}_i; \mathbf{w}))^2 + \lambda\|\mathbf{w}\|_1$$ LASSO

Solve Ridge Regression

- Derive the analytic solution to the optimization problem for ridge regression

$$\min_{\mathbf{w}} \|\mathbf{y} - \mathbf{X}\mathbf{w}\|^2 + \lambda \|\mathbf{w}\|^2$$

$$\Rightarrow \quad \min_{\mathbf{w}} (\mathbf{y} - \mathbf{X}\mathbf{w})^T (\mathbf{y} - \mathbf{X}\mathbf{w}) + \lambda \mathbf{w}^T \mathbf{w}$$

$$\Rightarrow \quad \min_{\mathbf{w}} -2\mathbf{y}^T \mathbf{X}\mathbf{w} + \mathbf{w}^T \mathbf{X}^T \mathbf{X}\mathbf{w} + \lambda \mathbf{w}^T \mathbf{w}$$

$$\Rightarrow \quad \min_{\mathbf{w}} -2\mathbf{y}^T \mathbf{X}\mathbf{w} + \mathbf{w}^T (\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I})\mathbf{w}$$

- Gradient: $-2\mathbf{X}^T \mathbf{y} + \mathbf{2}(\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I})\mathbf{w}$

- Set the gradient to 0

$$(\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I})\mathbf{w} = \mathbf{X}^T \mathbf{y}$$

$$\mathbf{w} = (\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^T \mathbf{y}$$

Ridge Regression

- Numerical stability of ridge regression over unregularized linear regression
  - Unregularized linear regression

$$(\mathbf{X}^T\mathbf{X})\mathbf{w} = \mathbf{X}^T\mathbf{y},\ \mathrm{X}^{\mathrm{T}}\mathrm{X} \textit{ might not be invertible}$$

$$\mathbf{w} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y},\ \textit{need pseudoinverse}$$

$\mathbf{w}$ is not unique when $\mathbf{X}^T\mathbf{X}$ is not invertible

  - Ridge Regression

$$(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})\mathbf{w} = \mathbf{X}^T\mathbf{y},\ (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}) \textit{ often invertible}$$

$$\mathbf{w} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y}$$

Unique solution

Solve Ridge Regression

- Gradient Descent
  - Algorithm

1. Set iteration $k = 0$, make an initial guess $\mathbf{w}_0$
2. repeat:
3. Compute the negative gradient of $E(\mathbf{w})$ at $\mathbf{w}_k$ and set it to be the search direction $\mathbf{d}_k$
4. Choose a step size α_k to sufficiently reduce $E(\mathbf{w}_k + \alpha_k \mathbf{d}_k)$
5. Update $\mathbf{w}_{k+1} = \mathbf{w}_k + \alpha_k \mathbf{d}_k$
6. $k = k + 1$
7. Until a determination rule is met

Exactly the same as the gradient descent for solving least squares

Solve Ridge Regression

- The only difference is how to compute the negative gradient

$$-\left.\frac{\partial E(\mathbf{w})}{\partial \mathbf{w}}\right|_{\mathbf{w}=\mathbf{w}_k} = 2\mathbf{X}^T(\mathbf{y} - \mathbf{X}\mathbf{w}_k)$$

Least Squares

$$-\left.\frac{\partial E(\mathbf{w})}{\partial \mathbf{w}}\right|_{\mathbf{w}=\mathbf{w}_k} = 2\mathbf{X}^T\mathbf{y} - 2(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})\mathbf{w}_k$$

Ridge Regression

Solve LASSO

$$E(\mathbf{w}) = \sum_{i=1}^{N} (y_i - f(x_i|\mathbf{w}))^2 + \lambda\|\mathbf{w}\|_1$$

Convex differentiable

Convex non-differentiable

Need Generalized Gradient Descent, instead of the gradient descent for ridge regression

Let us review **subgradient**, and **proximal operator**

Subgradient

Copy from CMU Optimization

Remember that for convex $f : \mathbb{R}^n \to \mathbb{R}$,

$$f(y) \geq f(x) + \nabla f(x)^T (y - x) \quad \text{all } x, y$$

I.e., linear approximation always underestimates f

A **subgradient** of convex $f : \mathbb{R}^n \to \mathbb{R}$ at x is any $g \in \mathbb{R}^n$ such that

