Applied Probability and Statistics

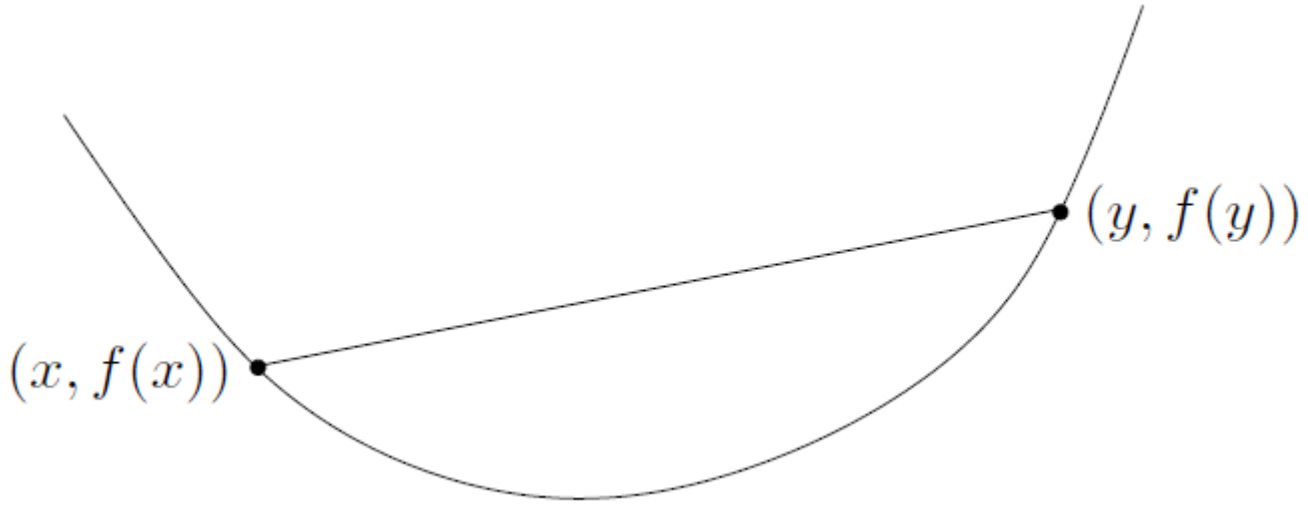
Module-1, Lecture-11

By: Dr. A K Yadav (9911375598)
Dept of CSE, ASET, AUUP

Convex Function

A function $f : \mathbf{R}^n \to \mathbf{R}$ is *convex* if $\mathbf{dom}\, f$ is a convex set and if for all x, $y \in \mathbf{dom}\, f$, and θ with $0 \leq \theta \leq 1$, we have

$$f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta)f(y). \tag{3.1}$$



Extended-value extensions

- A function is convex if and only if it is convex when restricted to any line that intersects its domain.

It is often convenient to extend a convex function to all of $\mathbf{R}^n$ by defining its value to be ∞ outside its domain. If f is convex we define its *extended-value extension* $\tilde{f} : \mathbf{R}^n \to \mathbf{R} \cup \{\infty\}$ by

$$\tilde{f}(x) = \begin{cases} f(x) & x \in \mathbf{dom}\, f \\ \infty & x \notin \mathbf{dom}\, f. \end{cases}$$

Example 3.1 *Indicator function of a convex set.* Let $C \subseteq \mathbf{R}^n$ be a convex set, and consider the (convex) function I_C with domain C and $I_C(x) = 0$ for all $x \in C$. In other words, the function is identically zero on the set C. Its extended-value extension is given by

$$\tilde{I}_C(x) = \begin{cases} 0 & x \in C \\ \infty & x \notin C. \end{cases}$$

The convex function $\tilde{I}_C$ is called the *indicator function* of the set C.

First-order conditions

Suppose f is differentiable (*i.e.*, its gradient ∇f exists at each point in **dom** f, which is open). Then f is convex if and only if **dom** f is convex and

$$f(y) \geq f(x) + \nabla f(x)^T (y - x) \tag{3.2}$$

holds for all $x,\ y \in \mathbf{dom}\, f$. This inequality is illustrated in figure 3.2.

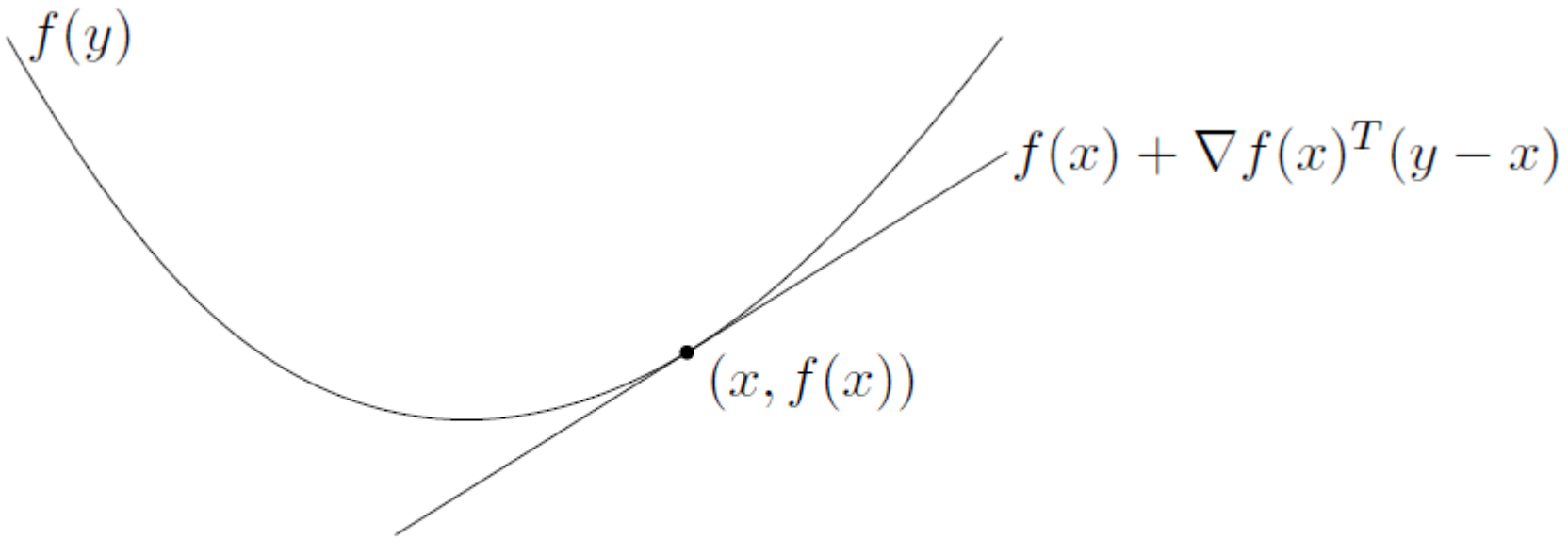


Figure 3.2 If f is convex and differentiable, then $f(x)+\nabla f(x)^T(y-x) \leq f(y)$ for all $x,\ y \in \mathbf{dom}\ f$.

Strict convexity can also be characterized by a first-order condition: f is strictly convex if and only if $\mathbf{dom}\, f$ is convex and for $x,\ y \in \mathbf{dom}\, f$, $x \neq y$, we have

$$f(y) > f(x) + \nabla f(x)^T (y - x). \tag{3.3}$$

Second-order conditions

We now assume that f is twice differentiable, that is, its *Hessian* or second derivative $\nabla^2 f$ exists at each point in $\mathbf{dom}\, f$, which is open. Then f is convex if and only if $\mathbf{dom}\, f$ is convex and its Hessian is positive semidefinite: for all $x \in \mathbf{dom}\, f$,

$$\nabla^2 f(x) \succeq 0.$$

For a function on $\mathbf{R}$, this reduces to the simple condition $f''(x) \geq 0$ (and $\mathbf{dom}\, f$ convex, *i.e.*, an interval), which means that the derivative is nondecreasing. The condition $\nabla^2 f(x) \succeq 0$ can be interpreted geometrically as the requirement that the graph of the function have positive (upward) curvature at x. We leave the proof

Applied Probability and Statistics

Module-1, Lecture-12

By: Dr. A K Yadav (9911375598)
Dept of CSE, ASET, AUUP

Duality, KKT conditions

- The Lagrangian

We consider an optimization problem in the standard form (4.1):

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leq 0, \quad i = 1, \ldots, m \\ & h_i(x) = 0, \quad i = 1, \ldots, p, \end{array} \tag{5.1}$$

with variable $x \in \mathbf{R}^n$. We assume its domain $\mathcal{D} = \bigcap_{i=0}^{m} \mathbf{dom}\, f_i \cap \bigcap_{i=1}^{p} \mathbf{dom}\, h_i$ is nonempty, and denote the optimal value of (5.1) by $p^\star$. We do not assume the problem (5.1) is convex.

The basic idea in Lagrangian duality is to take the constraints in (5.1) into account by augmenting the objective function with a weighted sum of the constraint functions. We define the *Lagrangian* $L : \mathbf{R}^n \times \mathbf{R}^m \times \mathbf{R}^p \to \mathbf{R}$ associated with the problem (5.1) as

$$L(x, \lambda, \nu) = f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{p} \nu_i h_i(x),$$

with $\mathbf{dom}\, L = \mathcal{D} \times \mathbf{R}^m \times \mathbf{R}^p$. We refer to λ_i as the *Lagrange multiplier* associated with the ith inequality constraint $f_i(x) \leq 0$; similarly we refer to ν_i as the Lagrange multiplier associated with the ith equality constraint $h_i(x) = 0$. The vectors λ and ν are called the *dual variables* or *Lagrange multiplier vectors* associated with the problem (5.1).

- The Lagrange dual function

We define the *Lagrange dual function* (or just *dual function*) $g : \mathbf{R}^m \times \mathbf{R}^p \to \mathbf{R}$ as the minimum value of the Lagrangian over x: for $\lambda \in \mathbf{R}^m$, $\nu \in \mathbf{R}^p$,

$$g(\lambda, \nu) = \inf_{x \in \mathcal{D}} L(x, \lambda, \nu) = \inf_{x \in \mathcal{D}} \left(f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{p} \nu_i h_i(x) \right).$$

When the Lagrangian is unbounded below in x, the dual function takes on the value $-\infty$. Since the dual function is the pointwise infimum of a family of affine functions of (λ, ν), it is concave, even when the problem (5.1) is not convex.

- Lower bounds on optimal value

The dual function yields lower bounds on the optimal value $p^\star$ of the problem (5.1): For any $\lambda \succeq 0$ and any ν we have

$$g(\lambda, \nu) \leq p^\star. \tag{5.2}$$

This important property is easily verified. Suppose $\tilde{x}$ is a feasible point for the problem (5.1), *i.e.*, $f_i(\tilde{x}) \leq 0$ and $h_i(\tilde{x}) = 0$, and $\lambda \succeq 0$. Then we have

$$\sum_{i=1}^{m} \lambda_i f_i(\tilde{x}) + \sum_{i=1}^{p} \nu_i h_i(\tilde{x}) \leq 0,$$

since each term in the first sum is nonpositive, and each term in the second sum is zero, and therefore

$$L(\tilde{x}, \lambda, \nu) = f_0(\tilde{x}) + \sum_{i=1}^{m} \lambda_i f_i(\tilde{x}) + \sum_{i=1}^{p} \nu_i h_i(\tilde{x}) \leq f_0(\tilde{x}).$$

Hence

$$g(\lambda, \nu) = \inf_{x \in \mathcal{D}} L(x, \lambda, \nu) \leq L(\tilde{x}, \lambda, \nu) \leq f_0(\tilde{x}).$$

Since $g(\lambda, \nu) \leq f_0(\tilde{x})$ holds for every feasible point $\tilde{x}$, the inequality (5.2) follows. The lower bound (5.2) is illustrated in figure 5.1, for a simple problem with $x \in \mathbf{R}$ and one inequality constraint.

The inequality (5.2) holds, but is vacuous, when $g(\lambda, \nu) = -\infty$. The dual function gives a nontrivial lower bound on $p^\star$ only when $\lambda \succeq 0$ and $(\lambda, \nu) \in \mathbf{dom}\, g$, *i.e.*, $g(\lambda, \nu) > -\infty$. We refer to a pair (λ, ν) with $\lambda \succeq 0$ and $(\lambda, \nu) \in \mathbf{dom}\, g$ as *dual feasible*, for reasons that will become clear later.

- Linear approximation interpretation

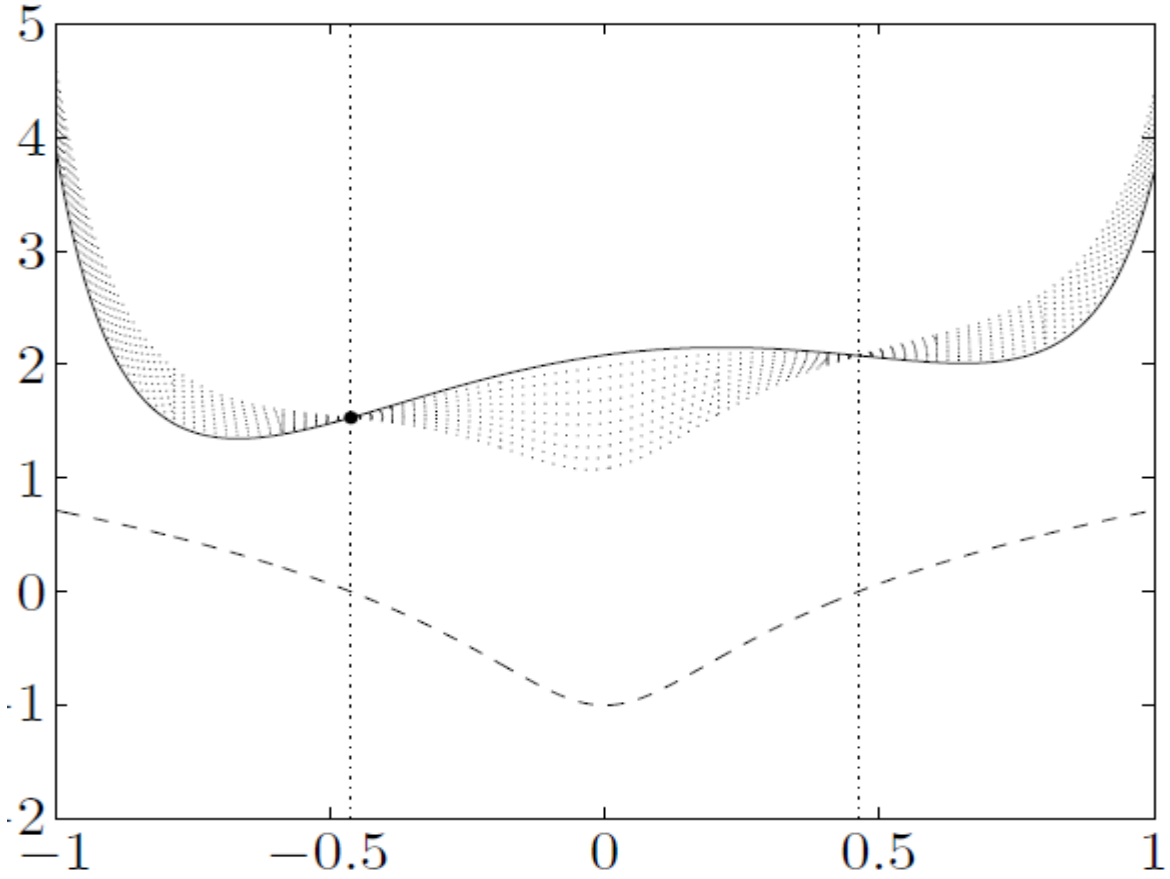


Figure 5.1 *Lower bound from a dual feasible point.* The solid curve shows the objective function f_0, and the dashed curve shows the constraint function f_1. The feasible set is the interval $[-0.46, 0.46]$, which is indicated by the two dotted vertical lines. The optimal point and value are $x^\star = -0.46$, $p^\star = 1.54$ (shown as a circle). The dotted curves show $L(x, \lambda)$ for $\lambda = 0.1,\ 0.2, \ldots, 1.0$. Each of these has a minimum value smaller than $p^\star$, since on the feasible set (and for $\lambda \geq 0$) we have $L(x, \lambda) \leq f_0(x)$.

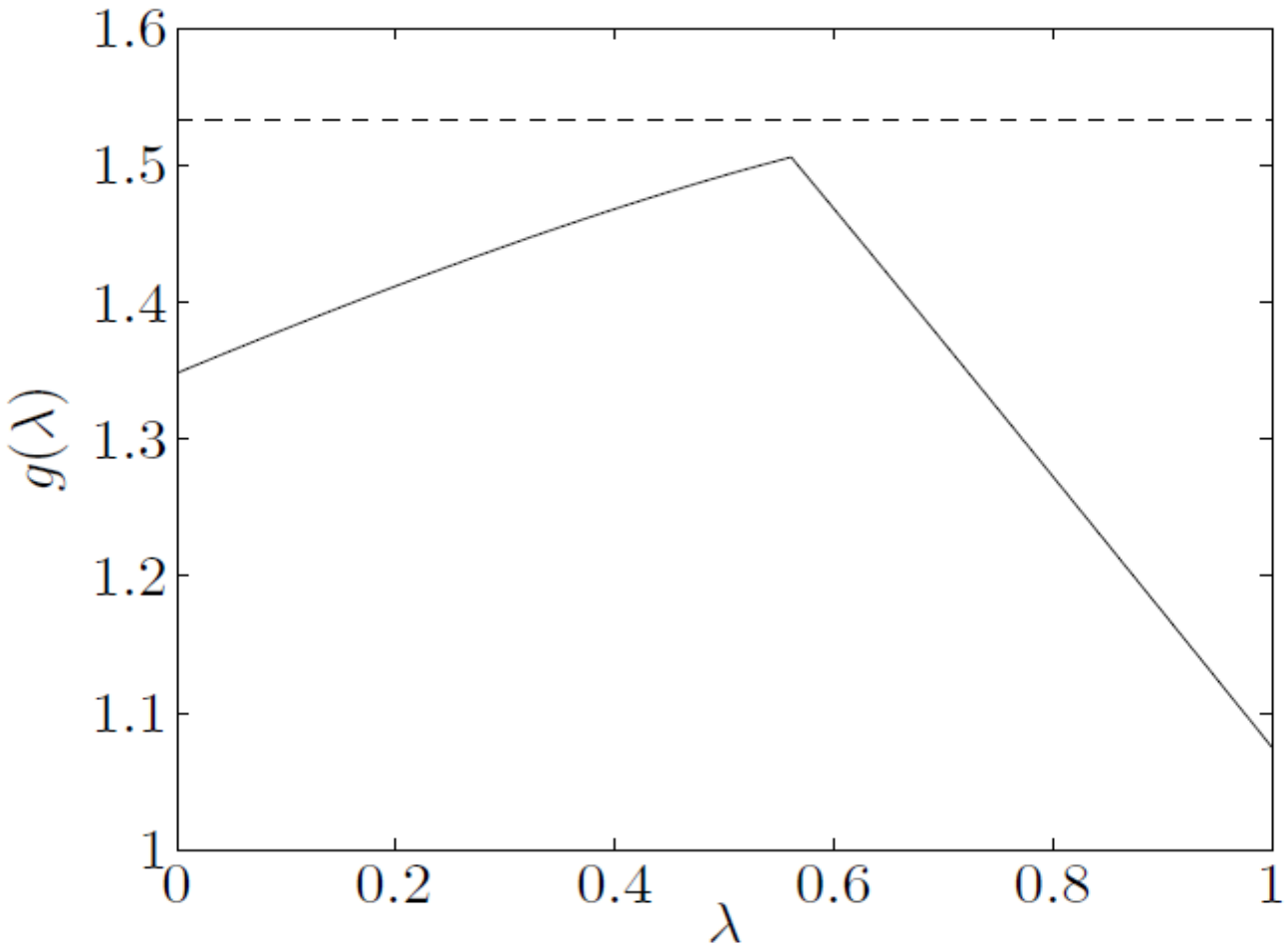


Figure 5.2 The dual function g for the problem in figure 5.1. Neither f_0 nor f_1 is convex, but the dual function is concave. The horizontal dashed line shows $p^\star$, the optimal value of the problem.

