Statistical Optimization: Lecture 9

Projected GD, Coordinate descent, Proximal GD and SGD

Zijian Guo

Zhejiang University
Center for Data Science

April 6, 2026

Objective

In this lecture, we briefly introduce four important extensions of standard gradient descent:

- projected gradient descent,
- coordinate descent,
- proximal gradient descent,
- stochastic gradient descent.

Our goal is to understand the basic idea of each method, what kind of problem it is designed for, and how it differs from standard gradient descent. We will also use Lasso as an illustrative example.

Outline

LASSO regression

Projected Gradient Descent

Coordinate Descent

Proximal Gradient Descent

Stochastic Gradient Descent

LASSO setup

LASSO (**L**east **A**bsolute **S**hrinkage and **S**election **O**perator) is a linear regression model with an ℓ_1 regularization term.

LASSO solves

$$\min_{\theta \in \mathbb{R}^d} \frac{1}{2n} \|y - X\theta\|_2^2 + \lambda \|\theta\|_1, \qquad \|\theta\|_1 = \sum_{j=1}^{d} |\theta_j|, \qquad \text{(penalized form)}$$

- $X \in \mathbb{R}^{n \times d}$: design matrix, $y \in \mathbb{R}^n$: response vector,
- $\theta \in \mathbb{R}^d$: regression coefficient, $\lambda > 0$: regularization parameter.

Compared with ordinary least squares, the ℓ_1 penalty shrinks coefficients toward zero and can make some of them exactly zero. Therefore, LASSO is useful when we want a sparse and interpretable model.

Equivalent forms of LASSO

LASSO can be written in another equivalent form for suitable choices of τ.

$$\min_{\theta \in \mathbb{R}^d} \frac{1}{2n} \|y - X\theta\|_2^2 \quad \text{s.t.} \quad \|\theta\|_1 \leq \tau. \qquad \text{(constrained form)}$$

Methods used:

- Coordinate descent for the penalized form,
- Proximal gradient descent for the penalized form,
- Stochastic proximal gradient descent for the penalized form,
- Projected gradient descent for the constrained form.

Outline

LASSO regression

Projected Gradient Descent

Coordinate Descent

Proximal Gradient Descent

Stochastic Gradient Descent

Projected Gradient Descent (PGD)

- We consider constrained optimization:

$$\min_{\theta \in \Theta} f(\theta), \qquad \Theta \subseteq \mathbb{R}^d \text{ convex.}$$

- **Projected Gradient Descent (PGD):**

$$\eta_{t+1} = \theta_t - \gamma \nabla f(\theta_t), \qquad \theta_{t+1} = \Pi_\Theta(\eta_{t+1}) = \arg\min_{\theta \in \Theta} \|\theta - \eta_{t+1}\|^2.$$

Remark.

- The method first takes a usual gradient step to get η_{t+1}, then project it back onto Θ to obtain θ_{t+1}.
- When Θ is simple (for example, a ball or a box), projection is easy to compute. If Θ is not simple, the projection step itself may be difficult or expensive, so PGD may no longer be efficient in practice.

Projection onto a convex set

For any $\eta \in \mathbb{R}^d$, the projection onto Θ is defined by

$$\Pi_\Theta(\eta) := \arg\min_{\theta \in \Theta} \|\theta - \eta\|^2.$$

In words, $\Pi_\Theta(\eta)$ is the point in Θ closest to η.

If $\Theta \subseteq \mathbb{R}^d$ is nonempty, closed, and convex, then the projection

$$\Pi_\Theta(\eta) = \arg\min_{\theta \in \Theta} \|\theta - \eta\|^2$$

exists and is unique for every $\eta \in \mathbb{R}^d$.

A simple fact about projection

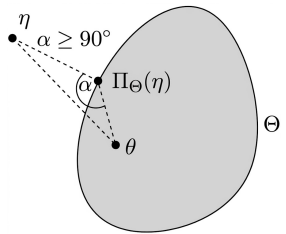
Fact. Let $\Theta \subseteq \mathbb{R}^d$ be closed and convex, $\theta \in \Theta$, and $\eta \in \mathbb{R}^d$. Then

(i)

$$(\theta - \Pi_\Theta(\eta))^\top (\eta - \Pi_\Theta(\eta)) \leq 0.$$

(ii)

$$\|\theta - \Pi_\Theta(\eta)\|^2 + \|\eta - \Pi_\Theta(\eta)\|^2 \leq \|\theta - \eta\|^2.$$



