Introduction to Machine Learning - CS725
Instructor: Prof. Ganesh Ramakrishnan
Lecture 09 - Optimization Foundations Applied to Regression Formulations

Building on questions on Least Squares Linear Regression

1. Is there a probabilistic interpretation?
   - Gaussian Error, Maximum Likelihood Estimate
2. Addressing overfitting
   - Bayesian and Maximum Aposteriori Estimates, Regularization, Support Vector Regression
3. How to minimize the resultant and more complex error functions?
   - Level Curves and Surfaces, Gradient Vector, Directional Derivative, Gradient Descent Algorithm, Convexity, Necessary and Sufficient Conditions for Optimality

(Optional) Subgradients

- An equivalent condition for convexity of $f(\mathbf{x})$:

$$\forall\ \mathbf{x}, \mathbf{y} \in \mathbf{dmn(f)},\ \ \mathbf{f(y)} \geq \mathbf{f(x)} + \nabla^\top \mathbf{f(x)(y - x)}$$

- $\mathbf{g_f}(\mathbf{x})$ is a *subgradient* for a function f at $\mathbf{x}$ if

$$\forall\ \mathbf{y} \in \mathbf{dmn(f)},\ \ \mathbf{f(y)} \geq \mathbf{f(x)} + \mathbf{g_f(x)}^\top\mathbf{(y - x)}$$

- Any convex (even non-differentiable) function will have a subgradient at any point in the domain!
- If a convex function f is differentiable at $\mathbf{x}$ then $\nabla f(\mathbf{x}) = \mathbf{g_f}(\mathbf{x})$
- $\mathbf{x}$ is a point of minimum of (convex) f if and only if $\mathbf{0}$ is a subgradient of f at $\mathbf{x}$

(Sub)Gradient Descent Algorithm

Find starting point $\mathbf{w^{(0)}} \epsilon \mathcal{D}$

- $\Delta\mathbf{w^k} = -\nabla\varepsilon(\mathbf{w^{(k)}})$
- Choose a step size $t^{(k)} > 0$ using exact or backtracking ray search.
- Obtain $\mathbf{w^{(k+1)}} = \mathbf{w^{(k)}} + \mathbf{t^{(k)}}\mathbf{\Delta w^{(k)}}$.
- Set $k = k + 1$. **until** stopping criterion
  (such as $\|\nabla\varepsilon(\mathbf{w}^{(k+1)})\| \leq \epsilon$) is satisfied

(Sub)Gradient Descent Algorithm

Exact line search algorithm to find $t^{(k)}$

- The line search approach first finds a descent direction along which the objective function f will be reduced and then computes a step size that determines how far **x** should move along that direction.
- In general,

$$t^{(k)} = \arg\min_{t} f\left(\mathbf{w}^{(\mathbf{k+1})}\right) \tag{1}$$

- Thus,

Exact line search algorithm to find $t^{(k)}$

- The line search approach first finds a descent direction along which the objective function f will be reduced and then computes a step size that determines how far **x** should move along that direction.
- In general,

$$t^{(k)} = \arg\min_t f\left(\mathbf{w^{(k+1)}}\right) \tag{1}$$

- Thus, for L_2 regularized least squared regression

$$t^{(k)} = \arg\min_t \epsilon\left(\mathbf{w^{(k)}} + \mathbf{2t}\left(\mathbf{\Phi^T y} - \mathbf{\Phi^T}\phi\mathbf{w^{(k)}} - \lambda\mathbf{w^{(k)}}\right)\right) \tag{2}$$

Illustration of (Sub)Gradient Descent Algorithm

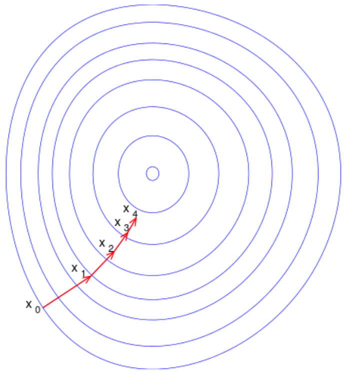


Figure 1: A red arrow originating at a point shows the direction of the negative gradient at that point. Note that the (negative) gradient at a point is orthogonal to the level curve going through that point. We see that gradient descent leads us to the bottom of the bowl, that is, to the point where the value of the function F is minimal. Source: Wikipidea

Gradient Descent and LS Regression (Tutorial 3+4)