- Linear approximation interpretation
- The Lagrangian and lower bound property can be given a simple interpretation, based on a linear approximation of the indicator functions of the sets $\{0\}$ and $-R_+$.

We first rewrite the original problem (5.1) as an unconstrained problem,

$$\text{minimize} \quad f_0(x) + \sum_{i=1}^{m} I_-(f_i(x)) + \sum_{i=1}^{p} I_0(h_i(x)), \tag{5.3}$$

where $I_- : \mathbf{R} \to \mathbf{R}$ is the indicator function for the nonpositive reals,

$$I_-(u) = \begin{cases} 0 & u \leq 0 \\ \infty & u > 0, \end{cases}$$

and similarly, I_0 is the indicator function of $\{0\}$. In the formulation (5.3), the function $I_-(u)$ can be interpreted as expressing our irritation or displeasure associated with a constraint function value $u = f_i(x)$: It is zero if $f_i(x) \leq 0$, and infinite if $f_i(x) > 0$. In a similar way, $I_0(u)$ gives our displeasure for an equality constraint value $u = h_i(x)$. We can think of I_- as a "brick wall" or "infinitely hard" displeasure function; our displeasure rises from zero to infinite as $f_i(x)$ transitions from nonpositive to positive.

Now suppose in the formulation (5.3) we replace the function $I_-(u)$ with the linear function $\lambda_i u$, where $\lambda_i \geq 0$, and the function $I_0(u)$ with $\nu_i u$. The objective becomes the Lagrangian function $L(x, \lambda, \nu)$, and the dual function value $g(\lambda, \nu)$ is the optimal value of the problem

$$\text{minimize} \quad L(x, \lambda, \nu) = f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{p} \nu_i h_i(x). \tag{5.4}$$

In this formulation, we use a linear or "soft" displeasure function in place of I_- and I_0. For an inequality constraint, our displeasure is zero when $f_i(x) = 0$, and is positive when $f_i(x) > 0$ (assuming $\lambda_i > 0$); our displeasure grows as the constraint becomes "more violated". Unlike the original formulation, in which any nonpositive value of $f_i(x)$ is acceptable, in the soft formulation we actually derive pleasure from constraints that have margin, *i.e.*, from $f_i(x) < 0$.

Clearly the approximation of the indicator function $I_-(u)$ with a linear function $\lambda_i u$ is rather poor. But the linear function is at least an *underestimator* of the indicator function. Since $\lambda_i u \leq I_-(u)$ and $\nu_i u \leq I_0(u)$ for all u, we see immediately that the dual function yields a lower bound on the optimal value of the original problem.

Applied Probability and Statistics

Module-1, Lecture-13

By: Dr. A K Yadav (9911375598)
Dept of CSE, ASET, AUUP

In this chapter we discuss *interior-point methods* for solving convex optimization problems that include inequality constraints,

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leq 0, \quad i = 1, \ldots, m \\ & Ax = b, \end{array} \tag{11.1}$$

where $f_0, \ldots, f_m : \mathbf{R}^n \to \mathbf{R}$ are convex and twice continuously differentiable, and $A \in \mathbf{R}^{p \times n}$ with $\mathbf{rank}\, A = p < n$. We assume that the problem is solvable, *i.e.*, an optimal $x^\star$ exists. We denote the optimal value $f_0(x^\star)$ as $p^\star$.

We also assume that the problem is strictly feasible, *i.e.*, there exists $x \in \mathcal{D}$ that satisfies $Ax = b$ and $f_i(x) < 0$ for $i = 1, \ldots, m$. This means that Slater's constraint qualification holds, so there exist dual optimal $\lambda^\star \in \mathbf{R}^m$, $\nu^\star \in \mathbf{R}^p$, which together with $x^\star$ satisfy the KKT conditions

$$\begin{aligned} Ax^\star = b, \quad f_i(x^\star) &\leq 0, \quad i = 1, \ldots, m \\ \lambda^\star &\succeq 0 \\ \nabla f_0(x^\star) + \textstyle\sum_{i=1}^m \lambda_i^\star \nabla f_i(x^\star) + A^T \nu^\star &= 0 \\ \lambda_i^\star f_i(x^\star) &= 0, \quad i = 1, \ldots, m. \end{aligned} \tag{11.2}$$

Our goal is to approximately formulate the inequality constrained problem (11.1) as an equality constrained problem to which Newton's method can be applied. Our first step is to rewrite the problem (11.1), making the inequality constraints implicit in the objective:

$$\begin{array}{ll} \text{minimize} & f_0(x) + \sum_{i=1}^m I_-(f_i(x)) \\ \text{subject to} & Ax = b, \end{array} \tag{11.3}$$

where $I_- : \mathbf{R} \to \mathbf{R}$ is the indicator function for the nonpositive reals,

$$I_-(u) = \begin{cases} 0 & u \leq 0 \\ \infty & u > 0. \end{cases}$$

Logarithmic barrier

The basic idea of the barrier method is to approximate the indicator function I_- by the function

$$\widehat{I}_-(u) = -(1/t)\log(-u), \qquad \mathbf{dom}\,\widehat{I}_- = -\mathbf{R}_{++},$$

where $t > 0$ is a parameter that sets the accuracy of the approximation. Like I_-, the function $\widehat{I}_-$ is convex and nondecreasing, and (by our convention) takes on the value ∞ for $u > 0$. Unlike I_-, however, $\widehat{I}_-$ is differentiable and closed: it increases to ∞ as u increases to 0. Figure 11.1 shows the function I_-, and the approximation $\widehat{I}_-$, for several values of t. As t increases, the approximation becomes more accurate.

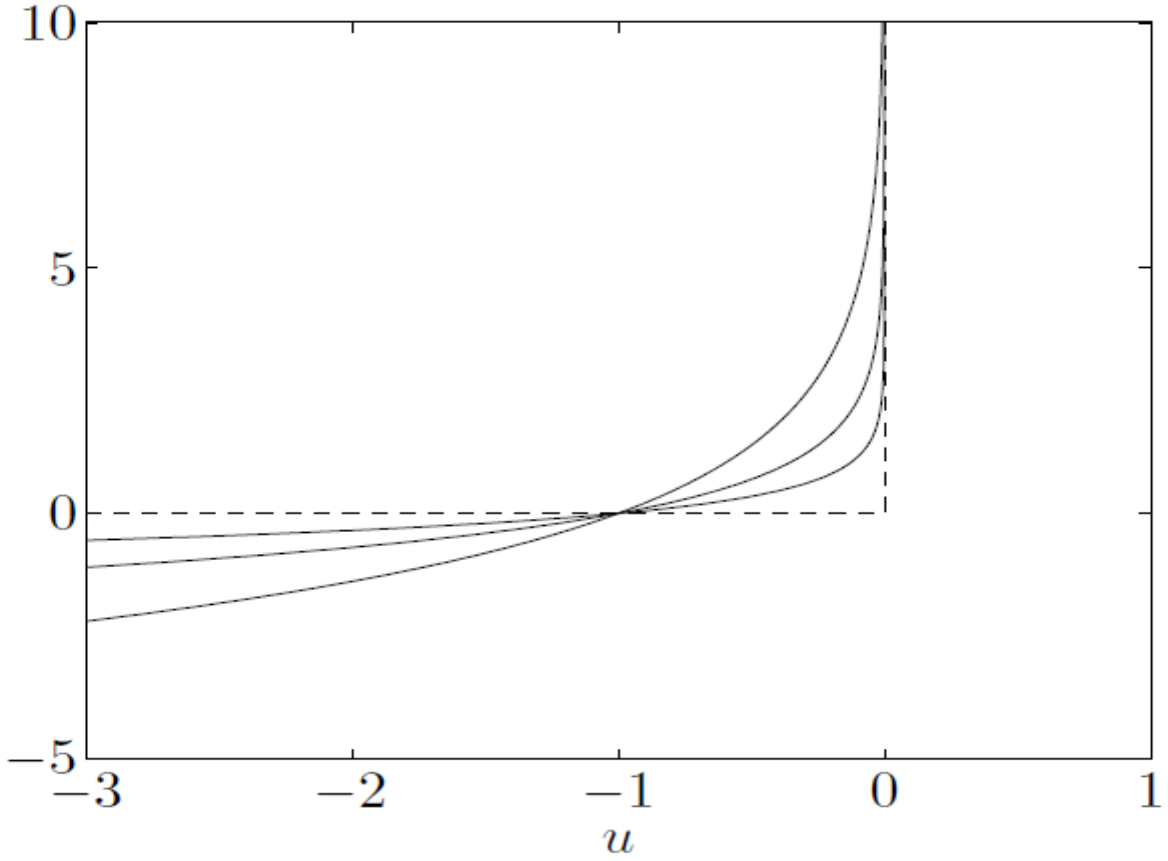


Figure 11.1 The dashed lines show the function $I_-(u)$, and the solid curves show $\widehat{I}_-(u) = -(1/t)\log(-u)$, for $t = 0.5,\ 1,\ 2$. The curve for $t = 2$ gives the best approximation.

Substituting $\widehat{I}_-$ for I_- in (11.3) gives the approximation

$$\begin{array}{ll} \text{minimize} & f_0(x) + \sum_{i=1}^{m} -(1/t)\log(-f_i(x)) \\ \text{subject to} & Ax = b. \end{array} \tag{11.4}$$

The objective here is convex, since $-(1/t)\log(-u)$ is convex and increasing in u, and differentiable. Assuming an appropriate closedness condition holds, Newton's method can be used to solve it.

The function

$$\phi(x) = -\sum_{i=1}^{m} \log(-f_i(x)), \tag{11.5}$$

with $\mathbf{dom}\,\phi = \{x \in \mathbf{R}^n \mid f_i(x) < 0,\ i = 1, \ldots, m\}$, is called the *logarithmic barrier* or *log barrier* for the problem (11.1). Its domain is the set of points that satisfy the inequality constraints of (11.1) strictly. No matter what value the positive parameter t has, the logarithmic barrier grows without bound if $f_i(x) \to 0$, for any i.

On the other hand, when the parameter t is large, the function $f_0 + (1/t)\phi$ is difficult to minimize by Newton's method, since its Hessian varies rapidly near the boundary of the feasible set. We will see that this problem can be circumvented by solving a *sequence* of problems of the form (11.4), increasing the parameter t (and therefore the accuracy of the approximation) at each step, and starting each Newton minimization at the solution of the problem for the previous value of t.

For future reference, we note that the gradient and Hessian of the logarithmic barrier function ϕ are given by

$$\nabla\phi(x) = \sum_{i=1}^{m} \frac{1}{-f_i(x)} \nabla f_i(x),$$

$$\nabla^2\phi(x) = \sum_{i=1}^{m} \frac{1}{f_i(x)^2} \nabla f_i(x) \nabla f_i(x)^T + \sum_{i=1}^{m} \frac{1}{-f_i(x)} \nabla^2 f_i(x)$$

Central path

We now consider in more detail the minimization problem (11.4). It will simplify notation later on if we multiply the objective by t, and consider the equivalent problem

$$\begin{array}{ll} \text{minimize} & tf_0(x) + \phi(x) \\ \text{subject to} & Ax = b, \end{array} \tag{11.6}$$

which has the same minimizers. We assume for now that the problem (11.6) can be solved via Newton's method, and, in particular, that it has a unique solution for each $t > 0$. (We will discuss this assumption in more detail in §11.3.3.)

For $t > 0$ we define $x^\star(t)$ as the solution of (11.6). The *central path* associated with problem (11.1) is defined as the set of points $x^\star(t)$, $t > 0$, which we call the *central points*. Points on the central path are characterized by the following necessary and sufficient conditions: $x^\star(t)$ is strictly feasible, *i.e.*, satisfies

$$Ax^\star(t) = b, \qquad f_i(x^\star(t)) < 0, \quad i = 1, \ldots, m,$$

and there exists a $\hat{\nu} \in \mathbf{R}^p$ such that

$$\begin{aligned} 0 &= t\nabla f_0(x^\star(t)) + \nabla\phi(x^\star(t)) + A^T\hat{\nu} \\ &= t\nabla f_0(x^\star(t)) + \sum_{i=1}^{m} \frac{1}{-f_i(x^\star(t))}\nabla f_i(x^\star(t)) + A^T\hat{\nu} \end{aligned} \tag{11.7}$$

holds.

Dual points from central path

From (11.7) we can derive an important property of the central path: Every central point yields a dual feasible point, and hence a lower bound on the optimal value $p^\star$. More specifically, define

$$\lambda_i^\star(t) = -\frac{1}{tf_i(x^\star(t))}, \quad i = 1, \ldots, m, \qquad \nu^\star(t) = \hat{\nu}/t. \tag{11.10}$$

We claim that the pair $\lambda^\star(t)$, $\nu^\star(t)$ is dual feasible.

First, it is clear that $\lambda^\star(t) \succ 0$ because $f_i(x^\star(t)) < 0$, $i = 1, \ldots, m$. By expressing the optimality conditions (11.7) as

$$\nabla f_0(x^\star(t)) + \sum_{i=1}^{m} \lambda_i^\star(t) \nabla f_i(x^\star(t)) + A^T \nu^\star(t) = 0,$$

we see that $x^\star(t)$ minimizes the Lagrangian

$$L(x, \lambda, \nu) = f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \nu^T (Ax - b),$$

for $\lambda = \lambda^\star(t)$ and $\nu = \nu^\star(t)$, which means that $\lambda^\star(t)$, $\nu^\star(t)$ is a dual feasible pair. Therefore the dual function $g(\lambda^\star(t), \nu^\star(t))$ is finite, and

$$\begin{aligned} g(\lambda^\star(t), \nu^\star(t)) &= f_0(x^\star(t)) + \sum_{i=1}^{m} \lambda_i^\star(t) f_i(x^\star(t)) + \nu^\star(t)^T (Ax^\star(t) - b) \\ &= f_0(x^\star(t)) - m/t. \end{aligned}$$

In particular, the duality gap associated with $x^\star(t)$ and the dual feasible pair $\lambda^\star(t)$, $\nu^\star(t)$ is simply m/t. As an important consequence, we have

$$f_0(x^\star(t)) - p^\star \leq m/t,$$

i.e., $x^\star(t)$ is no more than m/t-suboptimal. This confirms the intuitive idea that $x^\star(t)$ converges to an optimal point as $t \to \infty$.

We can also interpret the central path conditions (11.7) as a continuous deformation of the KKT optimality conditions (11.2). A point x is equal to $x^\star(t)$ if and only if there exists λ, ν such that

$$\begin{array}{rcll} Ax = b, \quad f_i(x) & \leq & 0, & i = 1, \ldots, m \\ \lambda & \succeq & 0 & \\ \nabla f_0(x) + \sum_{i=1}^m \lambda_i \nabla f_i(x) + A^T \nu & = & 0 & \\ -\lambda_i f_i(x) & = & 1/t, & i = 1, \ldots, m. \end{array} \tag{11.11}$$

The only difference between the KKT conditions (11.2) and the centrality conditions (11.11) is that the complementarity condition $-\lambda_i f_i(x) = 0$ is replaced by the condition $-\lambda_i f_i(x) = 1/t$. In particular, for large t, $x^\star(t)$ and the associated dual point $\lambda^\star(t)$, $\nu^\star(t)$ 'almost' satisfy the KKT optimality conditions for (11.1).

Applied Probability and Statistics

Module-1, Lecture-14

By: Dr. A K Yadav (9911375598)
Dept of CSE, ASET, AUUP

Projected gradients

- Constrained and unconstrained problem
- Understanding the geometry of projection
- PGD is a special case of proximal gradient

Constrained and unconstrained problem

For unconstrained minimization problem

$$\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x}),$$

any $\mathbf{x}$ in $\mathbb{R}^n$ can be a solution.

For constrained minimization problem with a given set $\mathcal{Q} \subset \mathbb{R}^n$

$$\min_{\mathbf{x}\in\mathcal{Q}} f(\mathbf{x}),$$

not any $\mathbf{x}$ can be a solution, the solution has to be inside the set $\mathcal{Q}$

An example of constrained minimization problem:

$$\min_{\mathbf{x}\in\mathbb{R}^n} \|\mathbf{Ax}-\mathbf{b}\|_2^2 \ \text{ s.t. } \ \|\mathbf{x}\|_2 \leq 1$$

can be expressed as

$$\min_{\|\mathbf{x}\|_2\leq 1} \|\mathbf{Ax}-\mathbf{b}\|_2^2.$$

Solving unconstrained problem by gradient descent

Gradient Descent (GD) is a standard (easy and simple) way to solve **unconstrained** optimization problem.

Starting from an initial point $\mathbf{x}_0 \in \mathbb{R}^n$, GD iterates the following equation until a stopping condition is met:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \nabla f(\mathbf{x}_k),$$

where ∇f is the gradient of f, the parameter $\alpha \geq 0$ is the step size, and k is the iteration counter.

Question: how about **constrained** problem? Is it possible to **tune** GD to fit constrained problem?

Answer: yes, and the key is **projection**.

Remark: If f is not differentiable, we can replace gradient by subgradient, and we get the so-called subgradient method.

Solving constrained problem by projected gradient descent

Projected Gradient Descent (PGD) is a standard (easy and simple) way to solve **constrained** optimization problem.