Proof of the projection facts

Consider the constrained problem

$$\Pi_\Theta(\eta) = \arg\min_{u \in \Theta} \|u - \eta\|^2.$$

Let

$$f(u) = \|u - \eta\|^2.$$

Since Θ is convex, the first-order optimality condition gives

$$\nabla f(\Pi_\Theta(\eta))^\top (\theta - \Pi_\Theta(\eta)) \geq 0, \qquad \forall \theta \in \Theta.$$

Because

$$\nabla f(u) = 2(u - \eta),$$

we obtain

$$(\Pi_\Theta(\eta) - \eta)^\top (\theta - \Pi_\Theta(\eta)) \geq 0, \qquad \forall \theta \in \Theta.$$

Equivalently,

$$(\theta - \Pi_\Theta(\eta))^\top (\eta - \Pi_\Theta(\eta)) \leq 0.$$

Proof of the projection facts

Moreover,

$$\|\theta - \eta\|^2 = \|\theta - \Pi_\Theta(\eta)\|^2 + \|\Pi_\Theta(\eta) - \eta\|^2 + 2(\theta - \Pi_\Theta(\eta))^\top (\Pi_\Theta(\eta) - \eta).$$

Using

$$(\theta - \Pi_\Theta(\eta))^\top (\Pi_\Theta(\eta) - \eta) \geq 0,$$

we get

$$\|\theta - \Pi_\Theta(\eta)\|^2 + \|\eta - \Pi_\Theta(\eta)\|^2 \leq \|\theta - \eta\|^2.$$

Projected gradient descent algorithm

Algorithm: Projected Gradient Descent

Input: $\theta_0 \in \Theta,\ \gamma > 0,\ tol > 0$
for $t = 0, 1, 2, \ldots, T-1$
$\quad \eta_{t+1} := \theta_t - \gamma \nabla f(\theta_t),$
$\quad \theta_{t+1} := \Pi_\Theta(\eta_{t+1}),$
$\quad$ if $\|\theta_{t+1} - \theta_t\|_2 \leq tol,$ stop
end

- The intermediate point η_{t+1} may lie outside Θ.
- The projection step brings it back to the feasible set.
- In implementation, the stopping rule is based on the parameter change $\|\theta_{t+1} - \theta_t\|_2$.

Projected gradient descent for Lasso

$$\min_{\theta \in \mathbb{R}^d} \frac{1}{2n} \|y - X\theta\|_2^2 \qquad \text{s.t.} \quad \|\theta\|_1 \le \tau.$$

Here

$$f(\theta) = \frac{1}{2n} \|y - X\theta\|_2^2, \qquad \Theta = \{\theta \in \mathbb{R}^d : \|\theta\|_1 \le \tau\}.$$

Its gradient is

$$\nabla f(\theta_t) = \frac{1}{n} X^\top (X\theta_t - y).$$

Thus PGD first takes a gradient step

$$\eta_{t+1} = \theta_t - \gamma \nabla f(\theta_t) = \theta_t - \frac{\gamma}{n} X^\top (X\theta_t - y),$$

and then projects back to the feasible set:

$$\theta_{t+1} = \Pi_\Theta(\eta_{t+1}) = \Pi_{\|\theta\|_1 \le \tau}(\eta_{t+1}).$$

How do we compute the projection?

The projection step is

$$\Pi_{\|\theta\|_1 \le \tau}(\eta) = \arg \min_{\|\theta\|_1 \le \tau} \|\theta - \eta\|_2^2.$$

In implementation:

- If $\|\eta\|_1 \le \tau$, then η is already feasible, so

$$\Pi_{\|\theta\|_1 \le \tau}(\eta) = \eta.$$

- Otherwise, the projection has the form (derived by KKT condition)

$$\left[\Pi_{\|\theta\|_1 \le \tau}(\eta)\right]_i = \operatorname{sign}(\eta_i)\ \max\left(|\eta_i| - \lambda, 0\right),$$