Consider solving the (L_2 regularized) Least Squares Linear Regression problem using the gradient descent algorithm. And let us say $w^{(0)} = 0$ and that the step length $t^{(k)}$ is computed using exact line search for each value of k. In how many steps will the gradient descent algorithm converge? What would be your answer if we had a different initialization for $w^{(0)}$

Subgradients and Lasso

$$\mathbf{w}_{Lasso} = \arg\min_{\mathbf{w}} \; ||\Phi\mathbf{w} - \mathbf{y}||_2^2 + ||\mathbf{w}||_1$$

- The unconstrained form for Lasso has no closed form solution
- But it can be solved using a generalization of gradient descent called *proximal subgradient descent*[1]

[1] https://www.cse.iitb.ac.in/~cs725/notes/classNotes/lassoElaboration.pdf

Iterative Soft Thresholding Algorithm for Solving Lasso

Proximal Subgradient Descent for Lasso

- Let $\varepsilon(\mathbf{w}) = \|\phi\mathbf{w} - \mathbf{y}\|_2^2$
- **Proximal Subgradient Descent Algorithm:**
  Initialization: Find starting point $\mathbf{w}^{(0)}$
  - Let $\widehat{\mathbf{w}}^{(\mathbf{k+1})}$ be a next gradient descent iterate for $\varepsilon(\mathbf{w}^k)$
  - Compute $\mathbf{w}^{(k+1)} = \underset{\mathbf{w}}{\mathbf{argmin}}||\mathbf{w} - \widehat{\mathbf{w}}^{(\mathbf{k+1})}||_{\mathbf{2}}^{\mathbf{2}} + \lambda\mathbf{t}||\mathbf{w}||_{\mathbf{1}}$ by setting subgradient of this objective to $\mathbf{0}$. This results in (see https://www.cse.iitb.ac.in/~cs725/notes/classNotes/lassoElaboration.pdf)
    1. ...
    2. ...
    3. ...
  - Set $k = k + 1$, **until** stopping criterion is satisfied (such as no significant changes in $\mathbf{w}^k$ w.r.t $\mathbf{w}^{(k-1)}$)

Iterative Soft Thresholding Algorithm (Proximal Subgradient Descent) for Lasso

- Let $\varepsilon(\mathbf{w}) = ||\phi\mathbf{w} - \mathbf{y}||_2^2$
- **Iterative Soft Thresholding Algorithm:**
  Initialization: Find starting point $\mathbf{w}^{(0)}$
  - Let $\widehat{\mathbf{w}}^{(\mathbf{k+1})}$ be a next iterate for $\varepsilon(\mathbf{w}^k)$ computed using using any (gradient) descent algorithm
  - Compute $\mathbf{w}^{(k+1)} = \underset{\mathbf{w}}{argmin}||\mathbf{w} - \widehat{\mathbf{w}}^{(\mathbf{k+1})}||_{\mathbf{2}}^{\mathbf{2}} + \lambda\mathbf{t}||\mathbf{w}||_{\mathbf{1}}$ by:
    1. If $\widehat{w}_i^{(k+1)} > \lambda t/2$, then $w_i^{(k+1)} = -\lambda t/2 + \widehat{w}_i^{(k+1)}$
    2. If $\widehat{w}_i^{(k+1)} < -\lambda t/2$, then $w_i^{(k+1)} = \lambda t/2 + \widehat{w}_i^{(k+1)}$
    3. 0 otherwise.
  - Set $k = k + 1$, **until** stopping criterion is satisfied (such as no significant changes in $\mathbf{w}^k$ w.r.t $\mathbf{w}^{(k-1)}$)

Constrained Least Squares Linear Regression

Find

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \|\phi\mathbf{w} - \mathbf{y}\|^2 \;\; s.t. \;\; \|\mathbf{w}\|_p \leq \zeta, \tag{3}$$

where

$$\|\mathbf{w}\|_p = \left(\sum_{i=1}^{n} |w_i|^p\right)^{\frac{1}{p}} \tag{4}$$

Claim: This is an equivalent reformulation of the penalized least squares. Why?