Consider a constraint set $\mathcal{Q} \subset \mathbb{R}^n$, starting from a initial point $\mathbf{x}_0 \in \mathcal{Q}$, PGD iterates the following equation until a stopping condition is met:

$$\mathbf{x}_{k+1} = P_{\mathcal{Q}}\Big(\mathbf{x}_k - \alpha_k \nabla f(\mathbf{x}_k)\Big).$$

$P_{\mathcal{Q}}(\,.\,)$ is the projection operator, and itself is also an optimization problem:

$$P_{\mathcal{Q}}(\mathbf{x}_0) = \arg\min_{\mathbf{x}\in\mathcal{Q}} \frac{1}{2}\|\mathbf{x} - \mathbf{x}_0\|_2^2,$$

i.e. given a point $\mathbf{x}_0$, $P_{\mathcal{Q}}$ try to find a point $\mathbf{x} \in \mathcal{Q}$ which is "closest" to $\mathbf{x}_0$.

$P_{\mathcal{Q}}(\,.\,)$ is a function from $\mathbb{R}^n$ to $\mathbb{R}^n$, and itself is an optimization problem:

$$P_{\mathcal{Q}}(\mathbf{x}_0) = \arg\min_{\mathbf{x}\in\mathcal{Q}} \frac{1}{2}\|\mathbf{x} - \mathbf{x}_0\|_2^2.$$

PGD is an "economic" algorithm if the problem is easy to solve. This is not true for general $\mathcal{Q}$ and there are lots of constraint sets that are very difficult to project onto.

If $\mathcal{Q}$ is a convex set, the optimization problem has a unique solution.

If $\mathcal{Q}$ is nonconvex, the solution to $P_{\mathcal{Q}}(\mathbf{x}_0)$ may not be unique: it gives more than one solution.

Comparing PGD to GD

GD

1. Pick an initial point $\mathbf{x}_0 \in \mathbb{R}^n$
2. Loop until stopping condition is met:
   - 2.1 Descent direction: pick the descent direction as $-\nabla f(\mathbf{x}_k)$
   - 2.2 Stepsize: pick a step size α_k
   - 2.3 Update: $\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \nabla f(\mathbf{x}_k)$

PGD

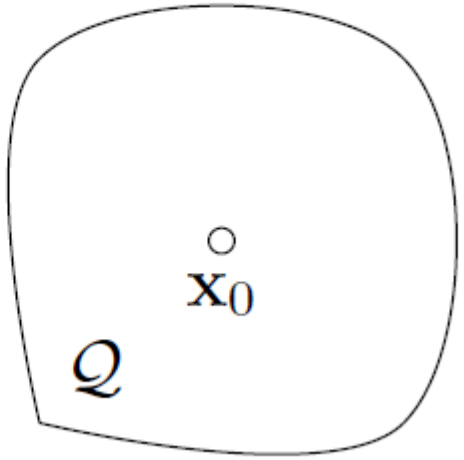
1. Pick an initial point $\mathbf{x}_0 \in \mathcal{Q}$
2. Loop until stopping condition is met:
   - 2.1 Descent direction: pick the descent direction as $-\nabla f(\mathbf{x}_k)$
   - 2.2 Stepsize: pick a step size α_k
   - 2.3 Update: $\mathbf{y}_{k+1} = \mathbf{x}_k - \alpha_k \nabla f(\mathbf{x}_k)$
   - 2.4 Projection: $\mathbf{x}_{k+1} = \underset{\mathbf{x} \in \mathcal{Q}}{\operatorname{argmin}} \frac{1}{2} \|\mathbf{x} - \mathbf{y}_{k+1}\|_2^2$

PGD has one more step: the projection.

The idea of PGD is simple: if the point $\mathbf{x}_k - \alpha_k \nabla f(\mathbf{x}_k)$ after the gradient update is leaving the set $\mathcal{Q}$, project it back.

Understanding the geometry of projection

Consider a convex set $\mathcal{Q}$ and a point $\mathbf{x}_0 \in \mathcal{Q}$.

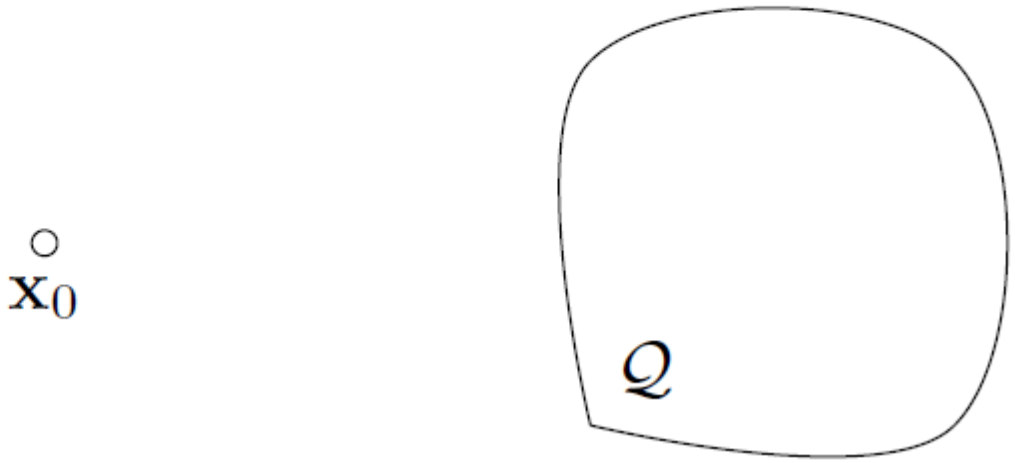


As $\mathbf{x}_0 \in \mathcal{Q}$, the closest point to $\mathbf{x}_0$ in $\mathcal{Q}$ will be $\mathbf{x}_0$ itself.

The distance between a point to itself is zero.

Mathematically: $\frac{1}{2}\|\mathbf{x} - \mathbf{x}_0\|_2^2 = 0$ gives $\mathbf{x} = \mathbf{x}_0$.

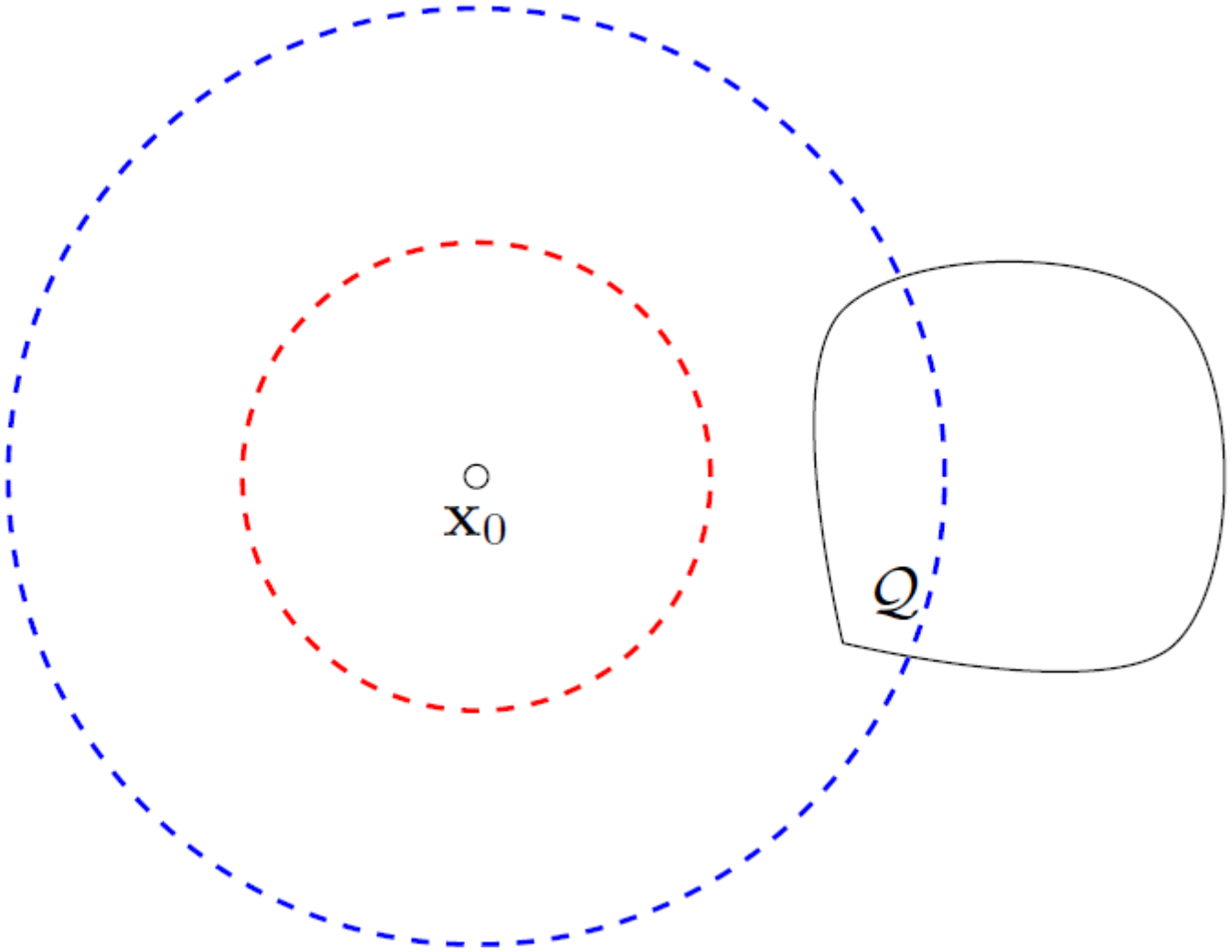
Now consider a convex set $\mathcal{Q}$ and a point $\mathbf{x}_0 \notin \mathcal{Q}$: outside $\mathcal{Q}$.



The circles are L_2 norm ball centered at $\mathbf{x}_0$ with different radius.

Points on these circles are **equidistant** to $\mathbf{x}_0$ (with different L_2 distance on different circles).

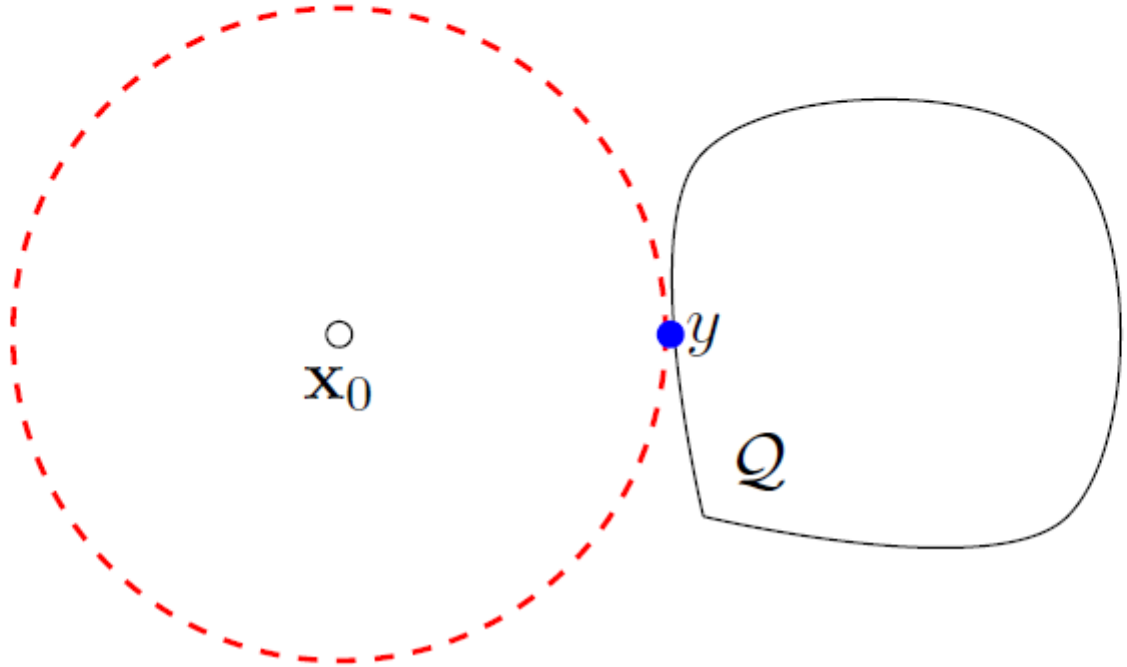
Note that some points on the blue circle are inside $\mathcal{Q}$.


$\mathbf{x}_0$
Q

The point inside $\mathcal{Q}$ which is closest to $\mathbf{x}_0$ is the point where the L_2 norm ball "touches" $\mathcal{Q}$.

In this example, the blue point $\mathbf{y}$ is the solution to

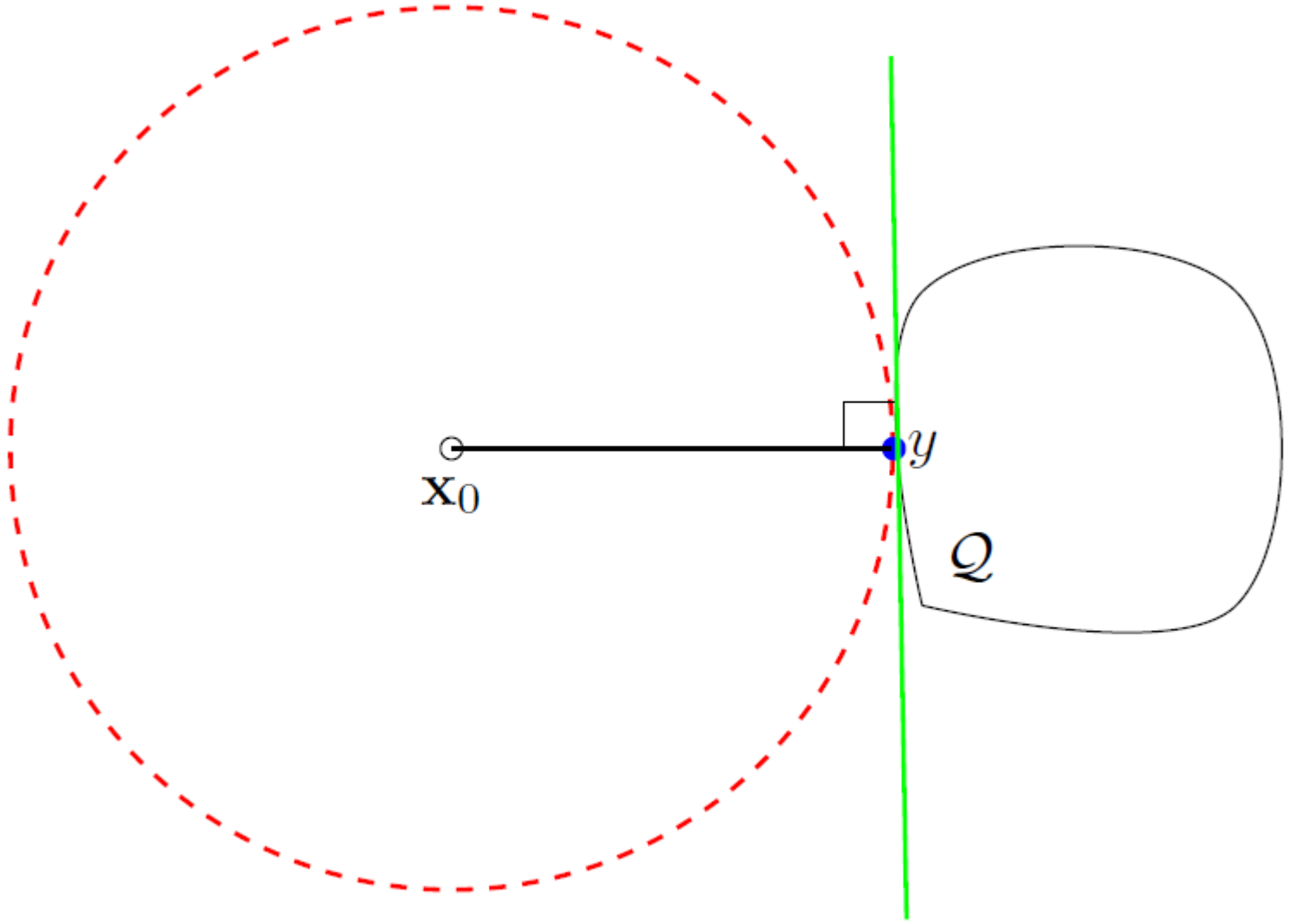
$$P_{\mathcal{Q}}(\mathbf{x}_0) = \underset{\mathbf{x}\in\mathcal{Q}}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{x} - \mathbf{x}_0\|_2^2.$$



In fact, it can be proved that, such point is always located on the **boundary** of $\mathcal{Q}$ for $\mathbf{x}_0 \notin \mathcal{Q}$. That is, mathematically,

$$\underset{\mathbf{x}\in\mathcal{Q}}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{x}-\mathbf{x}_0\|_2^2 \in \text{bd}\mathcal{Q} \text{ if } \mathbf{x}_0 \notin \mathcal{Q}.$$

Note that the projection is **orthogonal**: the blue point $\mathbf{y}$ is always on a straight line that is tangent to the norm ball and $\mathcal{Q}$.


$\mathbf{x}_0$
y
Q

The indicator function, denoted as $i(\mathbf{x})$, of a set $\mathcal{Q}$ is defined as follows: if $\mathbf{x} \in \mathcal{Q}$, then $i(\mathbf{x}) = 0$; if $\mathbf{x} \notin \mathcal{Q}$, then $i(\mathbf{x}) = \infty$.

With the indicator function, constrained problem has two equivalent expressions

$$\min_{\mathbf{x} \in \mathcal{Q}} f(\mathbf{x}) \quad \equiv \quad \min_{\mathbf{x}} f(\mathbf{x}) + i(\mathbf{x}).$$

Proximal gradient is a method to solve the optimization problem of a sum of differentiable and a non-differentiable function:

$$\min_{\mathbf{x}} f(\mathbf{x}) + g(\mathbf{x}),$$

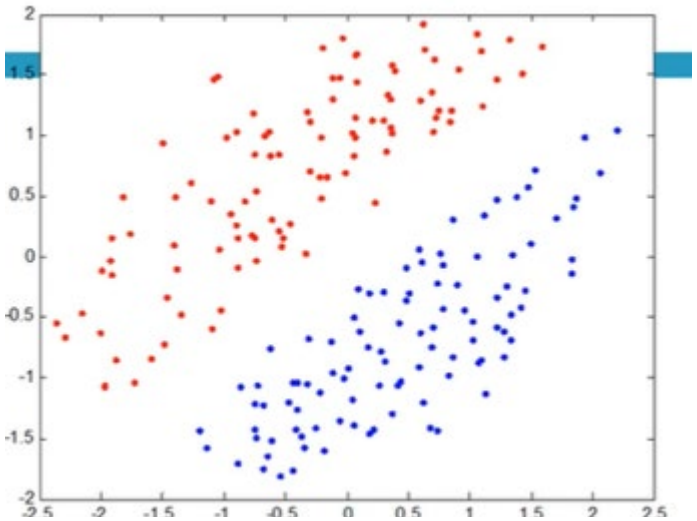
where g is a non-differentiable function.