  where the threshold $\lambda \ge 0$ is chosen so that

$$\sum_{i=1}^{d} \max\left(|\eta_i| - \lambda, 0\right) = \tau.$$

Projected gradient descent

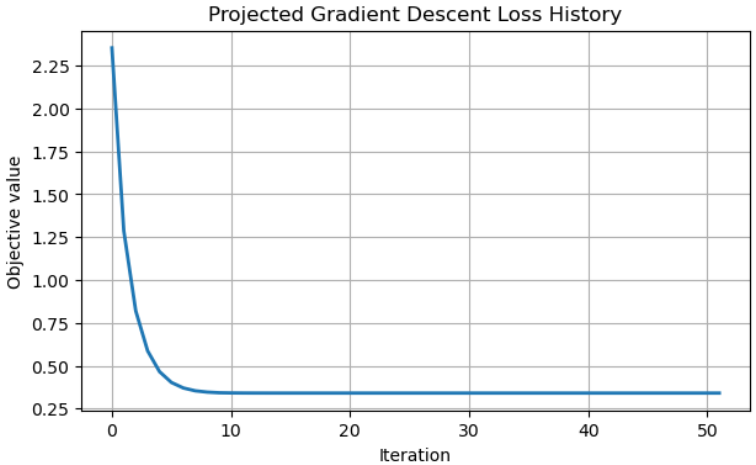


Figure: Projected gradient descent result

Outline

LASSO regression

Projected Gradient Descent

Coordinate Descent

Proximal Gradient Descent

Stochastic Gradient Descent

What is coordinate descent?

$$\min_{\theta \in \mathbb{R}^d} f(\theta).$$

In standard gradient descent, we update all coordinates at each step:

$$\theta_{t+1} = \theta_t - \gamma_t \nabla f(\theta_t).$$

In **coordinate descent**, we update only one coordinate at a time.

If the selected coordinate at iteration t is $i \in [d]$, then the update takes the form

$$\theta_{t+1} = \theta_t - \gamma_t \nabla_i f(\theta_t)\, e_i.$$

Here, $\nabla_i f(\theta_t)$ denotes the partial derivative of f with respect to the i-th coordinate, and e_i is the i-th standard basis vector.

Randomized coordinate descent

The simplest coordinate descent method is to choose the coordinate randomly.

At iteration t:

- sample $i \in [d]$ uniformly at random;
- update only the selected coordinate:

$$\theta_{t+1} = \theta_t - \gamma_t \nabla_i f(\theta_t)\, e_i.$$

This method is called **randomized coordinate descent**.

Coordinate descent for Lasso

Consider the penalized Lasso problem

$$\min_{\theta \in \mathbb{R}^d} \frac{1}{2n} \|y - X\theta\|_2^2 + \lambda \|\theta\|_1.$$

Here

$$\|\theta\|_1 = \sum_{i=1}^{d} |\theta_i|$$

is not differentiable at 0.

However, coordinate descent updates only one coordinate at a time.
Then the optimization problem is no longer d-dimensional. It becomes a minimization problem in one scalar variable only.

Coordinate descent for Lasso

Fix all coordinates except θ_i. Define the partial residual by

$$r^{(i)} = y - \sum_{k \neq i} X_{\cdot k}\theta_k.$$

$$y - X\theta = r^{(i)} - X_{\cdot i}\theta_i.$$

Therefore, when only θ_i is treated as a variable, the Lasso objective becomes

$$\min_{u \in \mathbb{R}} \frac{1}{2n}\|r^{(i)} - X_{\cdot i}u\|_2^2 + \lambda|u|.$$

So the ith coordinate update reduces to a one-dimensional optimization problem.

Coordinate descent for Lasso: solving the 1-D subproblem

Expand the quadratic term of optimization problem:

$$\frac{1}{2n}\|r^{(i)} - X_{\cdot i}u\|_2^2 = \frac{1}{2n}\|r^{(i)}\|_2^2 - \frac{1}{n}(X_{\cdot i}^\top r^{(i)})u + \frac{1}{2n}\|X_{\cdot i}\|_2^2 u^2.$$

Define

$$a_i = \frac{1}{n}\|X_{\cdot i}\|_2^2, \qquad \rho_i = \frac{1}{n}X_{\cdot i}^\top r^{(i)}.$$

Then the subproblem is equivalent to

$$\min_{u\in\mathbb{R}} \frac{a_i}{2}u^2 - \rho_i u + \lambda|u|,$$

since the constant term does not affect the minimizer.

Coordinate descent for Lasso: closed-form update

Now consider three cases.