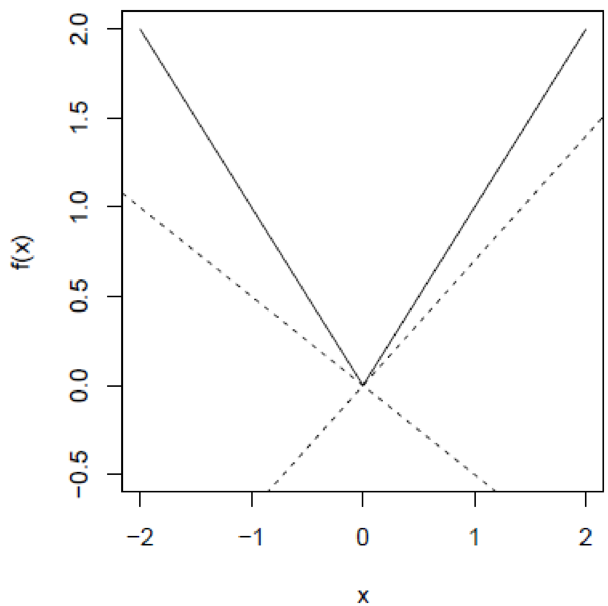
$$f(y) \geq f(x) + g^T (y - x), \quad \text{all } y$$

- Always exists
- If f differentiable at x, then $g = \nabla f(x)$ uniquely
- Actually, same definition works for nonconvex f (however, subgradient need not exist)

Subgradient

Copy from CMU Optimization

Consider $f : \mathbb{R} \to \mathbb{R}$, $f(x) = |x|$

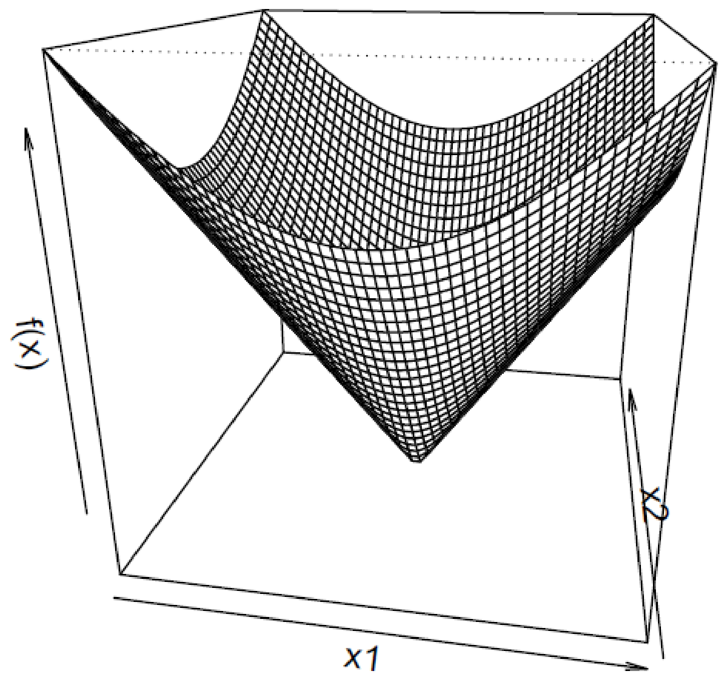


- For $x \neq 0$, unique subgradient $g = \mathrm{sign}(x)$
- For $x = 0$, subgradient g is any element of $[-1, 1]$

Subgradient

Copy from CMU Optimization

Consider $f : \mathbb{R}^n \to \mathbb{R}$, $f(x) = \|x\|$ (Euclidean norm)

