Statistical Machine Learning for Data Science

Zaid Harchaoui

DATA 558

Week 3

Lecture 3: Outline

- Ridge regression and ℓ_2^2-regularized Logistic Regression
- Gradient Descent with Adaptive Step-size

Ridge Regression

Training data $(x_1, y_1), \ldots, (x_n, y_n)$ in $\mathbb{R}^d \times R$.

$$\min_{\beta \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} (y_i - x_i^T \beta)^2 + \lambda \|\beta\|_2^2 ,$$

that is, if you expand

$$\min_{\beta \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} \left(y_i - \sum_{j=1}^{d} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{d} \beta_j^2 .$$

Empirical risk and Regularization

$$\text{Residual Sum of Squares (RSS)} \quad \sum_{i=1}^{n}(y_i - x_i^T\beta)^2$$

$$\text{Regularization Penalty} \quad \lambda\|\beta\|_2^2$$

Regularization

The term

$$\lambda \|\beta\|_2^2$$

is also called a *shrinkage penalty*.

It has the effect of *shrinking the estimates of* β_j *towards 0*.

$$\begin{aligned} \lambda \approx 0 \quad & \text{no effect} \\ \lambda \to \infty \quad & \text{shrinkage towards 0} \end{aligned}$$

Where is the intercept?

Standardize the data before you solve the ridge regression problem.

$$\tilde{x}_{ij} = \frac{x_{ij}}{\sqrt{1/n \sum_{i=1}^{n} (x_{ij} - \bar{x}_j)^2}}$$

Then the intercept is given by the simple formula

$$\hat{\beta}_0 = \frac{1}{n} \sum_{i=1}^{n} y_i \ .$$

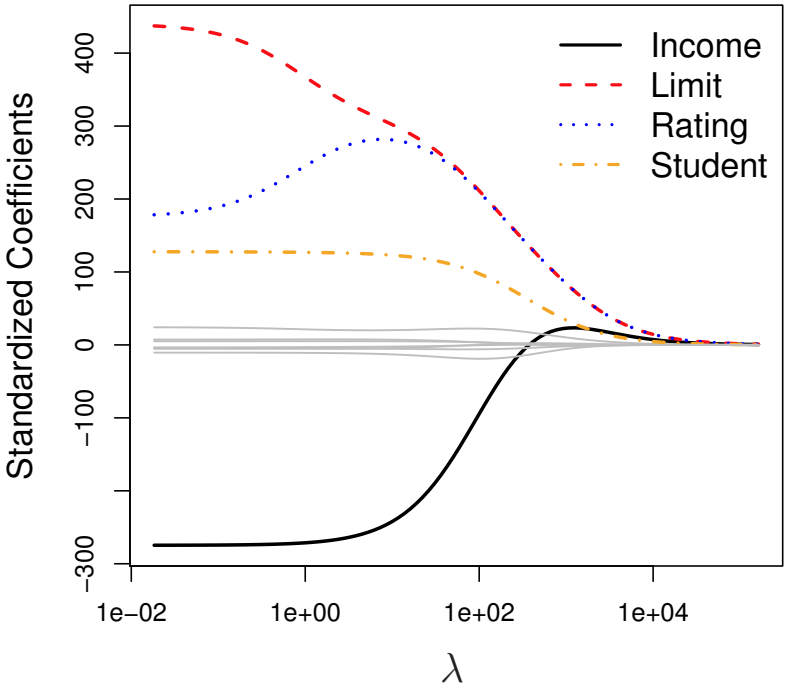
and does need to be found by the optimization algorithm.