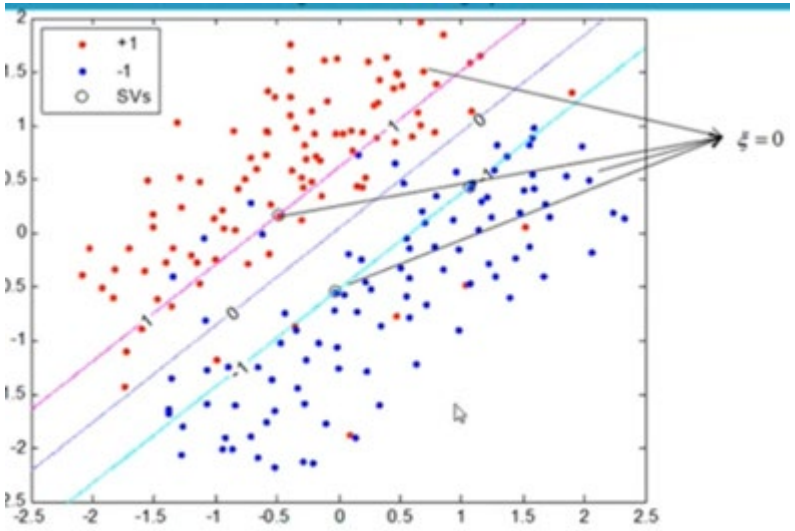
PGD is in fact the special case of proximal gradient where $g(\mathbf{x})$ is the indicator function of the constrain set.

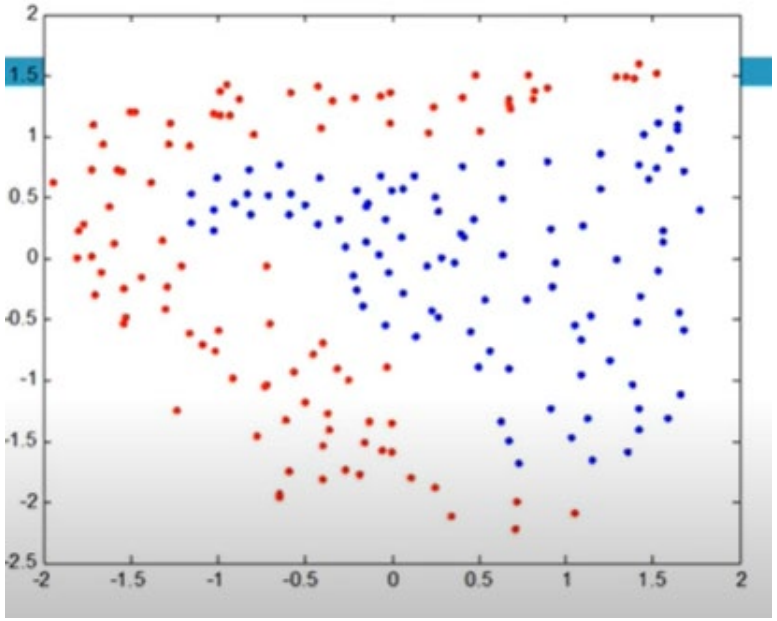
Applied Probability and Statistics

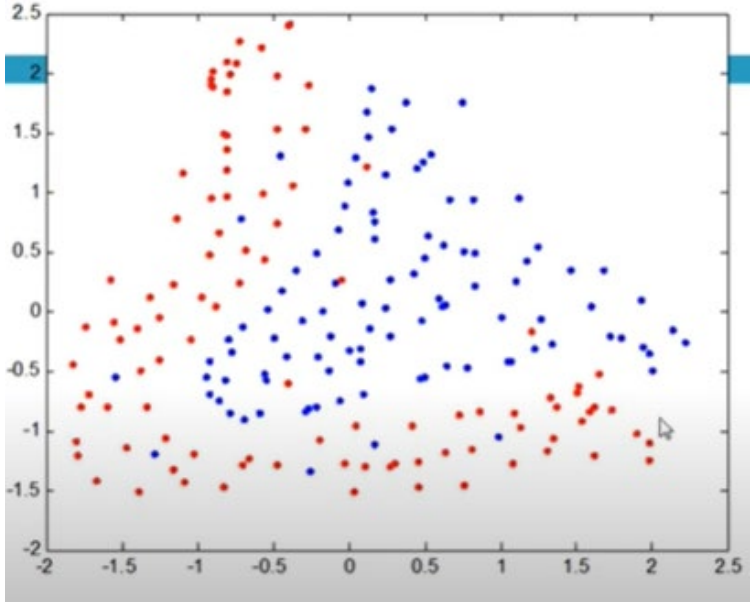
Module-1, Lecture-15

By: Dr. A K Yadav (9911375598)
Dept of CSE, ASET, AUUP




+1
-1
SVs
$\xi = 0$





The underlying problem -

- Classification.
  - Linearly separable
  - Nonlinearly separable
  - Non-separable linear boundary
  - Non-separable nonlinear boundary

- SVM: Maximum Margin Classifier
- Margin: Perpendicular distance to the closest point x_n in dataset from the line $y(x) = 0$
- Finds w & b to maximize the margin.

Hard Margin SVM

- Training data: $(\mathrm{x}_n, t_n);\ \mathrm{x}_n \in R^n;\ t_n \in \{-1, 1\}$
- Feature Mapping: $\phi : \mathrm{x} \to R^k$
- Value: $y(\mathrm{x}_n) = w^T \phi(\mathrm{x}_n) + b\ ;\ w \in R^k; b \in R$
- Class-1: $y(\mathrm{x}) \geq 1$
- Class-2: $y(\mathrm{x}) \leq -1$

Theoretical reason behind choosing the numbers

- Decision Boundary: $y(\mathrm{x}) = 0$

Distance

- Distance of a point x_n from $y(\mathrm{x}) = 0$:

$$dist(\mathrm{x}_n) = \frac{|\, y(\mathrm{x}_n)\,|}{\| w \|_2} = \frac{t_n y(\mathrm{x}_n)}{\| w \|_2} = \frac{t_n (w^T \phi(\mathrm{x}_n) + b)}{\| w \|_2}$$

- To maximize the margin:

$$\arg\max_{w,b} \left(\frac{1}{\| w \|_2} \min_n [t_n (w^T \phi(\mathrm{x}_n) + b)] \right)$$

- If we set $w \to kw$ and $b \to kb$ then the distance is unchanged.
- Exploiting this fact, we set the closest point as:

$$t_n(w^T\phi(\mathrm{x}_n)+b)=1$$

- For all other points: $t_n(w^T\phi(\mathrm{x}_n)+b)\geq 1$

Support Vector Machine

- For a separable (in feature space) data set:

$$\underset{w,b}{\text{minimize}} \frac{1}{2} \| w \|_2^2$$

$$subject\ to:$$

$$t_n(w^T \phi(\mathrm{x}_n) + b) \geq 1$$

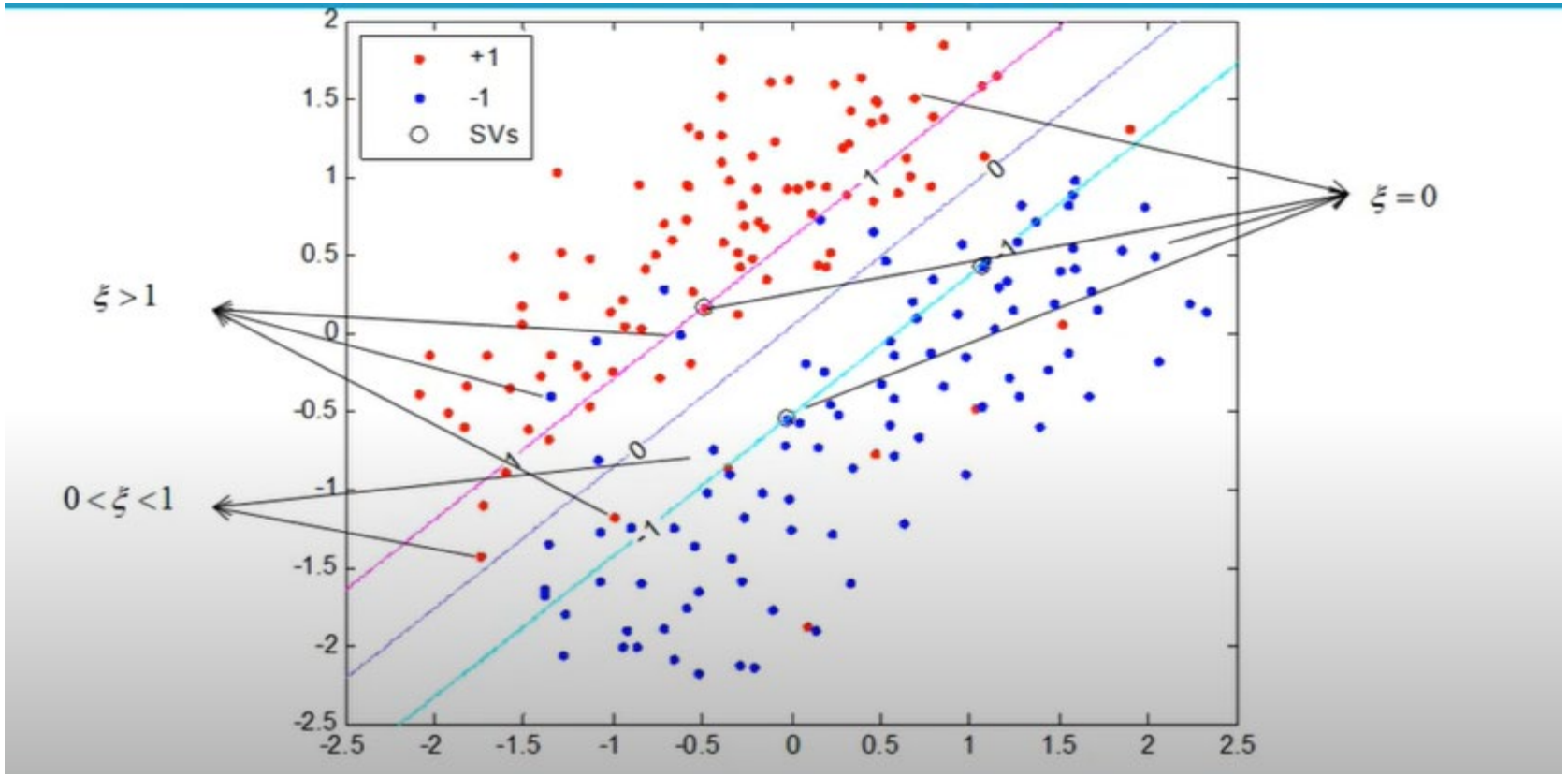
Quadratic Programming Problem

- Primal Form of SVM

Soft Margin SVM

- When the classes are overlapping or when we have no clue about the dataset.
- Relax the constraint with slack variable:

$$t_n(w^T\phi(\mathrm{x}_n)+b)\geq 1-\xi_n \;\; ;\xi_n \geq 0$$


+1
-1
SVs
$\xi = 0$
$\xi > 1$
$0 < \xi < 1$
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2.5
-2.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5

Soft Margin SVM

- SVM with relaxed constraints allows some misclassifications.
- Convex Quadratic Program

$$\underset{w,b}{\text{minimize}} \quad \frac{1}{2}\| w \|_2^2 + \alpha \sum_{i=1}^{N} \xi_i$$

$$subject\ to:$$

$$t_n(w^T \phi(\mathrm{x}_n) + b) \geq 1 - \xi_n$$

$$\xi_i \geq 0 \ \ for\ i = 1, \ldots, N$$

Primal Form: When to use ?

- If the dimensionality is very low.
- Easy to visualize what kind of features are required.
- If it's evident that the data set is linearly separable in the feature space defined by $\phi(\mathrm{x})$

Dual Problem: Usefulness -

- When we have no clue about the features or degree of the polynomial.
- If the dimension is large (say 10000), it's not possible to visualize the data.
- Better is to use Kernel Trick, which is what dual formulation is about.

Lagrangian Formulation:

- Lagrange Multiplier: λ, γ
  - Hard Margin:

$$L(w,b,\lambda) = \frac{1}{2}\| w \|_2^2 - \sum_{i=1}^{N} \lambda_i (t_i (w^T \phi(\mathrm{x}_i) + b) - 1)$$

  - Soft Margin:

$$L(w,b,\lambda,\gamma) = \frac{1}{2}\| w \|_2^2 + \sum_{n=1}^{N} \xi_n - \sum_{n=1}^{N} \lambda_n (t_n (w^T \phi(\mathrm{x}_n) + b) - 1 + \xi_n) - \sum_{n=1}^{N} \gamma_n \xi_n$$

Apply the KKT Conditions

- Gradient w.r.t w and b should vanish.

Hard Margin

$$\text{w.r.t } w : \frac{1}{2} \times 2w - \sum_{i=1}^{N} \lambda_i t_i \phi(x_i) = 0$$

$$w = \sum_{i=1}^{N} \lambda_i t_i \phi(x_i)$$

$$\text{w.r.t } b : -\sum_{i=1}^{N} \lambda_i t_i = 0 \Rightarrow \sum_{i=1}^{N} \lambda_i t_i = 0$$

- Substituting the value of w in the primal problem:

$$L(\lambda) = \sum_{n=1}^{N} \lambda_n - 0.5 \sum_{n=1}^{N} \sum_{m=1}^{N} \lambda_n t_n \lambda_m t_m < \phi(\mathrm{x}_n), \phi(\mathrm{x}_m) >$$

- Reformulating the objective: (k is a mercer kernel)

$$L(\lambda) = \sum_{n=1}^{N} \lambda_n - 0.5 \sum_{n=1}^{N} \sum_{m=1}^{N} \lambda_n t_n \lambda_m t_m k(\mathrm{x}_n, \mathrm{x}_m)$$

Hard Margin Dual SVM

- Form a vector: $\beta = (\beta_1, \ldots, \beta_N)$; where components $\beta_n = \lambda_n t_n$
- The optimization problem is

$$\text{maximize} \quad \sum_{n=1}^{N} \lambda_n - 0.5 \beta^T K \beta$$

$$subject\ to: \quad \lambda \succeq 0$$

$$\sum_{i=1}^{N} \beta_i = 0$$

KKT Conditions

- Optimal variable will satisfy following conditions

  feasiblity: $t_n(w^T\phi(\mathrm{x}_n)+b)-1\geq 0$

  Lagrange multiplier: $\lambda_i \geq 0$

  Complementary Slackness: $\lambda_n(t_n y(\mathrm{x}_n)-1)=0$

- Solution:

$$y(\mathrm{x})=\sum_{n=1}^{N}\lambda_n t_n k(\mathrm{x},\mathrm{x}_n)+b$$

- Where b is derived using the complementary slackness

- By complementary slackness:

- if $\lambda_n > 0 \rightarrow$ active constraint:for such point: $t_n y(\mathrm{x}_n) = 1$ (support vector)
- if $\lambda_n = 0 \rightarrow$ inactive constraint , $t_n y(\mathrm{x}_n) > 1$ and will not play any role in decision making.
- Such data points can be discarded (sparse solution)

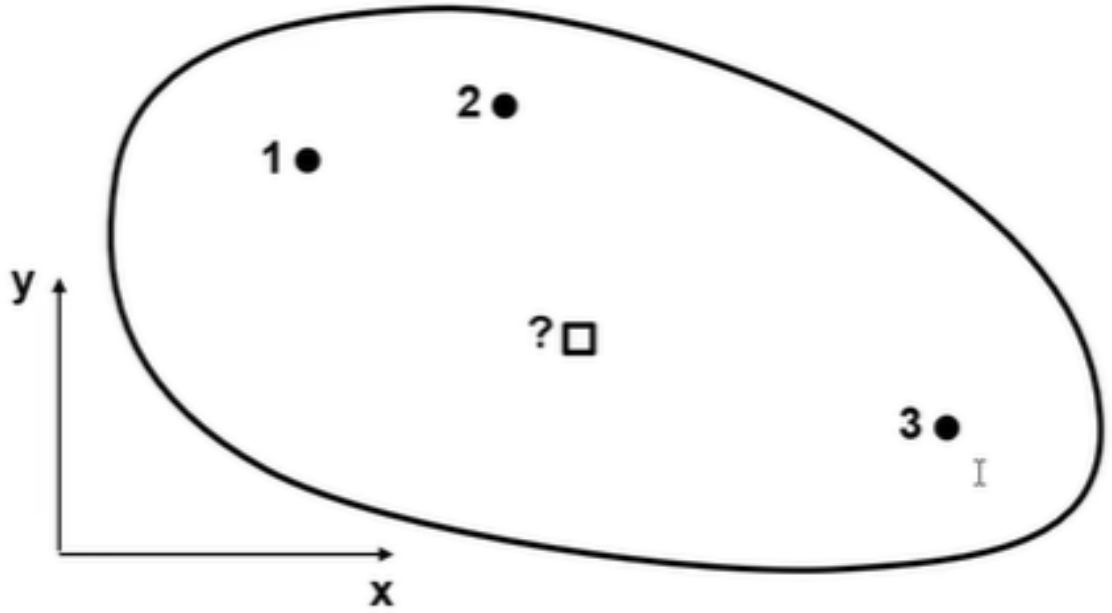
Applied Probability and Statistics

Module-1, Lecture-16

By: Dr. A K Yadav (9911375598)
Dept of CSE, ASET, AUUP

Kriging

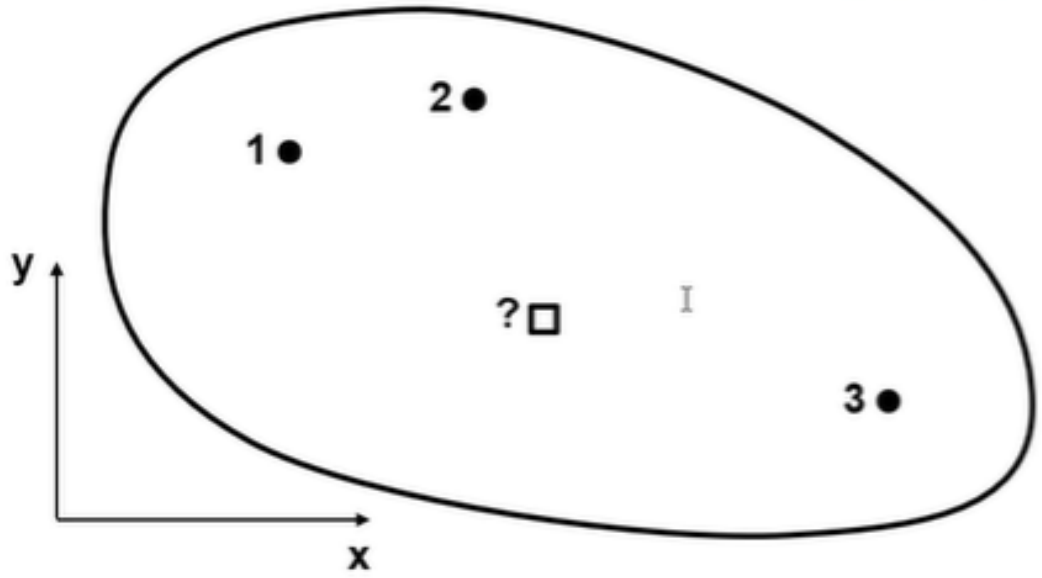
- Consider the case of estimating at some unsampled location:



- How would you do this given data, $z(\mathbf{u}_1)$, $z(\mathbf{u}_2)$, and $z(\mathbf{u}_3)$?