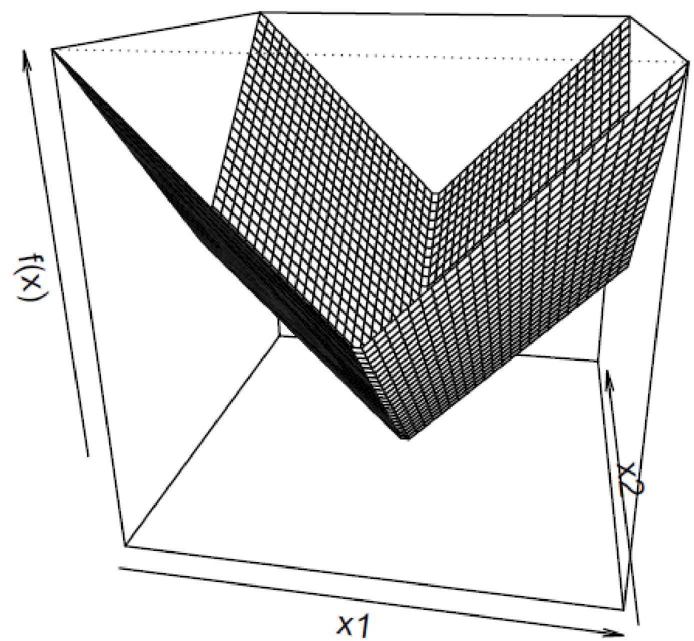


- For $x \neq 0$, unique subgradient $g = x/\|x\|$
- For $x = 0$, subgradient g is any element of $\{z : \|z\| \leq 1\}$

Subgradient

Copy from CMU Optimization

Consider $f : \mathbb{R}^n \to \mathbb{R}$, $f(x) = \|x\|_1$



- For $x_i \neq 0$, unique ith component $g_i = \mathrm{sign}(x_i)$
- For $x_i = 0$, ith component g_i is an element of $[-1, 1]$

Subgradient

Copy from CMU Optimization

Set of all subgradients of convex f is called the **subdifferential:**

$$\partial f(x) = \{g \in \mathbb{R}^n : g \text{ is a subgradient of } f \text{ at } x\}$$

- $\partial f(x)$ is closed and convex (even for nonconvex f)
- Nonempty (can be empty for nonconvex f)
- If f is differentiable at x, then $\partial f(x) = \{\nabla f(x)\}$
- If $\partial f(x) = \{g\}$, then f is differentiable at x and $\nabla f(x) = g$

Subgradient

Copy from CMU Optimization

Subgradients are important for two reasons:

- Convex analysis: optimality characterization via subgradients, monotonicity
- Convex optimization: if you can compute subgradients, then you can minimize (almost) any convex function

Subgradient

Copy from CMU Optimization

For convex f,

$$f(x^\star) = \min_{x \in \mathbb{R}^n} f(x) \quad \Leftrightarrow \quad 0 \in \partial f(x^\star)$$

I.e., $x^\star$ is a minimizer if and only if 0 is a subgradient of f at $x^\star$

Why? Easy: $g = 0$ being a subgradient means that for all y

$$f(y) \geq f(x^\star) + 0^T (y - x^\star) = f(x^\star)$$

Note analogy to differentiable case, where $\partial f(x) = \{\nabla f(x)\}$

Subgradient

Copy from CMU Optimization

Lasso problem can be parametrized as

$$\min_x \frac{1}{2}\|y - Ax\|^2 + \lambda\|x\|_1$$

where $\lambda \geq 0$. Consider simplified problem with $A = I$:

$$\min_x \frac{1}{2}\|y - x\|^2 + \lambda\|x\|_1$$

Claim: solution of simple problem is $x^\star = S_\lambda(y)$, where S_λ is the **soft-thresholding operator:**

$$[S_\lambda(y)]_i = \begin{cases} y_i - \lambda & \text{if } y_i > \lambda \\ 0 & \text{if } -\lambda \leq y_i \leq \lambda \\ y_i + \lambda & \text{if } y_i < -\lambda \end{cases}$$

Subgradient

Copy from CMU Optimization

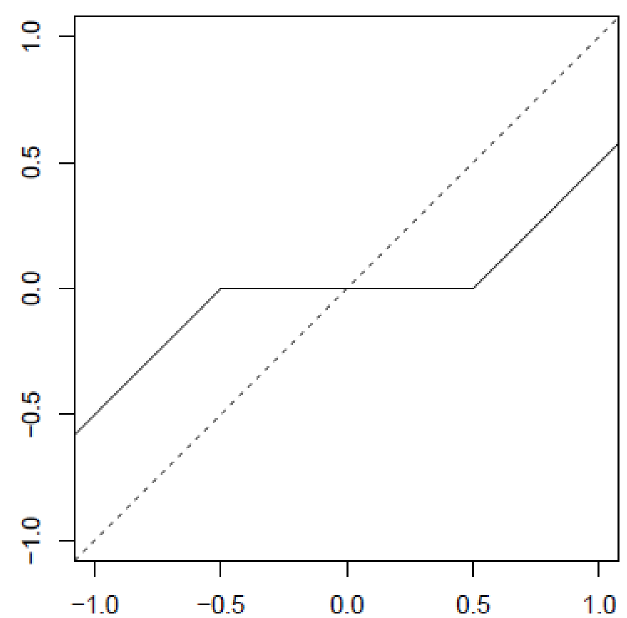
Why? Subgradients of $f(x) = \frac{1}{2}\|y - x\|^2 + \lambda\|x\|_1$ are

$$g = x - y + \lambda s,$$

where $s_i = \mathrm{sign}(x_i)$ if $x_i \neq 0$ and $s_i \in [-1, 1]$ if $x_i = 0$