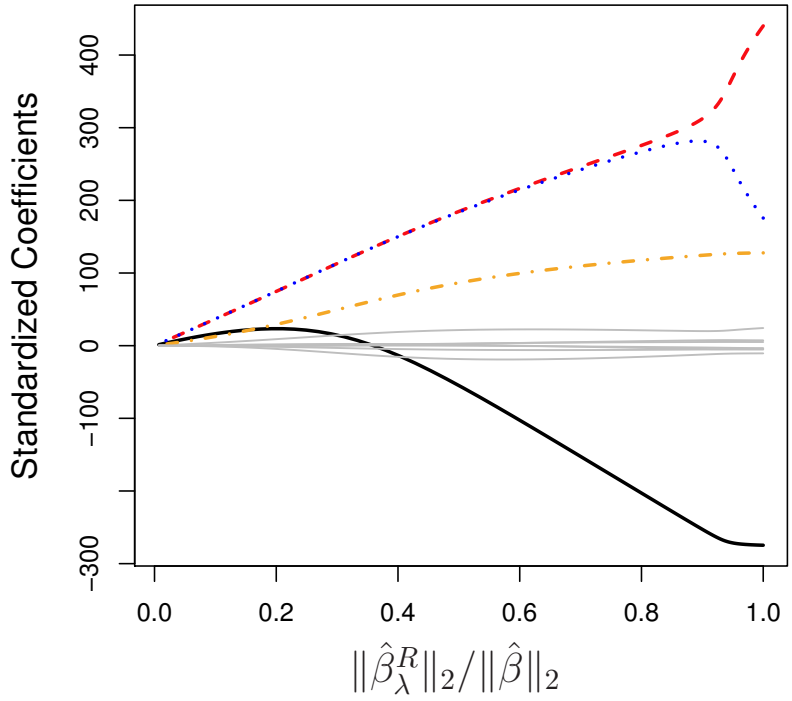
Example: Credit Data

Predict **balance** (average credit card debt for a number of individuals) from the predictors:

- age
- cards (number of credit cards)
- education (years of education)
- income (in thousands of dollars)
- limit (credit limit)
- rating (credit rating
- gender
- student (student
- status (marital status)
- ethnicity (Caucasian, African American or Asian).

Example: Credit Data





$\hat{\beta}_\lambda^R$ Ridge regression with reg. penalty λ

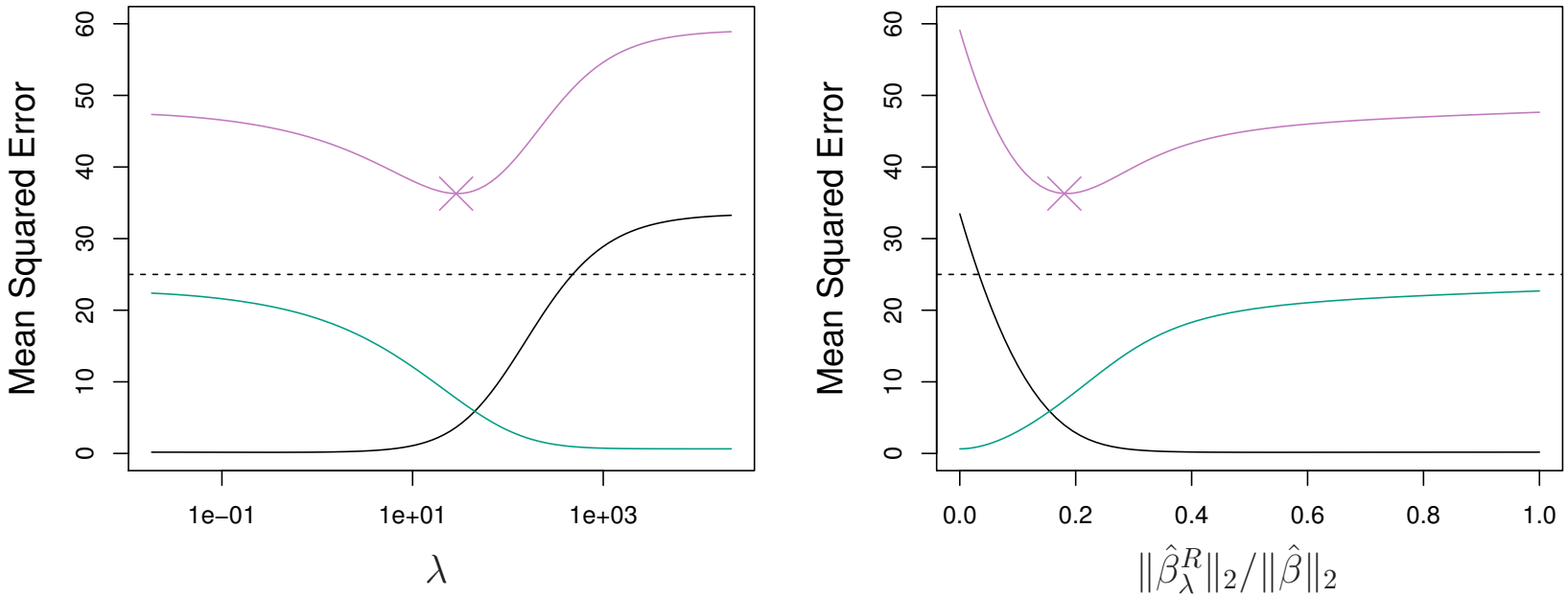
$\hat{\beta}$ Ridge regression with reg. penalty ∞ (OLS)