- Note: z is the variable of interest (e.g. porosity etc.) and $\mathbf{u}_i$ is the data locations.

- Consider the case of estimating at some unsampled location:



$z(\mathbf{u}_\alpha)$ is the data values

$z^*(\mathbf{u}_0)$ is an estimate

λ_α is the data weights

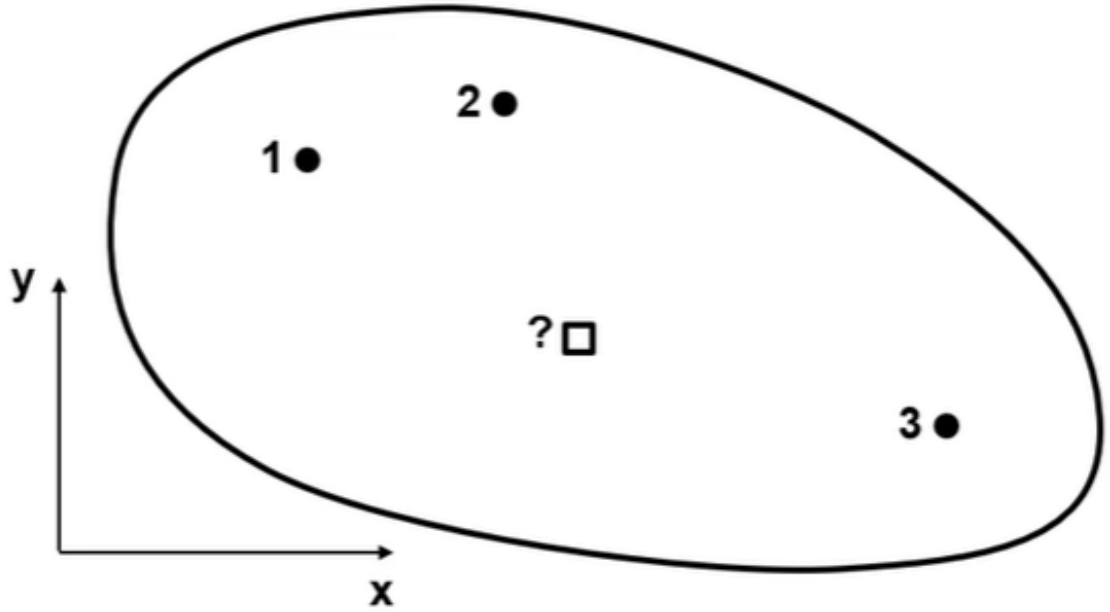
m_z is the global mean

- How would you do this given data, $z(\mathbf{u}_1)$, $z(\mathbf{u}_2)$, and $z(\mathbf{u}_3)$?

$$z^*(\mathbf{u}_0) = \sum_{\alpha=1}^{n} \lambda_\alpha z(\mathbf{u}_\alpha) + \left(1 - \sum_{\alpha=1}^{n} \lambda_\alpha\right) m_z$$

Unbiasedness Constraint
Weights sum to 1.0.

- Consider the case of estimating at some unsampled location:



- How would you do this given data, $z(\mathbf{u}_1)$, $z(\mathbf{u}_2)$, and $z(\mathbf{u}_3)$?

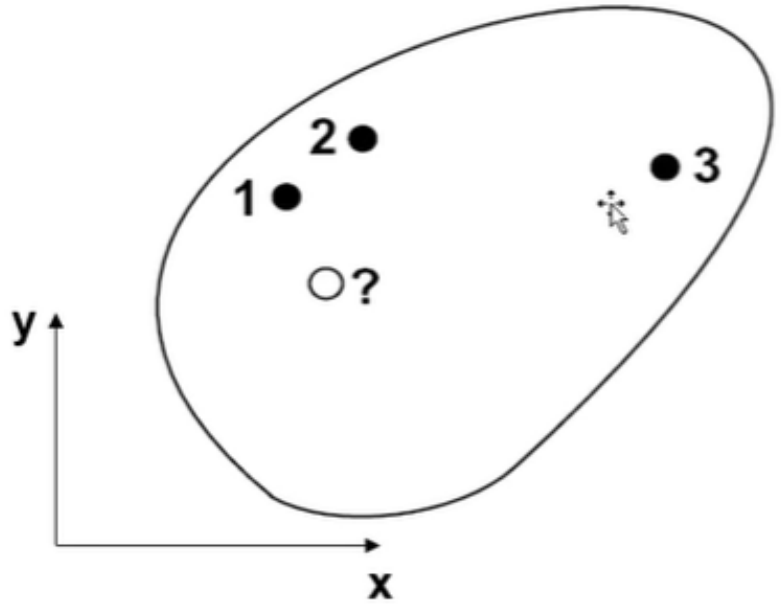
$$z^*(\mathbf{u}_0) - m_z(\mathbf{u}_0) = \sum_{\alpha=1}^{n} \lambda_\alpha \big(z(\mathbf{u}_\alpha) - m_z(\mathbf{u}_\alpha)\big)$$

In the case where the mean is non-stationary.

Given $y = z - m$, $y^*(\mathbf{u}_0) = \sum_{\alpha=1}^{n} \lambda_\alpha \, y(\mathbf{u}_\alpha)$

Simplified with residual, y.

- Consider the case of estimating at some unsampled location:

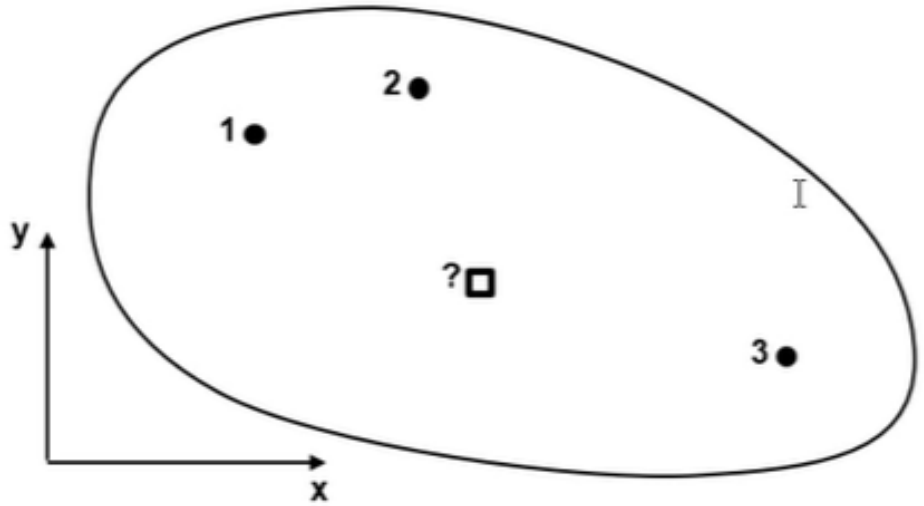


- Linear weighted, sound good. How do we get the weights? $\boldsymbol{\lambda_\alpha, \alpha = 1, \ldots, n}$

$$\boldsymbol{y^*(u_0) = \sum_{\alpha=1}^{n} \lambda_\alpha \, y(u_\alpha)}$$

Simplified with residual, y.

- Consider the case of estimating at some unsampled location:

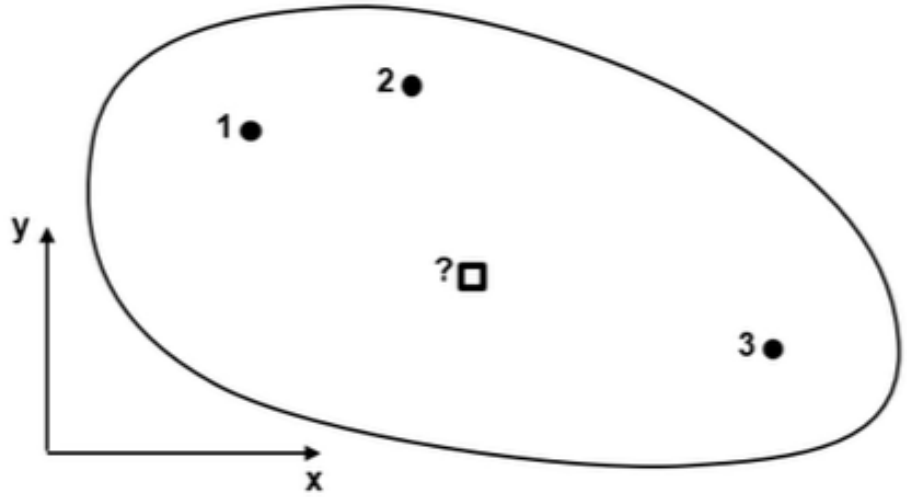


- Linear weighted, sound good. How do we get the weights? $\lambda_\alpha, \alpha = 1, \dots, n$
- Equal weighted / average? $\lambda_\alpha = 1/n$

Equal weight average of data

- What's wrong with that?

- Consider the case of estimating at some unsampled location:



- How do we get the weights? $\lambda_\alpha, \alpha = 1, \dots, n$

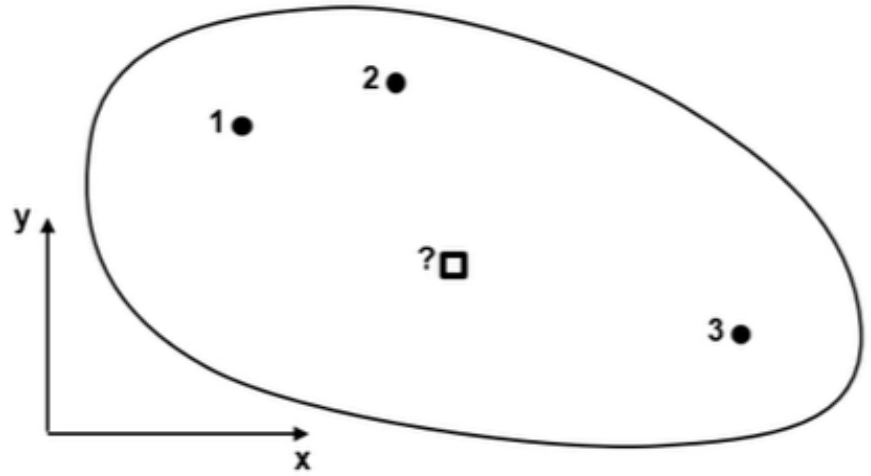
Inverse distance to power standardized so weights sum to 1.0.

- Inverse distance?

$$\lambda_\alpha = \frac{\frac{1}{dist(\mathbf{u}_0, \mathbf{u}_\alpha)^p}}{\sum_{\alpha=1}^{n} \lambda_\alpha}$$

- What's wrong with that?

- Consider the case of estimating at some unsampled location:



- How do we get the weights? $\lambda_\alpha, \alpha = 1, \ldots, n$

- It would be great to use weight that account for closeness (spatial correlation > distance alone), redundancy (once again with spatial correlation).

- How can we do that?

- Consider a linear estimator:

$$Y^*(\mathbf{u}) = \sum_{i=1}^{n} \lambda_i \cdot Y(\mathbf{u}_i)$$

where $Y(u_i)$ are the residual data (data values minus the mean) and $Y^*(u_i)$ is the estimate (add the mean back in when we are finished)

- The **estimation variance** is defined as:

Stationary Mean, Variogram

$$E\{Y\} = 0$$

$$2\gamma(\mathbf{h}) = E\left\{\left[Y(\mathbf{u}) - Y(\mathbf{u}+\mathbf{h})\right]^2\right\}$$

$$E\left\{\left[Y^*(u) - Y(u)\right]^2\right\} = \ldots$$

$$= E\left\{\left[Y^*(u)\right]^2\right\} - 2\,E\left\{Y^*(u)\,Y(u)\right\} + E\left\{\left[Y(u)\right]^2\right\}$$

$$= \sum_{i=1}^{n}\sum_{j=1}^{n} \lambda_i \lambda_j E\left\{Y(u_i)\,Y(u_j)\right\} - 2\sum_{i=1}^{n} \lambda_i E\left\{Y(u)\,Y(u_i)\right\} + C(0)$$

$$= \sum_{i=1}^{n}\sum_{j=1}^{n} \lambda_i \lambda_j C(u_i, u_j) - 2\sum_{i=1}^{n} \lambda_i C(u, u_i) + C(0)$$

redundancy closeness variance

$C(\mathbf{u}_i, \mathbf{u}_j)$ – covariance between data i and j, $C(\mathbf{u}_i, \mathbf{u})$covariance between data and unknown location and $C(0)$ is the variance.

- Optimal weights $\lambda_i, i = 1, \ldots, n$ may be determined by taking partial derivatives of the error variance w.r.t. the weights

$$\frac{\partial[\quad]}{\partial \lambda_i} = \sum_{j=1}^{n} \lambda_j C(\mathbf{u}_i, \mathbf{u}_i) - 2 \cdot C(\mathbf{u}, \mathbf{u}_i), i = 1, \ldots, n$$

and setting them to zero

$$\sum_{j=1}^{n} \lambda_j C(\mathbf{u}_i, \mathbf{u}_i) = \cdot C(\mathbf{u}, \mathbf{u}_i), i = 1, \ldots, n$$

- This system of *n* equations with *n* unknown weights is the simple kriging (SK) system

- Estimation approach that relies on linear weights that account for spatial continuity, data closeness and redundancy.
- Weights are unbiased and minimize the estimation variance.

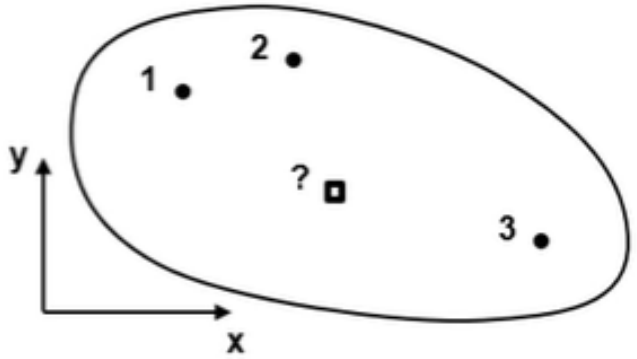
There are three equations to determine the three weights:

$$\lambda_1 \cdot C(\mathbf{u}_1, \mathbf{u}_1) + \lambda_2 \cdot C(\mathbf{u}_1, \mathbf{u}_2) + \lambda_3 \cdot C(\mathbf{u}_1, \mathbf{u}_3) = C(\mathbf{u}_0, \mathbf{u}_1)$$
$$\lambda_1 \cdot C(\mathbf{u}_2, \mathbf{u}_1) + \lambda_2 \cdot C(\mathbf{u}_2, \mathbf{u}_2) + \lambda_3 \cdot C(\mathbf{u}_2, \mathbf{u}_3) = C(\mathbf{u}_0, \mathbf{u}_2)$$
$$\lambda_1 \cdot C(\mathbf{u}_3, \mathbf{u}_1) + \lambda_2 \cdot C(\mathbf{u}_1, \mathbf{u}_2) + \lambda_3 \cdot C(\mathbf{u}_1, \mathbf{u}_3) = C(\mathbf{u}_0, \mathbf{u}_1)$$

In matrix notation: Recall that $C(\mathbf{h}) = C(0) - \gamma(\mathbf{h})$

$$\begin{bmatrix} C(\mathbf{u}_1, \mathbf{u}_1) & C(\mathbf{u}_1, \mathbf{u}_2) & C(\mathbf{u}_1, \mathbf{u}_3) \\ C(\mathbf{u}_2, \mathbf{u}_1) & C(\mathbf{u}_2, \mathbf{u}_2) & C(\mathbf{u}_2, \mathbf{u}_3) \\ C(\mathbf{u}_3, \mathbf{u}_1) & C(\mathbf{u}_3, \mathbf{u}_2) & C(\mathbf{u}_3, \mathbf{u}_3) \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix} = \begin{bmatrix} C(\mathbf{u}_o, \mathbf{u}_1) \\ C(\mathbf{u}_o, \mathbf{u}_2) \\ C(\mathbf{u}_o, \mathbf{u}_3) \end{bmatrix}$$

redundancy **closeness**



- Solution exists and is unique of matrix $\left[C(v_i, v_j)\right]$ is positive definite
- Kriging estimator is unbiased: $E\left\{\left[Z - Z^*\right]\right\} = 0$
- Minimum error variance estimator (just try to pick weights, you won't bet it)
- Best Linear Unbiased Estimator
- Provides a measure of the estimation (or kriging) variance (uncertainty in the estimate):

$$\sigma_E^2(\mathbf{u}) = C(0) - \sum_{\alpha=1}^{n} \lambda_\alpha C(\mathbf{u} - \mathbf{u}_\alpha) \qquad \boldsymbol{\sigma_E^2} \rightarrow [\mathbf{0}, \boldsymbol{\sigma_x^2}]$$

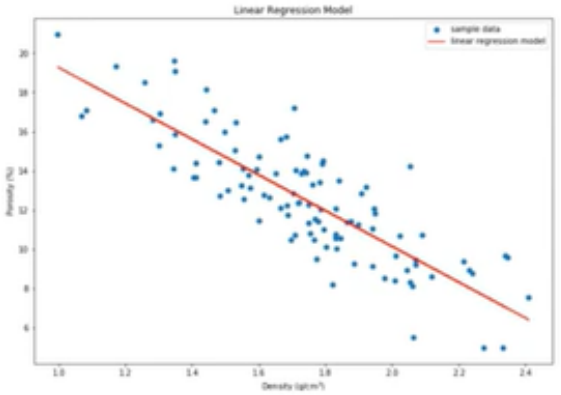
- Exact interpolator: at data location
- Kriging variance can be calculated before getting the sample information, homoscedastic!
- Kriging takes into account:
  - distance of the information: $C(\mathbf{u},\mathbf{u}_i)$
  - configuration of the data: $C(\mathbf{u}_i,\mathbf{u}_j)$
  - structural continuity of the variable being considered: $C(\mathbf{h})$
- The smoothing effect of kriging can be forecast – we will return to this with simulation.
- Kriging theory is part of the probabilistic theory of *projectors*: orthogonal projection onto space of linear combinations of the *n* data (Hilbert space)

Isotonic Regression

Recall: linear regression fits the function

$$y = \sum_{\alpha=1}^{m} b_\alpha x_\alpha + b_0$$

- $x_1, \ldots, x_m$ predictor features
- y response features
- $b_0, b_1, \ldots, b_m$ parameters