p-Norm level curves

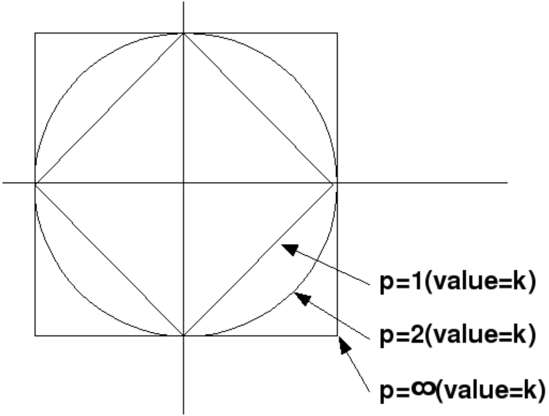


Figure 2: p-Norm curves for constant norm value and different p

Convex Optimization Problem

- Formally, a convex optimization problem is an optimization problem of the form

$$\begin{aligned} minimize\ f(\mathbf{w}) \qquad (5) \\ subject\ to\ c\ \in\ C \qquad (6) \end{aligned}$$

where f is a convex function, C is a convex set, and $\mathbf{x}$ is the optimization variable.

- A specific form of the above would be

$$\begin{aligned} minimize\ f(\mathbf{w}) & & \qquad (7) \\ subject\ to\ g_i(\mathbf{w}) & \leq 0,\ \ i = 1, ..., m \qquad (8) \\ h_i(\mathbf{w}) & = 0, i = 1, ..., p \qquad (9) \end{aligned}$$

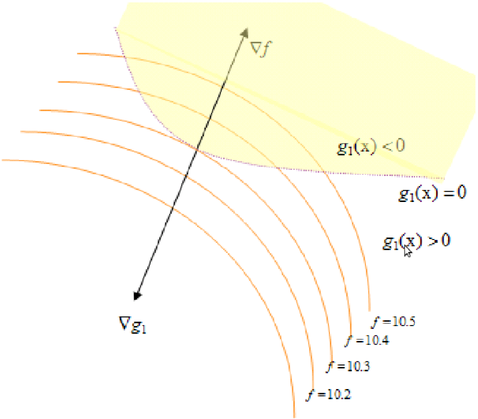
where f is a convex function, g_i are convex functions, and h_i are affine (linear) functions, and $\mathbf{x}$ is the vector of optimization variables.

Constrained convex problems

Q. *How to solve such constrained problems?*

A. Canonical example:

$$\textit{Minimize } f(\mathbf{w}) \text{ } s.t. \text{ } g_1(\mathbf{w}) \leq 0 \tag{10}$$



Constrained Convex Problems

- If $\mathbf{w}^*$ is on the boundary of g_1, *i.e.*, if $g_1(\mathbf{w}^*) = 0$,

$$\nabla f(\mathbf{w}^*) = -\lambda \nabla g_1(\mathbf{w}^*) \textit{ for some } \lambda \geq 0$$

- **Intuition:**

[2] $\nabla_{\perp} g_1(\mathbf{w}^*)$ is the direction orthogonal to $\nabla g_1(\mathbf{w}^*)$

[3] Section 4.4, pg-72: cs725/notes/BasicsOfConvexOptimization.pdf

Constrained Convex Problems

- If $\mathbf{w}^*$ is on the boundary of g_1, *i.e.*, if $g_1(\mathbf{w}^*) = 0$,

$$\nabla f(\mathbf{w}^*) = -\lambda \nabla g_1(\mathbf{w}^*) \textit{ for some } \lambda \geq 0$$

- **Intuition:** If the above didn't hold, then we would have $\nabla f(\mathbf{w}^*) = \lambda_1 \nabla g_1(\mathbf{w}^*) + \lambda_2 \nabla_\perp g_1(\mathbf{w}^*)$, where, by moving in direction[2] $\pm\nabla_\perp g_1(\mathbf{w}^*)$ (or $-\nabla g_1(\mathbf{w}^*)$), we remain on boundary $g_1(\mathbf{w}^*) = 0$, (or within $g_1(\mathbf{w}^*) \leq 0$) while decreasing the value of f, which is not possible at the point of optimality.
- Thus, at the point of optimality[3],

[2] $\nabla_\perp g_1(\mathbf{w}^*)$ is the direction orthogonal to $\nabla g_1(\mathbf{w}^*)$

[3] `Section 4.4, pg-72: cs725/notes/BasicsOfConvexOptimization.pdf`

Constrained Convex Problems

- If $\mathbf{w}^*$ is on the boundary of g_1, *i.e.*, if $g_1(\mathbf{w}^*) = 0$,

$$\nabla f(\mathbf{w}^*) = -\lambda \nabla g_1(\mathbf{w}^*) \textit{ for some } \lambda \geq 0$$