Case 1: $u > 0$. Then $|u| = u$, so we minimize

$$\frac{a_i}{2}u^2 - \rho_i u + \lambda u.$$

Its derivative is

$$a_i u - \rho_i + \lambda.$$

Setting it to zero gives

$$u = \frac{\rho_i - \lambda}{a_i}.$$

This is valid only when $\rho_i > \lambda$.

Coordinate descent for Lasso: closed-form update

Case 2: $u < 0$. Then $|u| = -u$, so we minimize

$$\frac{a_i}{2}u^2 - \rho_i u - \lambda u.$$

Its derivative is

$$a_i u - \rho_i - \lambda.$$

Setting it to zero gives

$$u = \frac{\rho_i + \lambda}{a_i},$$

which is valid only when $\rho_i < -\lambda$.

Case 3: $u = 0$. This is optimal when neither of the above cases is valid, i.e.

$$|\rho_i| \leq \lambda.$$

Coordinate descent for Lasso: closed-form update

Therefore,

$$u^\star = \begin{cases} \dfrac{\rho_i - \lambda}{a_i}, & \rho_i > \lambda, \\ 0, & |\rho_i| \le \lambda, \\ \dfrac{\rho_i + \lambda}{a_i}, & \rho_i < -\lambda. \end{cases}$$

Hence the coordinate update is

$$\theta_i \leftarrow u^\star.$$

Coordinate descent result

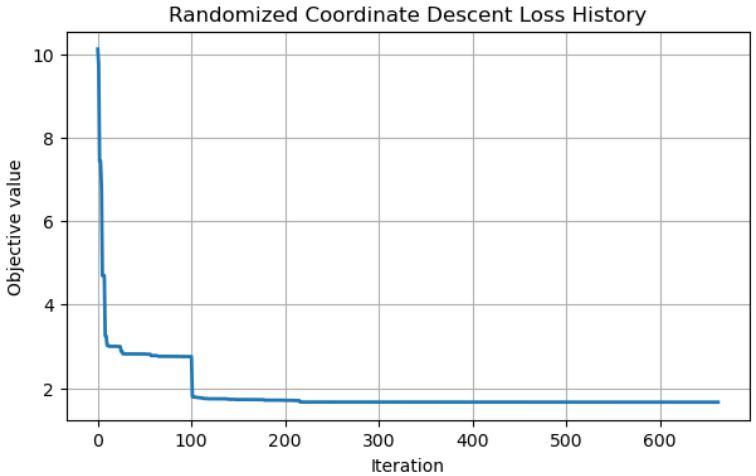


Figure: Coordinate descent result. Here one iteration means one coordinate update.

Other ways to choose the coordinate

All coordinate descent methods use the same update form:

$$\theta_{t+1} = \theta_t - \gamma_t \nabla_i f(\theta_t)\, e_i.$$

The main difference is the rule for selecting i.

Common choices include:

- **Uniform random choice:** choose each coordinate with probability $1/d$;
- **Importance sampling:** assign different probabilities to different coordinates;
- **Greedy choice:** select the coordinate with the largest partial derivative magnitude.

Outline

LASSO regression

Projected Gradient Descent

Coordinate Descent

Proximal Gradient Descent

Stochastic Gradient Descent

Why proximal gradient descent?

We consider composite optimization problems of the form

$$\min_{\theta \in \mathbb{R}^d} F(\theta) := g(\theta) + h(\theta),$$

where g is differentiable and h is convex but possibly non-differentiable.

- If we only look at the smooth part g, then from the current iterate θ_t, the usual gradient step would be
$$\eta_{t+1} = \theta_t - \gamma \nabla g(\theta_t).$$
- Take a usual gradient step for the smooth part g,
- Modify this step according to the non-smooth part h.