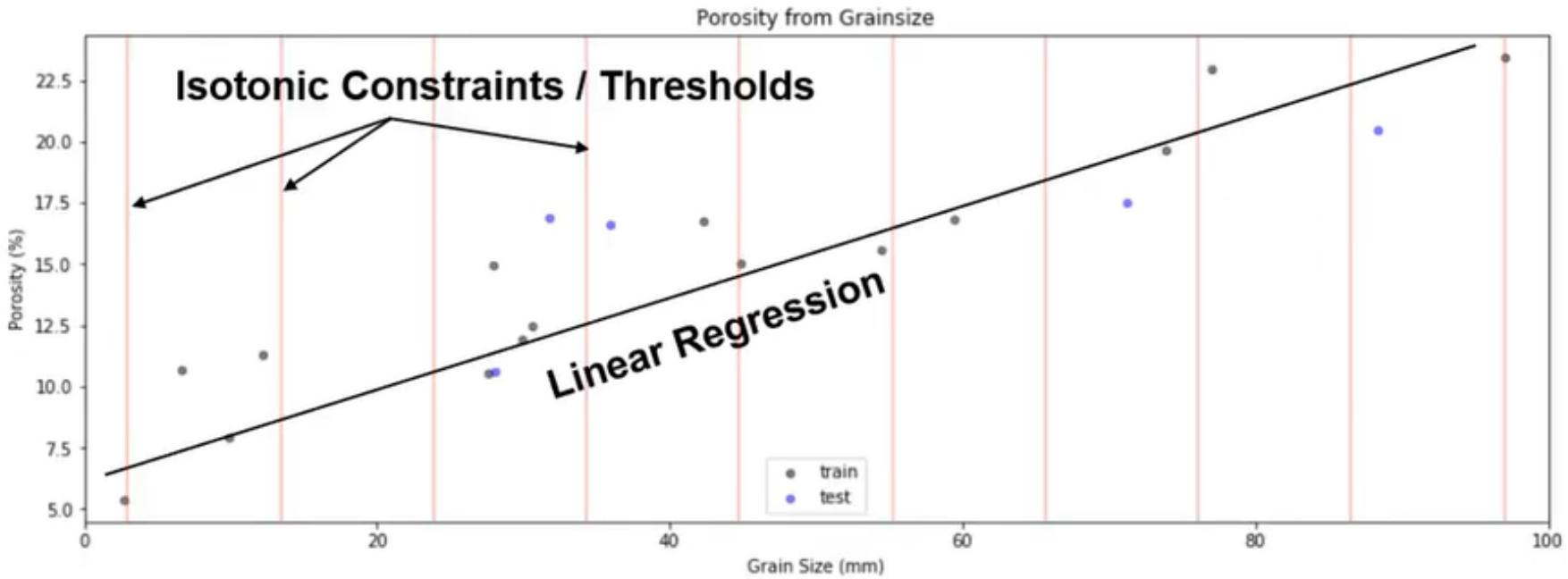
Under the constraint:

$$RSS = \sum_{i=1}^{n} \left(y_i - \sum_{\alpha=1}^{m} b_\alpha x_\alpha + b_0 \right)^2$$

minimize residual sum of squares (RSS) over the training data.

General Comments:

- A single linear model is quite inflexible
- Could we break up the problem into many linear model segments?



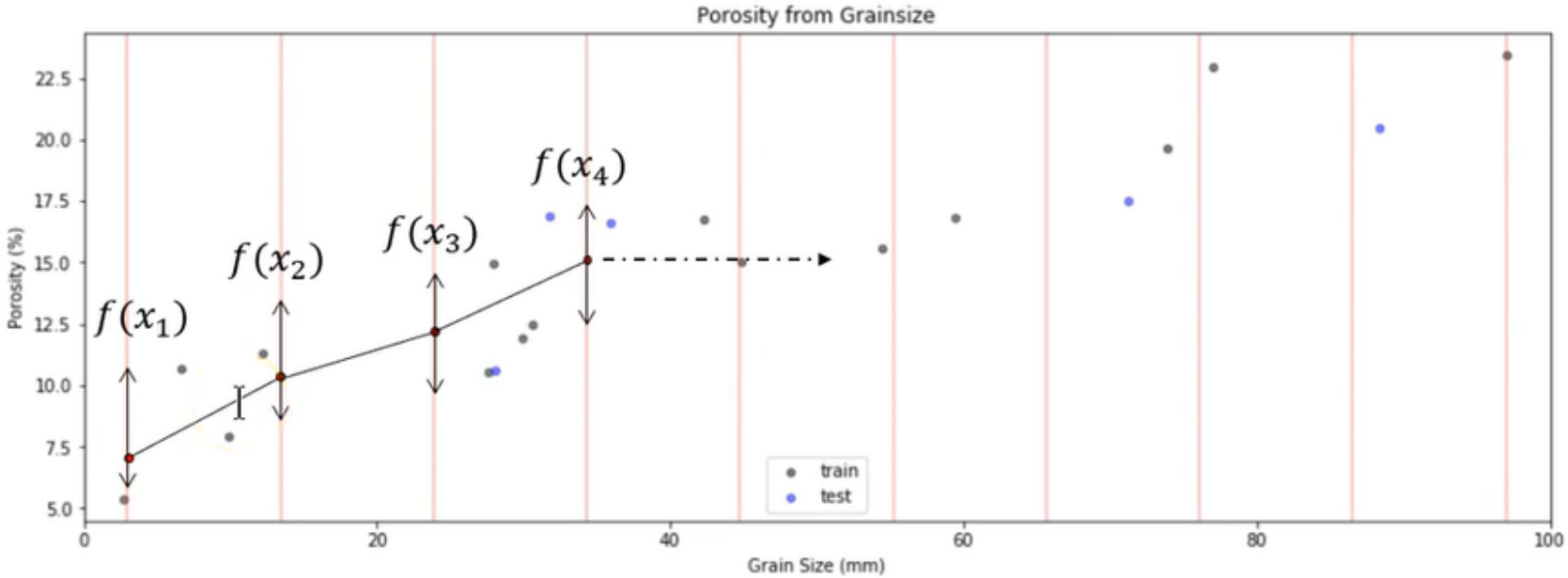
- We provide a set of thresholds in the predictor feature, $x_1, x_2, \ldots x_k$

Isotonic

- Piece-wise linear model
- Monotonic constraint

General Comments:

- The model is now parameterized by a set of predictions at the thresholds, $f(x_1), f(x_2), \ldots f(x_K), k = 1, \ldots, K$ thresholds



- The estimates between the thresholds are linear interpolations

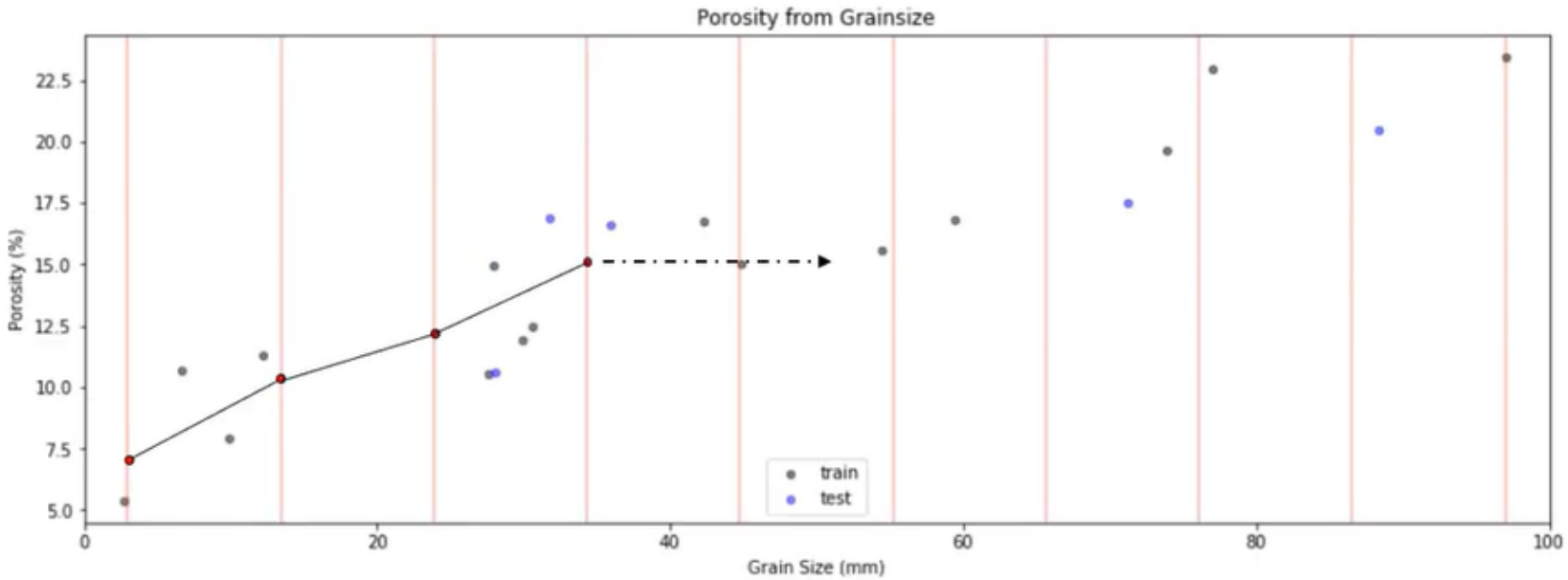
$$f(x_0) = f(x_{k-1}) + (x_0 - x_{k-1})\frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}}, \text{ where } x_{k-1} \leq x_0 \leq x_k$$

General Comments:

- We are also able to impose a monotonic constraint on the solution.

$$f(x_1) \leq f(x_2) \leq \cdots \leq f(x_k)$$

- The solution slope is always nonnegative



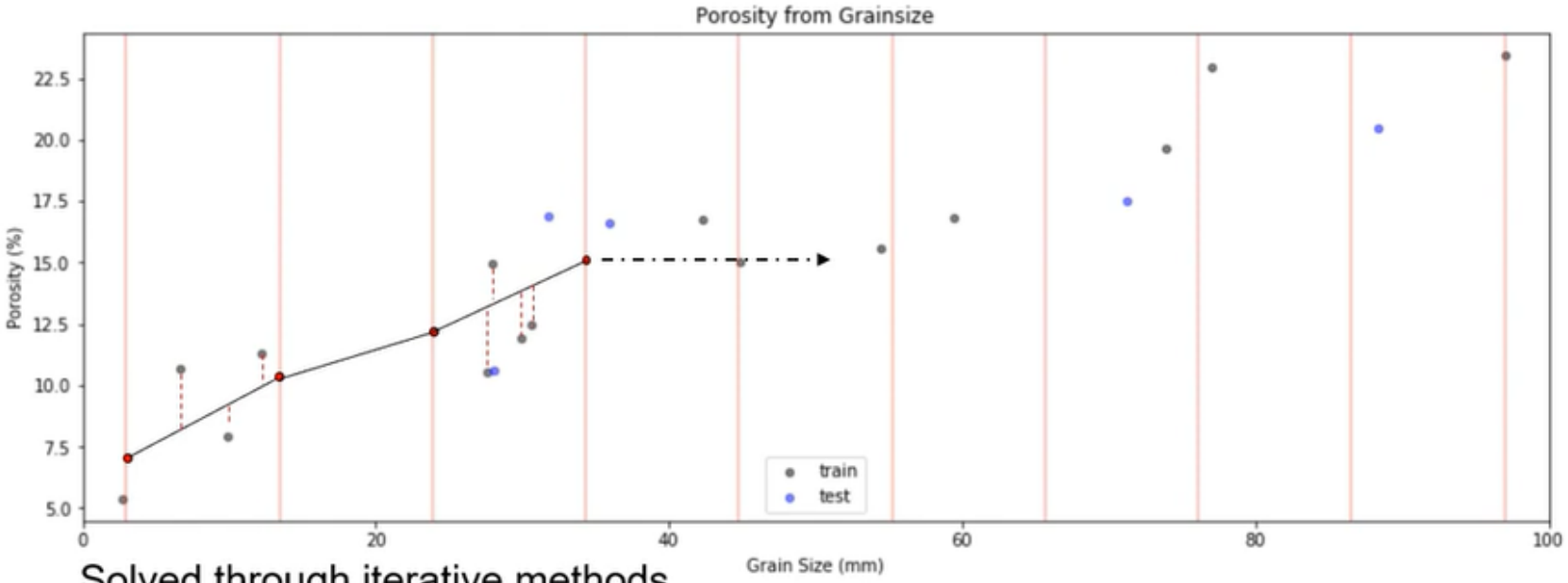
General Comments:

- We have the following loss function based on our piece-wise function.

$$min\left(\sum_{i=1}^{n}(y_i - \hat{y}_i)^2\right)$$

Under constraint:

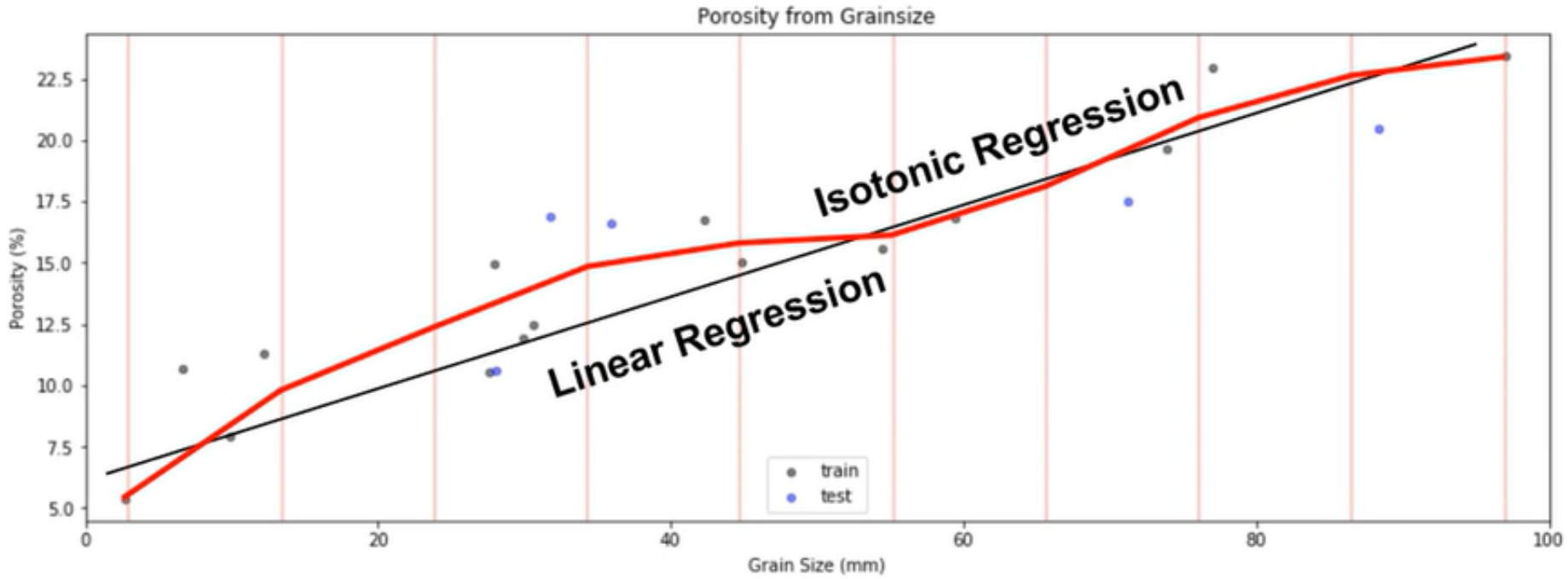
$$f(x_1) \leq f(x_2) \leq \cdots \leq f(x_k)$$



Solved through iterative methods.

The Isotonic Regression Model:

- The result is quite a flexible model.



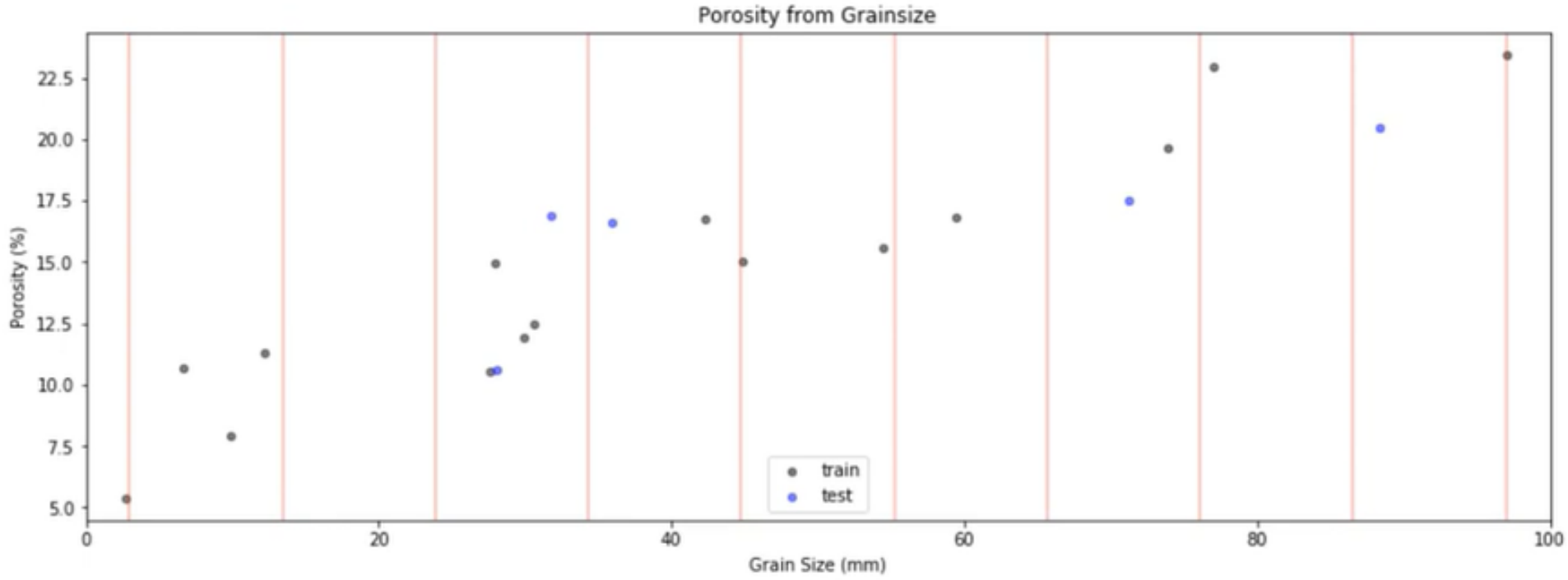
The Isotonic Regression Issue:

- As the number of isotonic constraints, K, and the number of predictor features increases, m, the number of parameters to train:

$$p = K^m$$

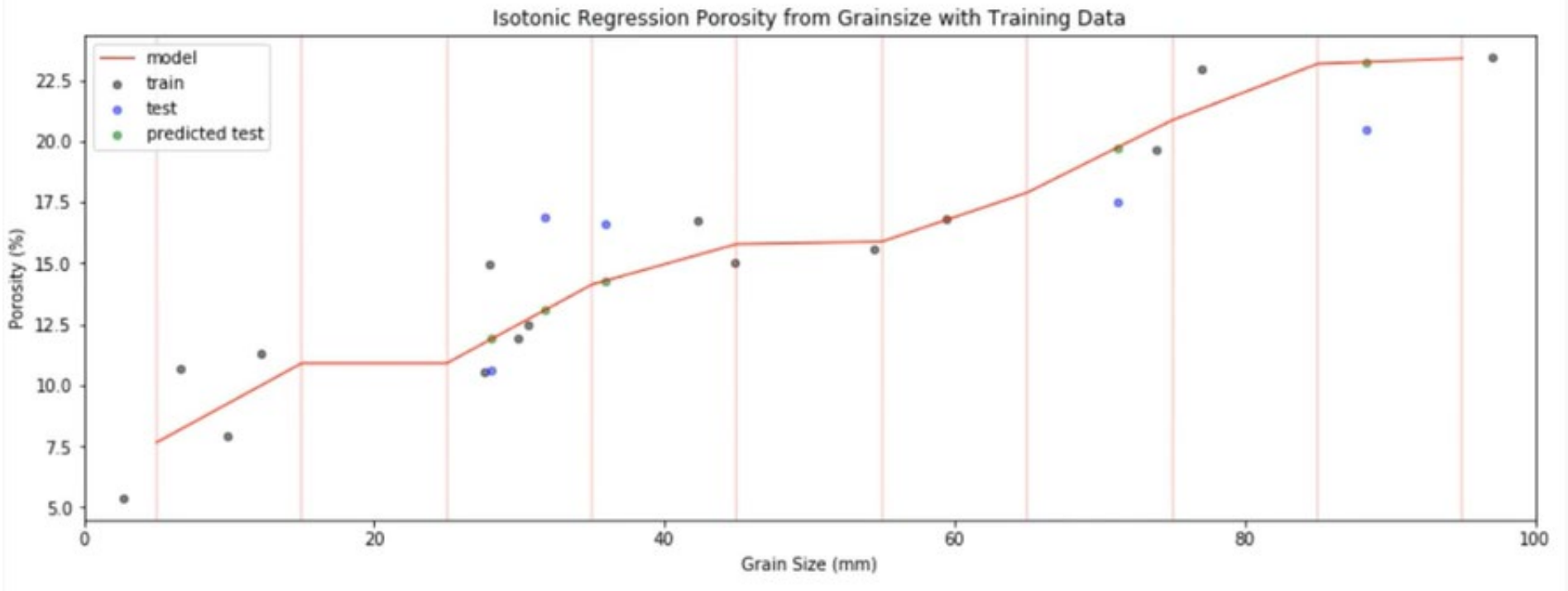
- Strong risk of overfit for large, K, solution is to use a small K value

- Test and train with porosity and grainsize data
- Monotonic increase is reasonable and may be nonlinear
- Assume $K = 10$



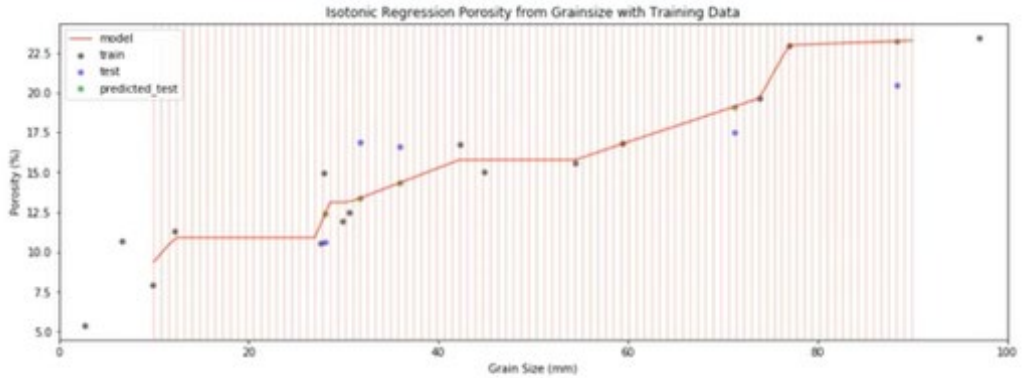
The Isotonic Regression Model:

- The result is quite a flexible model.

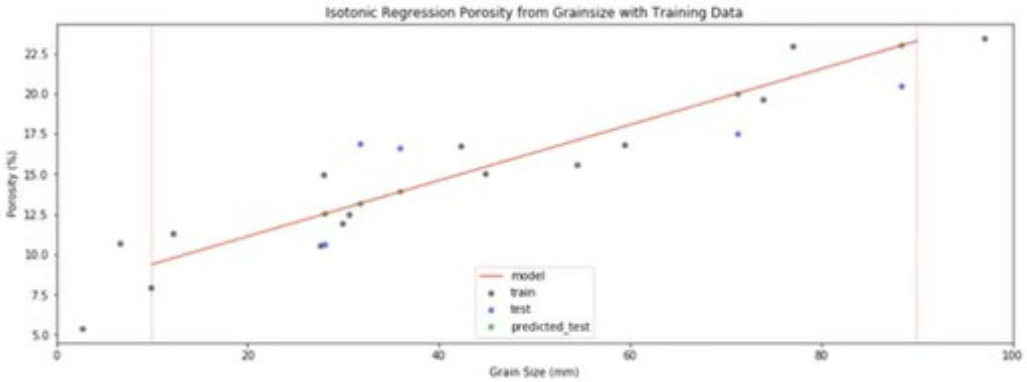


The Hyperparameter, K

- Determines the degree of fit, overfit vs. underfit.



Variance explained
40%



Variance explained
27%

Applied Probability and Statistics

Module-1, Lecture-17

By: Dr. A K Yadav (9911375598)
Dept of CSE, ASET, AUUP

Least absolute shrinkage and selection operator

- Least absolute shrinkage and selection operator (LASSO) regression is a form of supervised statistical learning (i.e., machine learning) aimed at improving prediction
- LASSO regression effectively selects only the most important predictor variables for predicting an outcome by shrinking the regression coefficients associated with the least important predictor variables to zero

- General reasons for applying LASSO regression:
  - ✓To avoid overfitting a model to the data on which it was estimated (i.e., trained), which can be problematic with conventional regression techniques (e.g., linear, logistic), especially when there is a large number of predictor variables
  - ✓To select the most important predictor variables (i.e., features) from a much larger number of predictor variables

- LASSO regression is a regularization method (specifically an L1 regularization method) and is related to other regularization methods like ridge regression and elastic net
- The purpose of regularization is to reduce variance of parameter estimates (i.e., regression coefficients), even if it comes at the expense of some additional bias; ultimately, this means finding the optimal level of model complexity

- Technically, LASSO regression involves two tuning parameters called alpha and lambda, but because alpha=1 for LASSO regression.
- We will focus on lambda because it can be varied during model training
- The lambda tuning parameter places a constraint on the maximum absolute value of the regression coefficients in the model and adds a penalty to non-zero regression coefficients

- Effects of different lambda values:
  - ✓When lambda is zero, the results will approximate a conventional (e.g., ordinary least squares [OLS] regression) model, and no regression coefficients associated with predictor variables shrink to zero {i.e., be eliminated)
  - ✓When lambda is large, regression coefficients with smaller absolute values shrink toward zero
  - ✓When lambda becomes too large, all regression coefficients shrink to zero

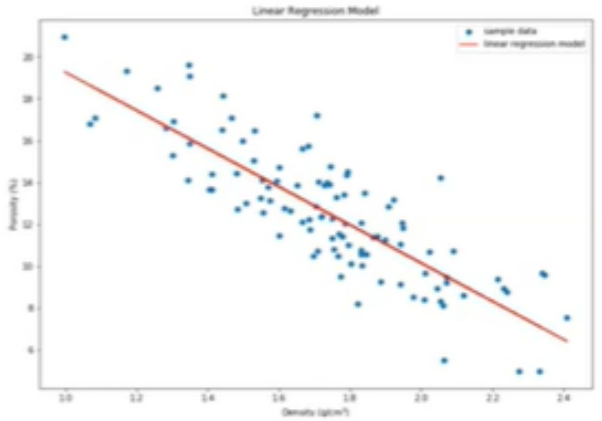
- ✓ As lambda gets smaller, variance grows larger
- ✓ As lambda gets larger, bias grows larger

- Traditionally, LASSO regression has been applied to linear regression models
- LASSO regression can, however, also be applied to other families of models, such as generalized linear models which include logistic regression models

Recall: linear regression fits the function

$$y = \sum_{\alpha=1}^{m} b_\alpha x_\alpha + b_0$$

- $x_1, \ldots, x_m$ predictor features
- y response features
- $b_0, b_1, \ldots, b_m$ parameters



Under the constraint:

$$RSS = \sum_{i=1}^{n} \left(y_i - \sum_{\alpha=1}^{m} b_\alpha x_\alpha + b_0 \right)^2$$

minimize residual sum of squares (RSS) over the training data.

With ridge regression we add a new term to the optimization:

$$min\left(\underbrace{\sum_{i=1}^{n}\left(y_i - \sum_{\alpha=1}^{m} b_\alpha x_\alpha + b_0\right)^2}_{\text{RSS}} + \underbrace{\lambda \sum_{\alpha=1}^{m} b_\alpha{}^2}_{\text{Shrinkage Penalty}}\right)$$

Now we have the standard residual sum squares and the shrinkage penalty

This is known as a shrinkage method

With lasso regression we add a new term to the optimization:

$$min\left(\sum_{i=1}^{n}\left(y_i - \sum_{\alpha=1}^{m} b_\alpha x_\alpha + b_0\right)^2 + \lambda \sum_{\alpha=1}^{m} |b_\alpha|\right)$$

RSS + Shrinkage Penalty

Lasso Cost / Loss function

Compare this with ridge regression.

$$min\left(\sum_{i=1}^{n}\left(y_i - \sum_{\alpha=1}^{m} b_\alpha x_\alpha + b_0\right)^2 + \lambda \sum_{\alpha=1}^{m} b_\alpha{}^2\right)$$

RSS + Shrinkage Penalty

Ridge Cost / Loss function

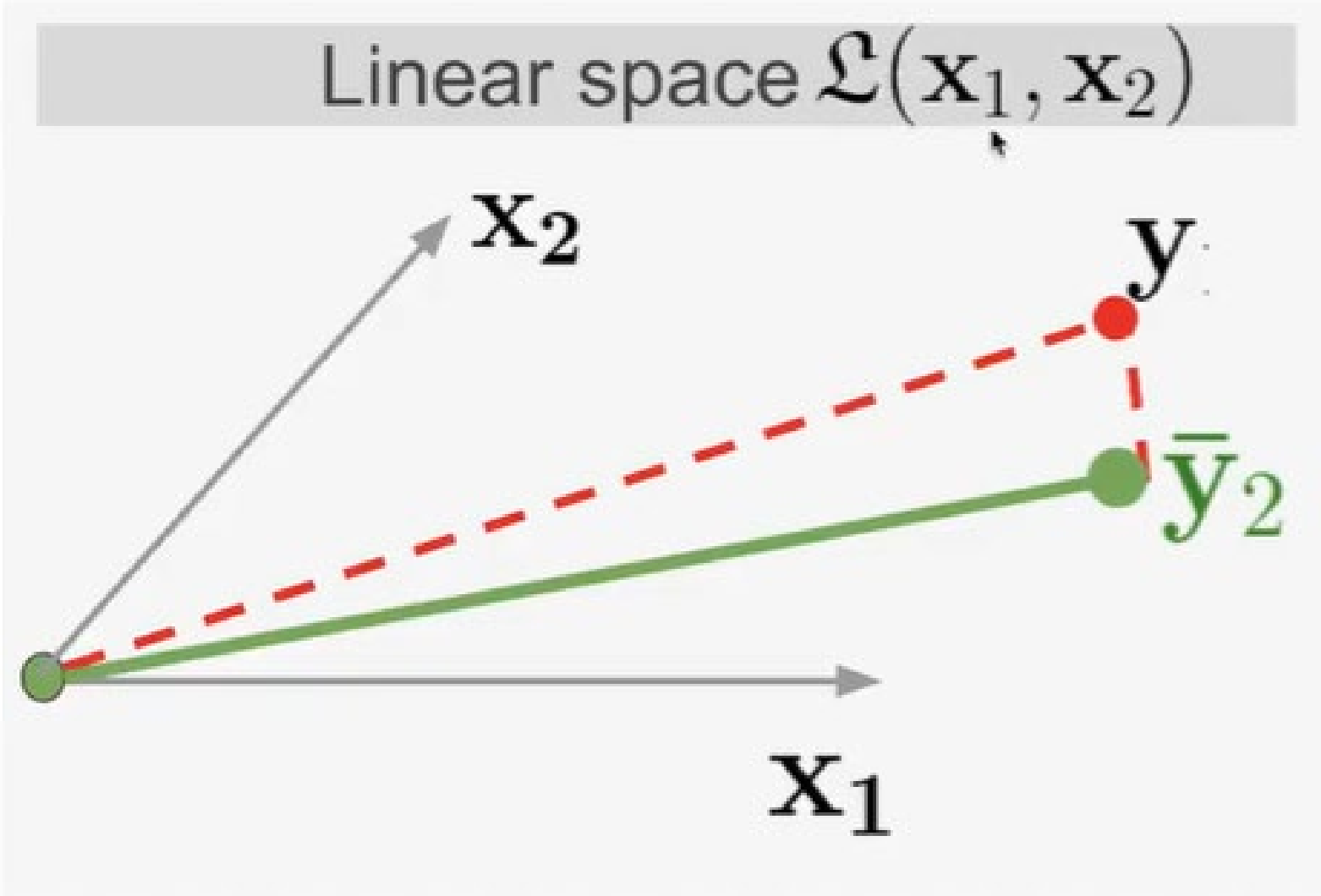
The difference is a L^1 vs L^2 norm for the shrinkage penalty term.

Applied Probability and Statistics

Module-1, Lecture-18

By: Dr. A K Yadav (9911375598)
Dept of CSE, ASET, AUUP

Least Angle Regression (LARS)



- Where $x_1, x_2, \ldots, x_k, \ldots, x_p$ are the p feature or predictor or covariates
- $\hat{\mu}_0, \hat{\mu}_1, \ldots, \hat{\mu}_p$ are the prediction vector
- $\bar{y}_1, \bar{y}_2, \ldots, \bar{y}_p$ are projection
- u_2 is unit vector along bisector
- $y = \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p$
- $\langle x_j, r \rangle$ is a dot product between x_j and residual r

LARS Algorithms

1. Standardize features x_j to $mean = 0$ and $variance = 1$
2. Get residual $r = y - \bar{y}$ where coefficients $\beta_1, \beta_2, \ldots, \beta_p = 0$
3. Find a feature x_j "most correlated" with residual r
4. Incrementally update coefficients β_j from zero to least squares coefficient $\langle x_j, r \rangle$ until another feature x_k has the same correlation as x_j with residual r.
5. Now, move (β_j, β_k) in the direction defined by their joint least square coefficients of current residual on $(x_j x_k)$, until some another competitor x_l has same correlation with the residual r.
6. Repeat for all p features
7. After p steps, we arrive at full least squares solution

covariates $\mathbf{x_1}$ $\mathbf{x_2}$

prediction vector $\hat{\mu}_0$ $\hat{\mu}_1$

projection $\bar{\mathbf{y}}_1$ $\bar{\mathbf{y}}_2$

unit vector along bisector $\mathbf{u_2}$

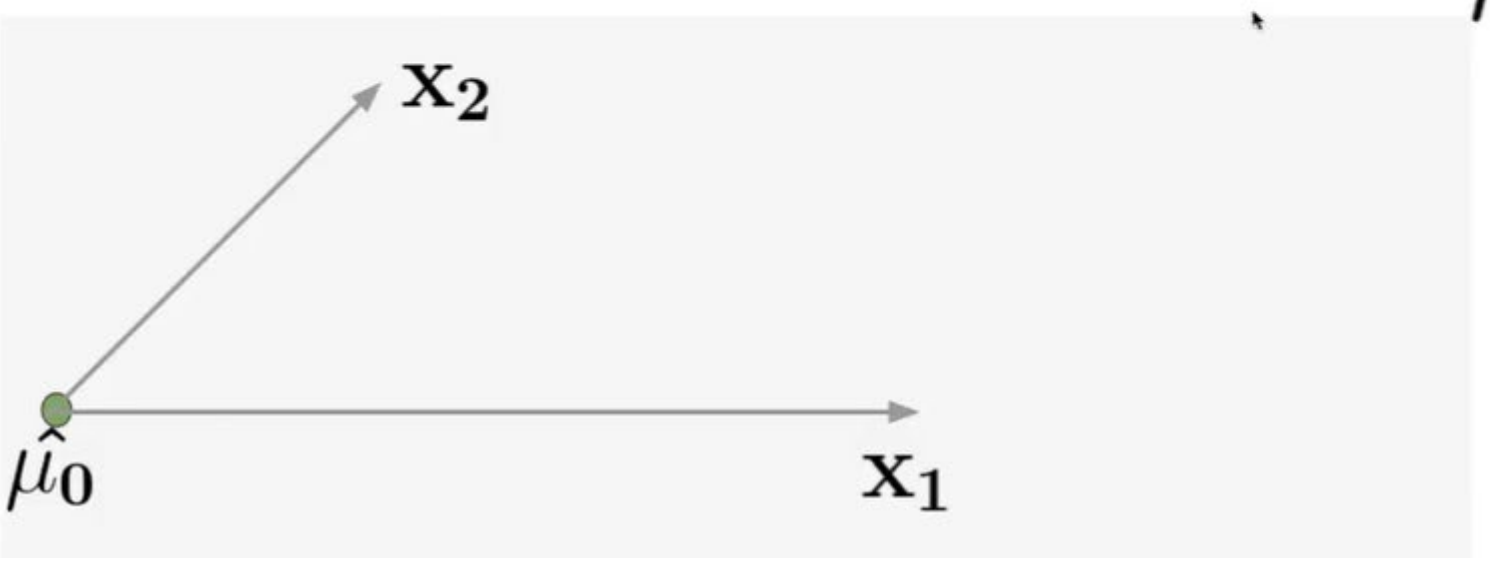
Begin …

Standardize $\mathbf{x_1}, \mathbf{x_2}$ to mean of zero.

$\hat{\mu}_0$ is prediction based solely on an intercept.