- **Intuition:** If the above didn't hold, then we would have $\nabla f(\mathbf{w}^*) = \lambda_1 \nabla g_1(\mathbf{w}^*) + \lambda_2 \nabla_\perp g_1(\mathbf{w}^*)$, where, by moving in direction[2] $\pm \nabla_\perp g_1(\mathbf{w}^*)$ (or $-\nabla g_1(\mathbf{w}^*)$), we remain on boundary $g_1(\mathbf{w}^*) = 0$, (or within $g_1(\mathbf{w}^*) \leq 0$) while decreasing the value of f, which is not possible at the point of optimality.
- Thus, at the point of optimality[3], for some $\lambda \geq 0$,

$$\textit{Either } g_1(\mathbf{w}^*) < 0 \quad \& \quad \nabla f(\mathbf{w}^*) = 0 \tag{11}$$
$$\textit{Or } g_1(\mathbf{w}^*) = 0 \quad \& \quad \nabla f(\mathbf{w}^*) = -\lambda \nabla g_1(\mathbf{w}^*) \tag{12}$$

[2] $\nabla_\perp g_1(\mathbf{w}^*)$ is the direction orthogonal to $\nabla g_1(\mathbf{w}^*)$

[3] `Section 4.4, pg-72: cs725/notes/BasicsOfConvexOptimization.pdf`

Explaining the Figure

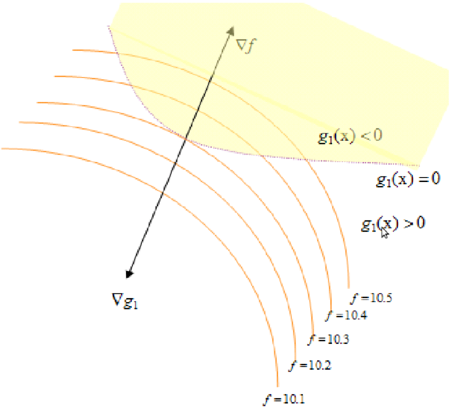


Figure 4: Two conditions under which a minimum can occur: a) When the minimum is on the constraint function boundary, in which case the gradients are in opposite directions; b) When point of minimum is inside the constraint space (shown in yellow shade), in which case $\nabla f(\mathbf{w}^*) = \mathbf{0}$.

More Explanation and Lagrange Function

- The first condition occurs when minima lies on the boundary of function g. In this case, gradient vectors corresponding to the functions f and g, at $\mathbf{w}^*$, point in opposite directions barring multiplication by a real constant.
- Second condition represents the case that point of minimum lies inside the constraint space. This space is shown shaded in Figure 1. Clearly, for this case, $\nabla f(\mathbf{w}) = \mathbf{0}$.
- An Alternative Representation: $\nabla L(\mathbf{w}, \lambda) = 0$ for some $\lambda \geq 0$ where

$$L(\mathbf{w}, \lambda) = f(\mathbf{w}) + \lambda \mathbf{g}(\mathbf{w}); \lambda \in \mathbb{R}$$

is called the lagrange function which has objective function augmented by weighted sum of constraint functions

Duality and KKT conditions

For a convex objective and constraint function, the minima, $\mathbf{w}^*$, can satisfy one of the following two conditions:

1. $g(\mathbf{w}^*) = \mathbf{0}$ and $\nabla f(\mathbf{w}^*) = -\lambda \nabla \mathbf{g}(\mathbf{w}^*)$
2. $g(\mathbf{w}^*) < \mathbf{0}$ and $\nabla f(\mathbf{w}^*) = \mathbf{0}$

Duality and KKT conditions

- Here, we wish to penalize higher magnitude coefficients, hence, we wish $g(\mathbf{w})$ to be negative while minimizing the lagrangian. In order to maintain such direction, we must have $\lambda \geq 0$. Also, for solution $\mathbf{w}$ to be feasible, $\nabla g(\mathbf{w}) \leq \mathbf{0}$.
- Due to complementary slackness condition, we further have $\lambda g(\mathbf{w}) = \mathbf{0}$, which roughly suggests that the lagrange multiplier is zero unless constraint is active at the minimum point. As $\mathbf{w}$ minimizes the lagrangian $L(\mathbf{w}, \lambda)$, gradient must vanish at this point and hence we have $\nabla f(\mathbf{w}) + \lambda \nabla \mathbf{g}(\mathbf{w}) = \mathbf{0}$