Bias-Variance trade-off

$$\underbrace{\mathbb{E}(y - \hat{f}(x))^2}_{\text{Expected test MSE}} = \text{Var}(f(x)) + [\text{Bias}(f(x))]^2 + \text{Var}(\epsilon)$$

- **Expected MSE**: average test MSE over infinite number of training sets.
- **Variance**: the amount of which $\hat{f}$ would change if we estimated it using a different training set.
- **Bias**: the error introduced by adopting this particular model

Bias-Variance trade-off



Squared bias (black), variance (green), and test mean squared error (purple) for ridge regression predictions on simulated data set.
Horizontal dashed lines indicate minimum possible MSE.
Purple crosses indicate ridge regression models for which the MSE is smallest.

ℓ_2^2-regularized Logistic Regression

Statistics notation

Training data $(x_1, y_1), \ldots, (x_n, y_n)$ in $\mathbb{R}^d \times \{0, 1\}$.

$$\min_{\beta \in \mathbb{R}} \frac{1}{n} \sum_{i=1}^{n} \left\{ y_i \beta^T x_i \log\left(1 + \exp(\beta^T x_i)\right) \right\} + \lambda \|\beta\|_2^2 .$$

Machine Learning notation

Training data $(x_1, y_1), \ldots, (x_n, y_n)$ in $\mathbb{R}^d \times \{-1, 1\}$.

$$\min_{\beta \in \mathbb{R}} \frac{1}{n} \sum_{i=1}^{n} \left\{ \log\left(1 + \exp(-y_i \beta^T x_i)\right) \right\} + \lambda \|\beta\|_2^2 ,$$

Penalized likelihood interpretation

Logit Model

$$p(X) := \mathbb{P}(Y = 1|X = x) = \frac{\exp(f_\beta(x))}{1 + \exp(f_\beta(x))} \ ,$$