Now just plug in $x = S_\lambda(y)$ and check we can get $g = 0$

Soft-thresholding in one variable:



Generalized Gradient Descent

Copy from CMU Optimization

Suppose

$$f(x) = g(x) + h(x)$$

- g is convex, differentiable
- h is convex, not necessarily differentiable

If f were differentiable, gradient descent update:

$$x^+ = x - t\nabla f(x)$$

Recall motivation: minimize quadratic approximation to f around x, replace $\nabla^2 f(x)$ by $\frac{1}{t}I$,

$$x^+ = \underset{z}{\operatorname{argmin}} \; \underbrace{f(x) + \nabla f(x)^T(z - x) + \frac{1}{2t}\|z - x\|^2}_{\widehat{f}_t(z)}$$

Proximal operator

Generalized Gradient Descent

Copy from CMU Optimization

In our case f is not differentiable, but $f = g + h$, g differentiable

Why don't we make quadratic approximation to g, leave h alone?

I.e., update

$$\begin{aligned} x^+ &= \operatorname*{argmin}_z \widehat{g}_t(z) + h(z) \\ &= \operatorname*{argmin}_z g(x) + \nabla g(x)^T (z - x) + \frac{1}{2t}\|z - x\|^2 + h(z) \\ &= \operatorname*{argmin}_z \frac{1}{2t}\big\|z - (x - t\nabla g(x))\big\|^2 + h(z) \end{aligned}$$

Proximal operator

$\frac{1}{2t}\|z - (x - t\nabla g(x))\|^2$ be close to gradient update for g

$h(z)$ also make h small

Generalized Gradient Descent

Copy from CMU Optimization

$$\mathrm{prox}_t(x) = \arg\min_{z \in R^d} \frac{1}{2t}\|z - x\|^2 + h(z)$$

- If h(z) = the 1-norm penalty, how to solve the above problem

$$\mathrm{prox}_t(x) = \arg\min_{z \in R^d} \frac{1}{2t}\|z - x\|^2 + \lambda\|z\|_1$$

  - Soft-thresholding

$$z_j = \begin{cases} x_j - \lambda t, & \text{if } x_j > \lambda t \\ 0, & \text{if } -\lambda t \le x_j \le \lambda t \\ x_j + \lambda t, & \text{if } x_j < -\lambda t \end{cases}$$

Generalized Gradient Descent

Copy from CMU Optimization

$$\mathrm{prox}_t(x) = \arg\min_{z \in R^d} \frac{1}{2t} \|z - x\|^2 + \lambda \|z\|_1$$

- Generalized gradient descent
  - Initialize x0,
  - Repeat

$$x_{k+1} = \mathrm{prox}_t\left(x_k - t_k \nabla g\left(x_k\right)\right)$$

  - Until a termination rule is met

Generalized Gradient Descent

Copy from CMU Optimization

- How to solve

$$x_{k+1} = \mathrm{prox}_t\left(x_k - t_k \nabla g\left(x_k\right)\right)$$

- By definition, it solves the following problem

$$\arg\min_{z \in R^d} \frac{1}{2t}\left\|z - \left(x_k - t_k \nabla g\left(x_k\right)\right)\right\|^2 + \lambda \|z\|_1$$

$$z_j = \begin{cases} \left(x_k - t_k \nabla g\left(x_k\right)\right)_j - \lambda t, & \text{if } \left(x_k - t_k \nabla g\left(x_k\right)\right)_j > \lambda t \\ 0, & \text{if } -\lambda t \le \left(x_k - t_k \nabla g\left(x_k\right)\right)_j \le \lambda t \\ \left(x_k - t_k \nabla g\left(x_k\right)\right)_j + \lambda t, & \text{if } \left(x_k - t_k \nabla g\left(x_k\right)\right)_j < -\lambda t \end{cases}$$