LARS estimate:

$$\hat{\mu}_0 = 0$$



covariates $\mathbf{x_1}$ $\mathbf{x_2}$

prediction vector $\hat{\mu}_0$ $\hat{\mu}_1$

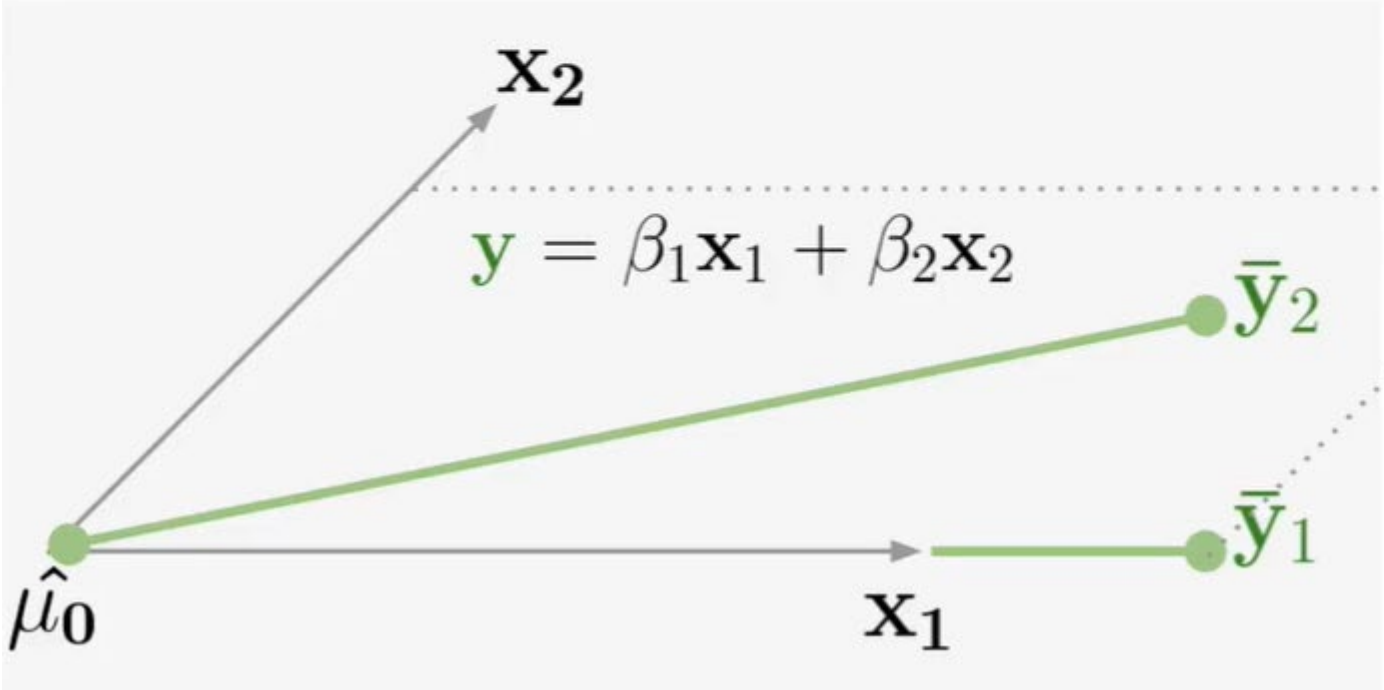
projection $\bar{\mathbf{y}}_1$ $\bar{\mathbf{y}}_2$

unit vector along bisector $\mathbf{u_2}$

Pick a variable "most correlated" to $\mathbf{r} \Rightarrow \mathbf{x_1}$

$\bar{\mathbf{y}}_2 - \hat{\mu}_0$ making a smaller angle with $\mathbf{x_1}$ than $\mathbf{x_2}$

$\bar{\mathbf{y}}_2 - \hat{\mu}_0$ has greater correlation with $\mathbf{x_1}$ than $\mathbf{x_2}$



LARS estimate:

$$\hat{\mu}_0 = 0$$

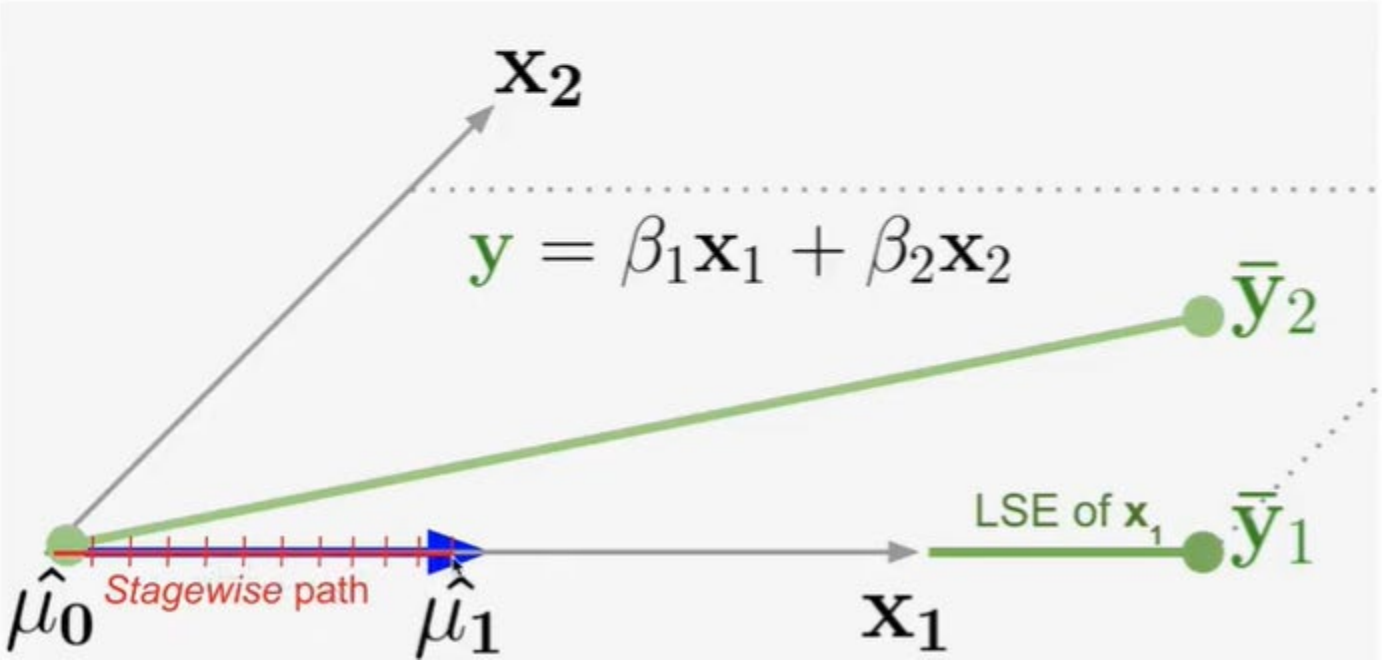
covariates $\mathbf{x_1}$ $\mathbf{x_2}$

prediction vector $\hat{\mu_0}$ $\hat{\mu_1}$

projection $\bar{\mathbf{y}}_1$ $\bar{\mathbf{y}}_2$

unit vector along bisector $\mathbf{u_2}$

Augment $\hat{\mu_0}$ in the direction of $\mathbf{x_1}$ to $\hat{\mu_1} = \hat{\mu_0} + \hat{\gamma}_1 \mathbf{x_1}$



LARS estimate:

$$\hat{\mu_0} = 0$$

$$\hat{\mu_1} = \hat{\mu_0} + \hat{\gamma}_1 \mathbf{x_1}$$

covariates $\mathbf{x_1}$ $\mathbf{x_2}$

prediction vector $\hat{\mu}_0$ $\hat{\mu}_1$

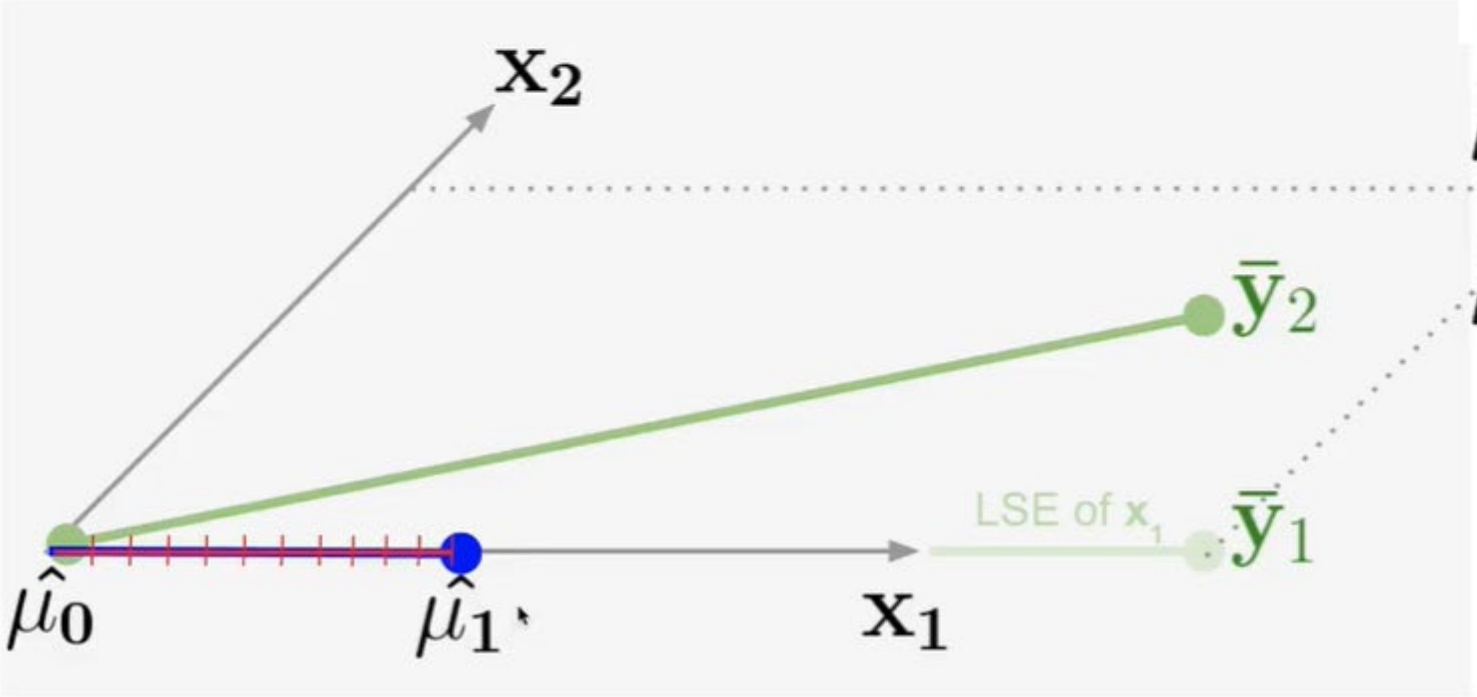
projection $\bar{\mathbf{y}}_1$ $\bar{\mathbf{y}}_2$

unit vector along bisector $\mathbf{u_2}$

$\bar{\mathbf{y}}_2 - \hat{\mu}$ is equally correlated with $\mathbf{x_1}$ and $\mathbf{x_2}$

Proceed to $\bar{\mathbf{y}}_2$

$$\mathbf{y} = \beta_1 \mathbf{x}_1 + \beta_2 \mathbf{x}_2$$

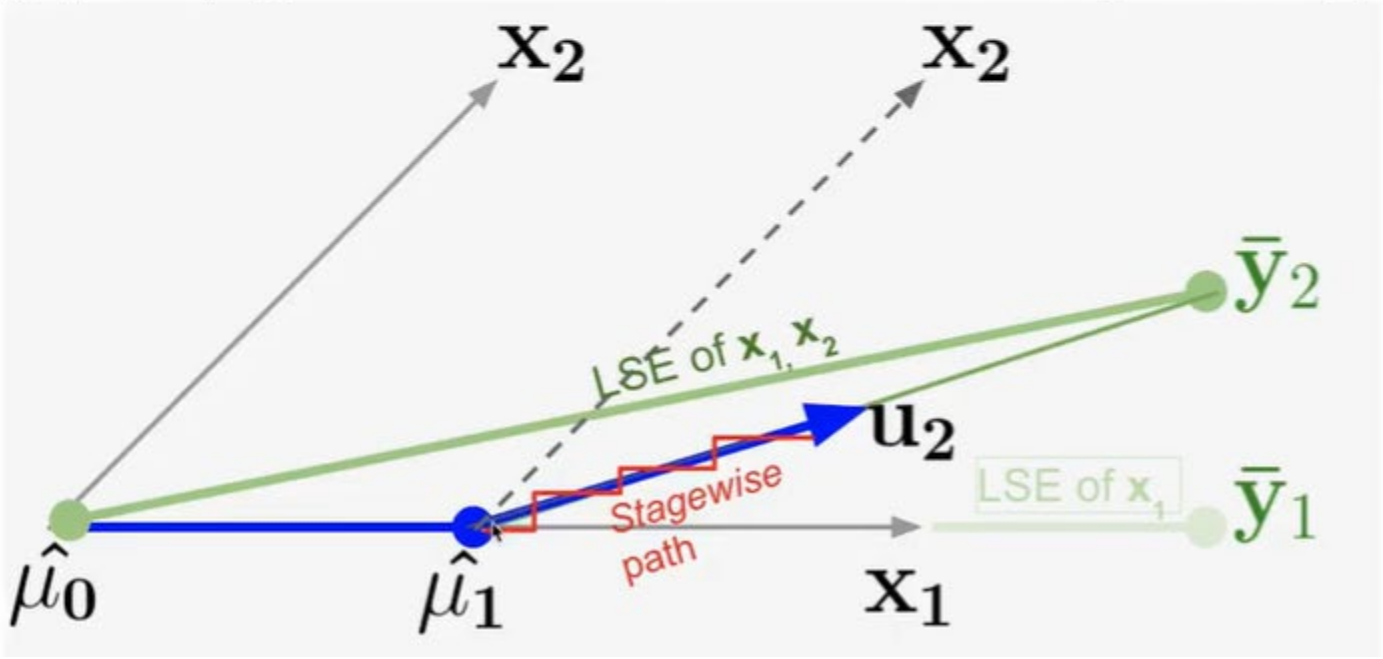


$$\hat{\mu}_0 = 0$$

$$\hat{\mu}_1 = \hat{\mu}_0 + \hat{\gamma}_1 \mathbf{x}_1$$

Choose intermediate value of $\hat{\gamma}_1$ that makes $\bar{\mathbf{y}}_2 - \hat{\mu}_1$ equally correlated with $\mathbf{x}_1$ and $\mathbf{x}_2$

$\bar{\mathbf{y}}_2 - \hat{\mu}_1$ bisects the angle between $\mathbf{x}_1$ and $\mathbf{x}_2$



covariates $\mathbf{x}_1$ $\mathbf{x}_2$

prediction vector $\hat{\mu}_0$ $\hat{\mu}_1$

projection $\bar{\mathbf{y}}_1$ $\bar{\mathbf{y}}_2$

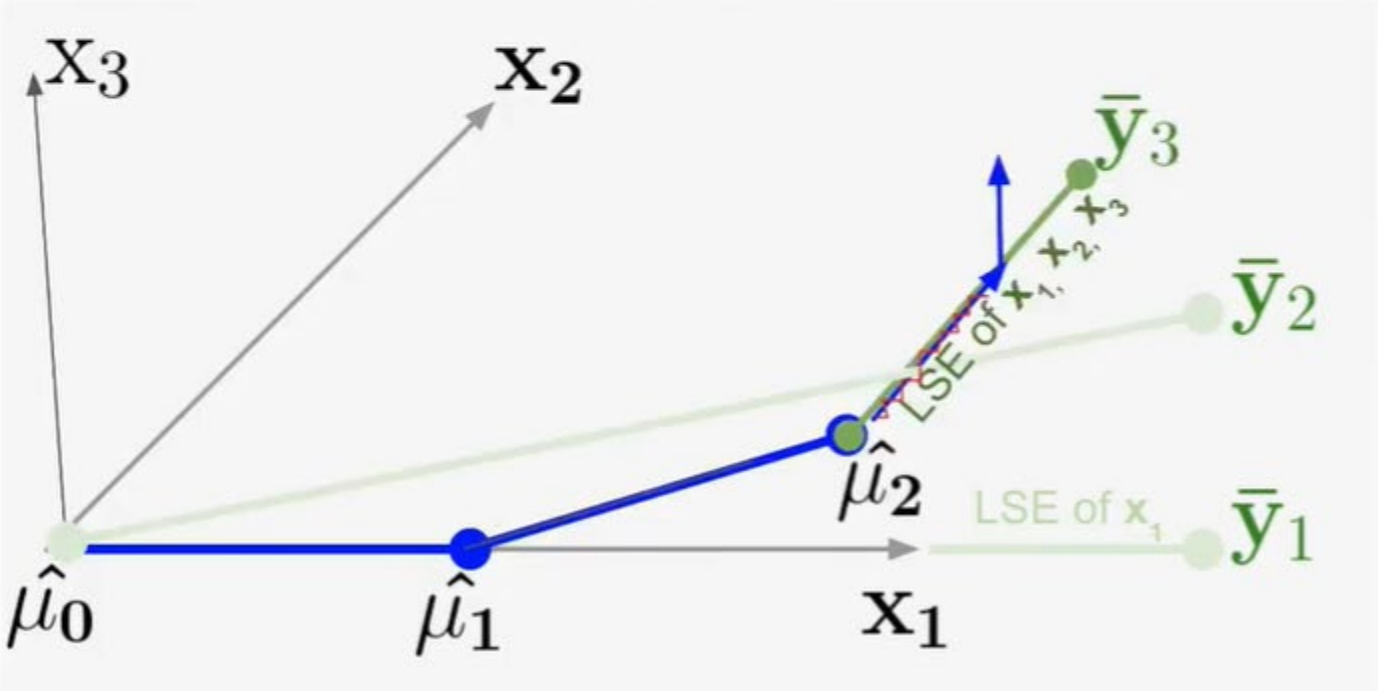
unit vector along bisector $\mathbf{u}_2$

LARS estimate:

$$\hat{\mu}_0 = 0$$

$$\hat{\mu}_1 = \hat{\mu}_0 + \hat{\gamma}_1 \mathbf{x}_1$$

Next, LARS estimate is $\hat{\mu}_2 = \hat{\mu}_1 + \hat{\gamma}_2 \mathbf{u}_2$
with $\hat{\gamma}_2$ chosen to make $\hat{\mu}_2 = \bar{\mathbf{y}}_2$



covariates $\mathbf{x}_1$ $\mathbf{x}_2$
prediction vector $\hat{\mu}_0$ $\hat{\mu}_1$
projection $\bar{\mathbf{y}}_1$ $\bar{\mathbf{y}}_2$
unit vector along bisector $\mathbf{u}_2$

LARS estimate:

$$\hat{\mu}_0 = 0$$

$$\hat{\mu}_1 = \hat{\mu}_0 + \hat{\gamma}_1 \mathbf{x}_1$$

$$\hat{\mu}_2 = \hat{\mu}_1 + \hat{\gamma}_2 \mathbf{u}_2$$