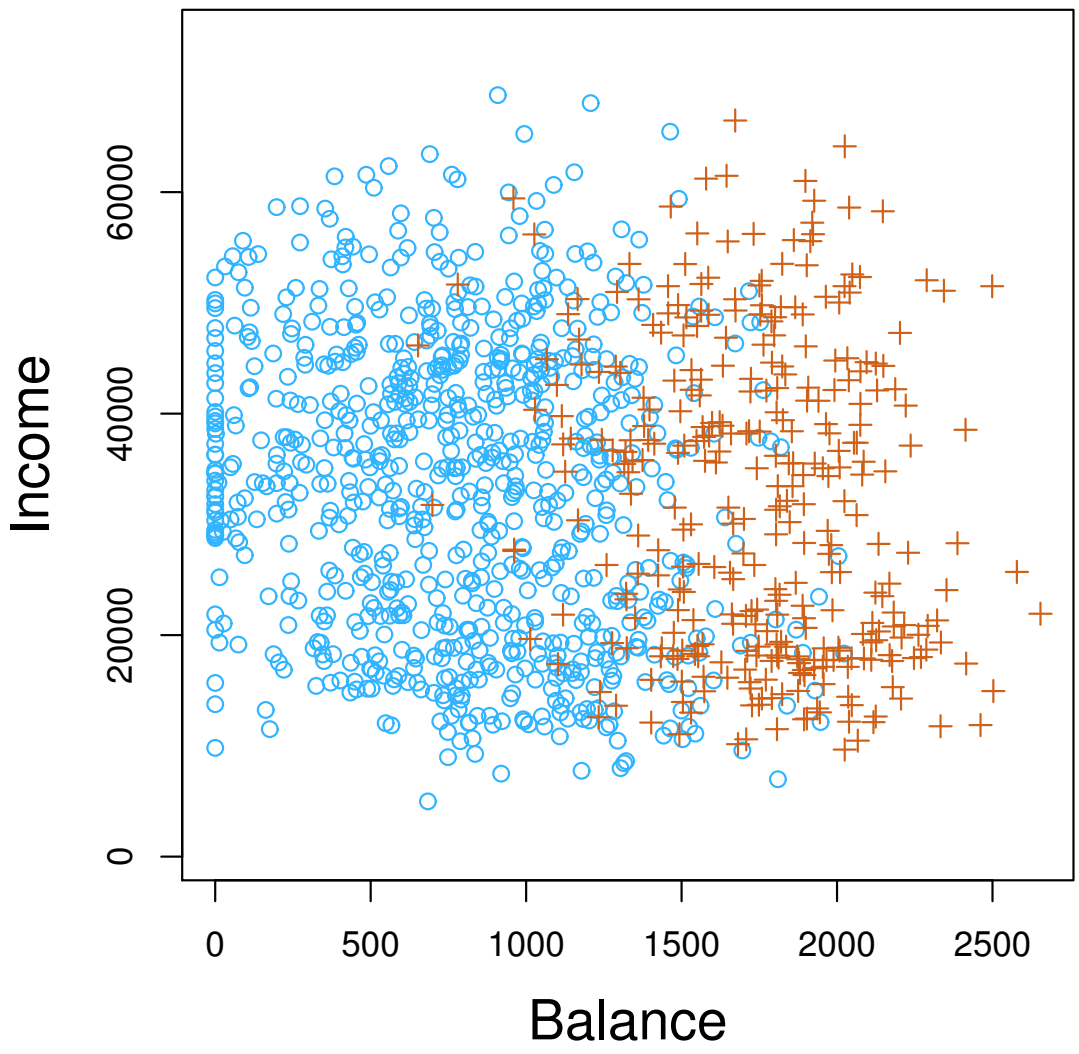
or

$$log\left(\frac{p(X)}{1 - p(X)}\right) = f_\beta(x) \ ,$$

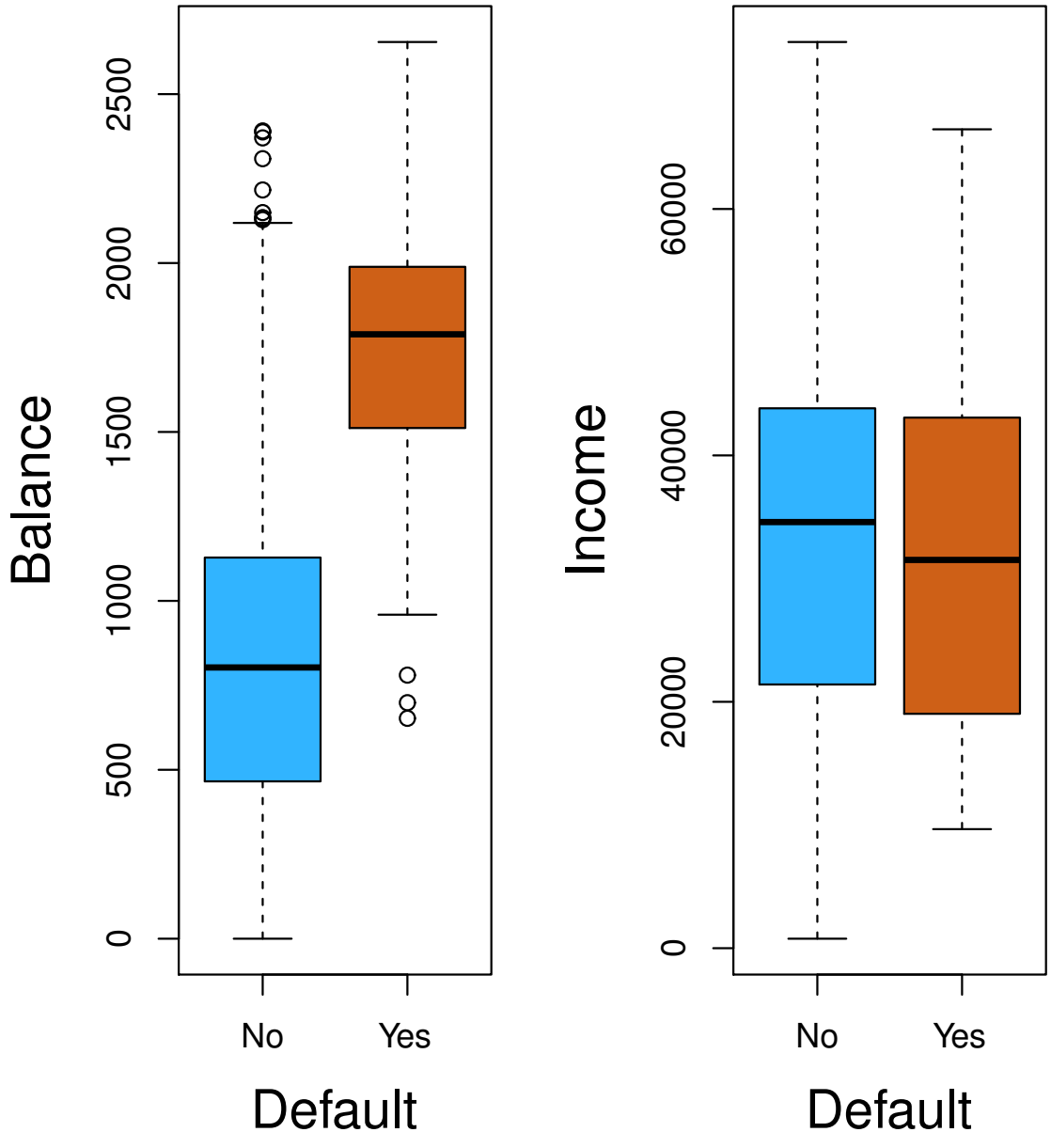
where

$$f_\beta(x) = \beta_0 + \beta^T x \ .$$

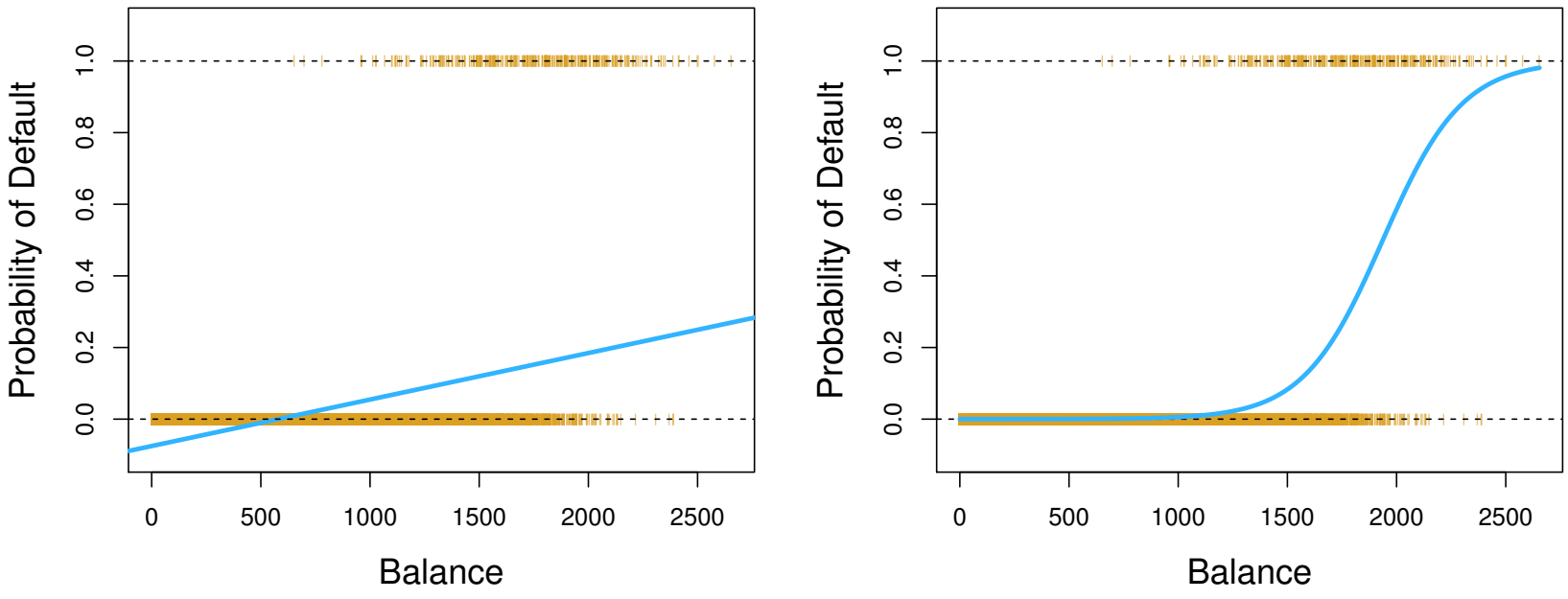
Example: Credit card default



Example: Credit card default



Example: Credit card default



Logistic regression ensures that our estimate for $p(X)$ lies between 0 and 1.

What about the intercept?

Standardize the data before you solve the logistic regression problem.

$$\tilde{x}_{ij} = \frac{x_{ij}}{\sqrt{1/n \sum_{i=1}^{n} (x_{ij} - \bar{x}_j)^2}}$$