The proximal correction

- Starting from the gradient step η_{t+1}, we choose the next iterate by solving

$$\theta_{t+1} := \arg\min_{\theta \in \mathbb{R}^d} \left\{ h(\theta) + \frac{1}{2\gamma} \|\theta - \eta_{t+1}\|^2 \right\}.$$

  - $h(\theta)$ promotes the desired structure.
  - $\|\theta - \eta_{t+1}\|^2$ keeps the next iterate close to the gradient step.
- This defines the **proximal operator**:

$$\mathrm{prox}_{\gamma h}(\eta) := \arg\min_{\theta \in \mathbb{R}^d} \left\{ h(\theta) + \frac{1}{2\gamma} \|\theta - \eta\|^2 \right\}.$$

- Therefore, the update can be written as

$$\theta_{t+1} = \mathrm{prox}_{\gamma h}\big(\theta_t - \gamma \nabla g(\theta_t)\big).$$

Proximal gradient descent algorithm

Algorithm: Proximal Gradient Descent

Input: $\theta_0 \in \mathbb{R}^d$, $\gamma > 0$, $tol > 0$
for $t = 0, 1, 2, \ldots, T-1$
$\quad \eta_{t+1} := \theta_t - \gamma \nabla g(\theta_t)$,
$\quad \theta_{t+1} := \mathrm{prox}_{\gamma h}(\eta_{t+1})$,
$\quad$ if $\|\theta_{t+1} - \theta_t\|_2 \leq tol$, stop
end

- The first step is an ordinary gradient step on the smooth part g.
- The second step handles the non-smooth part h through the proximal operator.
- In implementation, the stopping rule is again based on the parameter change $\|\theta_{t+1} - \theta_t\|_2$.

Relation to projected gradient descent

- Projected gradient descent can be viewed as a special case of proximal gradient descent.
- Suppose

$$h(\theta) = I_\Theta(\theta),$$

where I_Θ is the indicator function of a closed convex set Θ:

$$I_\Theta(\theta) = \begin{cases} 0, & \theta \in \Theta, \\ +\infty, & \theta \notin \Theta. \end{cases}$$

- Then

$$\mathrm{prox}_{\gamma I_\Theta}(\eta) = \Pi_\Theta(\eta),$$

which is exactly the Euclidean projection onto Θ.

- Hence projected gradient descent fits into the same framework:

$$\eta_{t+1} = \theta_t - \gamma \nabla g(\theta_t), \qquad \theta_{t+1} = \mathrm{prox}_{\gamma I_\Theta}(\eta_{t+1}).$$

The proximal step for Lasso

- For Lasso, the non-smooth part is

$$h(\theta) = \lambda \|\theta\|_1.$$

- Therefore, after the gradient step

$$\eta_{t+1} = \theta_t - \gamma \frac{1}{n} X^\top (X\theta_t - y),$$

the next iterate is defined by

$$\theta_{t+1} = \mathrm{prox}_{\gamma\lambda\|\cdot\|_1}(\eta_{t+1}),$$

where

$$\mathrm{prox}_{\gamma\lambda\|\cdot\|_1}(\eta) = \arg\min_{\theta \in \mathbb{R}^d} \left\{ \lambda\|\theta\|_1 + \frac{1}{2\gamma}\|\theta - \eta\|^2 \right\}.$$

The proximal step for Lasso

- Both terms are separable across coordinates:

$$\|\theta\|_1 = \sum_{j=1}^{d} |\theta_j|, \qquad \|\theta - \eta\|^2 = \sum_{j=1}^{d} (\theta_j - \eta_j)^2.$$

- Hence

$$\lambda\|\theta\|_1 + \frac{1}{2\gamma}\|\theta - \eta\|^2 = \sum_{j=1}^{d} \left[\lambda|\theta_j| + \frac{1}{2\gamma}(\theta_j - \eta_j)^2\right].$$

- Therefore, the proximal problem splits into *d* independent one-dimensional problems. For each coordinate *j*, we solve

$$\min_{u \in \mathbb{R}} \left\{\lambda|u| + \frac{1}{2\gamma}(u - \eta_j)^2\right\}.$$

Soft-thresholding

- Hence the solution is the **soft-thresholding operator**

$$S_{\gamma\lambda}(\eta_j) = \begin{cases} \eta_j - \gamma\lambda, & \eta_j > \gamma\lambda, \\ 0, & |\eta_j| \leq \gamma\lambda, \\ \eta_j + \gamma\lambda, & \eta_j < -\gamma\lambda. \end{cases}$$

- Therefore,

$$\theta_{t+1} = S_{\gamma\lambda}(\eta_{t+1}),$$

where $S_{\gamma\lambda}$ is applied coordinate-wise.