If there is class imbalance (case-control studies), then the intercept can be adjusted from data for better performance.

Penalized likelihood interpretation

Multinomial Logit Model

$$log\left(\frac{\mathbb{P}(Y=1|X=x)}{\mathbb{P}(Y=k|X=x)}\right) = f_{\beta_1}(x)$$
$$log\left(\frac{\mathbb{P}(Y=2|X=x)}{\mathbb{P}(Y=k|X=x)}\right) = f_{\beta_2}(x)$$
$$\cdots\cdots$$
$$log\left(\frac{\mathbb{P}(Y=k-1|X=x)}{\mathbb{P}(Y=k|X=x)}\right) = f_{\beta_{k-1}}(x)$$

where the choice of the reference class is arbitrary.

Penalized likelihood interpretation

Multinomial Logit Model

We have for all $\ell = 1, \ldots, k-1$

$$\mathbb{P}(Y = \ell | X = x) = \frac{\exp(\beta_{\ell,0} + \beta_\ell^T x)}{1 + \sum_{m=1}^{k-1} \exp(\beta_{m,0} + \beta_m^T x)}$$

and

$$\sum_{\ell=1}^{k} \mathbb{P}(Y = \ell | X = x) = 1 \, .$$

Supervised learning

General objective

Let $(x_1, y_1), \ldots, (x_n, y_n) \in \mathbb{R}^d \times \{1, \ldots, k\}$ be labelled training examples

$$\min_{B \in \mathbb{R}^{d \times k}} \quad \lambda \Omega(B) + \frac{1}{n} \sum_{i=1}^{n} L\left(y_i, B^T x_i\right)$$

Multi-task setting

$$B = [\beta_1^T, \ldots, \beta_k^T]$$

Supervised learning

General objective

Let $(x_1, y_1), \ldots, (x_n, y_n) \in \mathbb{R}^d \times \{1, \ldots, k\}$ be labelled training examples

$$\min_{B \in \mathbb{R}^{d \times k}} \quad \lambda \, \Omega(B) + \frac{1}{n} \sum_{i=1}^{n} L\big(y_i, B^T x_i\big)$$

Large-scale setting

$$n \gg 1, \quad d \gg 1, \quad k \gg 1$$

Gradient descent

Gradient descent

Grouping the regularization penalty and the empirical risk in the objective

$$\nabla_B F(B) = \frac{1}{n} \sum_{i=1}^{n} \left\{ n\lambda \, \Omega(B) + L\left(y_i, B^T x_i\right) \right\}$$

Gradient descent

Gradient descent

Grouping the regularization penalty and the empirical risk, and expanding the sum onto the examples

$$\nabla_B F(B) = \frac{1}{n} \sum_{i=1}^{n} \left\{ n\lambda \, \Omega(B) + L\left(y_i, B^T x_i\right) \right\}$$

$$= \nabla_B \left\{ \frac{1}{n} \sum_{i=1}^{n} L(B; x_i, y_i) \right\}$$

Gradient descent

Gradient descent

- **Initialize**: $B = 0$
- **Iterate**:

$$
\begin{aligned}
B_{t+1} &= B_t - \eta_t \nabla F(B) \\
&= B_t - \eta_t \nabla_B \left\{ \frac{1}{n} \sum_{i=1}^{n} L(B; x_i, y_i) \right\}
\end{aligned}
$$

Gradient descent

Gradient descent

- **Initialize**: $B = 0$
- **Iterate**:

$$
\begin{aligned}
B_{t+1} &= B_t - \eta_t \nabla_B F(B) \\
&= B_t - \eta_t \nabla_B \left\{ \frac{1}{n} \sum_{i=1}^{n} L(B; x_i, y_i) \right\}
\end{aligned}
$$

Gradient descent

Strengths and weaknesses

- Strength: robust to setting of step-size sequence (line-search)
- Weakness: demanding disk/memory requirements

Tuning the step-size

Initial step-size estimate

- **standardize the data**
- use the formula

$$\eta_0 \propto \frac{1}{L}\,, \quad \text{where } L = \text{maximum-eigenvalue}\left(\frac{1}{n}X^T X\right) + \lambda$$

Practical advice

- Subsample the dataset
- Python $\rightarrow$ `largest-eigh`

Tuning the step-size

Backtracking rule

- **Initialize**: Stepsize $\eta_0 = \eta_{t-1}$ if $t \geq 1$ or $\eta_0 \approx 1/L$ if $t = 1$; Iteration counter $\ell = 1$.
- **Iterate**:

$$\eta_\ell = \gamma \eta_{\ell-1}$$

until the condition is satisfied

$$F(B_t - \eta_\ell \nabla F(B_t)) \leq F(B_t) + \alpha \eta_\ell \|\nabla F(B_t)\|^2 \ .$$

where $\alpha = 0.5$ and $\gamma = 0.8$. Once condition is satisfied, set η_t to the value found.

Gradient descent with adaptive step-size

- **Initialize**: $B_0 = 0$.
- **Iterate**:

  Find η_t with backtracking rule.

$$
\begin{aligned}
B_{t+1} &= B_t - \eta_t \nabla_B F(B) \\
&= B_t - \eta_t \nabla_B \left\{ \frac{1}{n} \sum_{i=1}^{n} L(B; x_i, y_i) \right\}
\end{aligned}
$$

Fast Gradient Method

- **Initialize**: $B = 0$ and $\theta_0 = 0$.
- **Iterate**:

  Find η_t with backtracking rule.

$$B_{t+1} = \theta_t - \eta_t \nabla_\theta F(\theta)$$

$$\theta_{t+1} = B_{t+1} + \frac{t}{t+3}(B_{t+1} - B_t)$$