Proximal gradient descent result

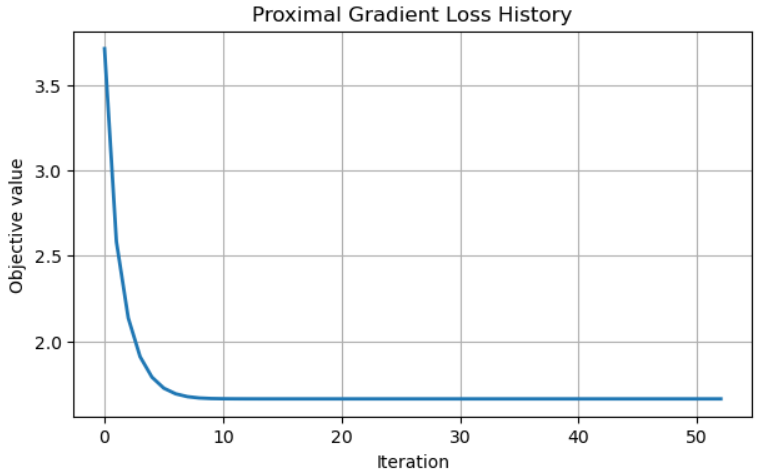


Figure: Proximal gradient descent result

Outline

LASSO regression

Projected Gradient Descent

Coordinate Descent

Proximal Gradient Descent

Stochastic Gradient Descent

Stochastic Gradient Descent

We start from the standard gradient descent update

$$\theta_{t+1} = \theta_t - \gamma_t \nabla f(\theta_t).$$

When the objective is large-scale, computing the full gradient $\nabla f(\theta_t)$ at every iteration may be expensive. So instead of using the full gradient, we use a random vector g_t that serves as a cheaper approximation.

This gives the stochastic gradient descent (SGD) update:

$$\theta_{t+1} = \theta_t - \gamma_t g_t.$$

- $\gamma_t > 0$ is the stepsize;
- g_t is a random vector constructed at iteration t and g_t should point in the same direction as the true gradient on average.

Stochastic Gradient Descent

A standard requirement is the **conditional unbiasedness**:

$$\mathbb{E}[g_t \mid \theta_t] = \nabla f(\theta_t).$$

More formally, the conditioning can be taken on all information available up to step t.

So SGD has the same form as GD, but replaces the exact gradient by a random and cheaper estimator.

A remaining question is: how do we construct g_t?

Finite-sum setting

In supervised learning, we often have training samples

$$(x_1, y_1), \ldots, (x_n, y_n).$$

A common objective is the empirical risk

$$f(\theta) = \frac{1}{n} \sum_{i=1}^{n} f_i(\theta),$$

where $f_i(\theta)$ is the loss contributed by the ith sample.
Then the full gradient is

$$\nabla f(\theta) = \frac{1}{n} \sum_{i=1}^{n} \nabla f_i(\theta).$$

When n is large, computing $\nabla f(\theta_t)$ can be expensive.

Finite-sum setting

A natural idea is to sample one index

$$i_t \in \{1, \dots, n\},$$

and define

$$g_t = \nabla f_{i_t}(\theta_t).$$

So SGD uses only one sample gradient at each iteration.

Why this choice of g_t makes sense

Assume that i_t is sampled uniformly from $\{1, \dots, n\}$. Then, conditioning on θ_t, we have

$$\mathbb{E}[g_t \mid \theta_t] = \mathbb{E}\big[\nabla f_{i_t}(\theta_t) \mid \theta_t\big].$$

Because each index is chosen with probability $1/n$,

$$\mathbb{E}[g_t \mid \theta_t] = \frac{1}{n} \sum_{i=1}^{n} \nabla f_i(\theta_t) = \nabla f(\theta_t).$$

So g_t is random, but it is an unbiased estimator of the full gradient.

This is why $g_t = \nabla f_{i_t}(\theta_t)$ is a natural choice in SGD.

Summary

- **Projected gradient descent** extends gradient descent to constrained optimization by adding a projection step onto the feasible set.
- **Coordinate descent** updates only one coordinate at a time, which can make each iteration much cheaper in high dimensions.
- **Proximal gradient descent** extends gradient descent to composite objectives

$$F(\theta) = g(\theta) + h(\theta),$$

  where the smooth part is handled by a gradient step and the non-smooth part by a proximal step.
- **Stochastic proximal gradient descent** uses one sampled data point to build a stochastic gradient for the smooth part, followed by a proximal soft-threshold step for the ℓ_1 term.