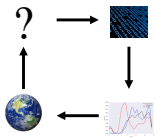
Finishing

Regularization

Slides by:

Joseph E. Gonzalez

jegonzal@cs.berkeley.edu



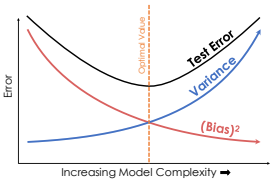
Last Time

$$\mathbf{E}\left[(y - f_\theta(x))^2\right] =$$

$$\mathbf{E}\left[(y - h(x))^2\right] + \quad \textbf{Noise Term}$$

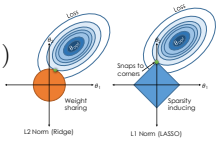
$$\mathbf{E}\left[(h(x) - \mathbf{E}_\theta[f_\theta(x)])^2\right] + \quad \textbf{(Bias)}^2 \textbf{ Term}$$

$$\mathbf{E}\left[(\mathbf{E}_\theta[f_\theta(x)] - f_\theta(x))^2\right] \quad \textbf{Variance Term}$$



Regularization

$$\hat{\theta} = \arg\min_\theta \frac{1}{n}\sum_{i=1}^{n} \mathbf{Loss}\left(y_i, f_\theta(x_i)\right) + \lambda \mathbf{R}(\theta)$$



Basic Idea of Regularization

Fit the Data

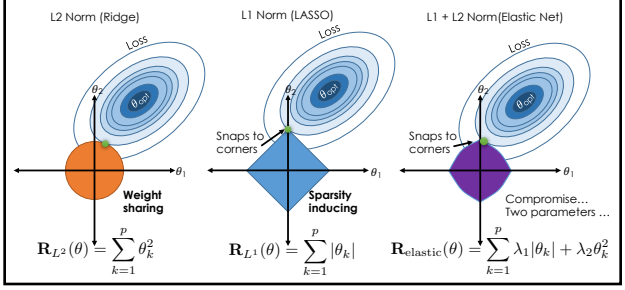
Penalize Complex Models

$$\hat{\theta} = \arg\min_\theta \frac{1}{n}\sum_{i=1}^{n} \mathbf{Loss}\left(y_i, f_\theta(x_i)\right) + \lambda \mathbf{R}(\theta)$$

Regularization Parameter

- How should we define **R(θ)**?
- How do we determine λ?

The Regularization Function **R(θ)**



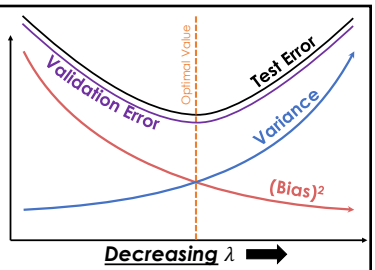
Determining the Optimal λ

$$\hat{\theta} = \arg\min_\theta \frac{1}{n}\sum_{i=1}^{n} \mathbf{Loss}\left(y_i, f_\theta(x_i)\right) + \lambda \mathbf{R}(\theta)$$

- Value of λ determines bias-variance tradeoff
  - Larger values → more regularization → more bias → less variance

Determining the Optimal λ

How do we determine λ?

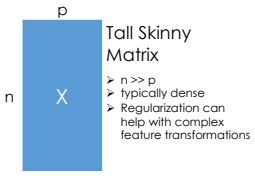


- Value of λ determines bias-variance tradeoff
  - Larger values → more regularization → more bias → less variance
- Determined through cross validation

Python Demo!

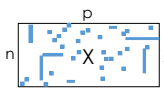
Regularization and **High-Dimensional Data**

Regularization is often used with high-dimensional data



Tall Skinny Matrix

- n >> p
- typically dense
- Regularization can help with complex feature transformations



High-dimensional sparse matrix

- p > n
- **requires** regularization
- Choice or Regularization depends on goal:
  - to determine **informative dimensions**
    - Consider L1 Regularization. Why?
  - to make **robust predictions**
    - Consider L2 (+L1) Regularization

Connection to Bayesian Priors

Regularization can be seen as a prior on the parameters

- Ridge Regression:

Lik.: $y \sim \mathbf{N}(x^T\theta, \sigma^2_{\text{noise}})$
Prior: $\theta \sim \mathbf{N}(0, 1/\lambda)$

Posterior (Assume IID) → (proportional)

$$\prod_{I=1}^{n} \exp\left(-\frac{(y - x^T\theta)^2}{2\sigma^2_{\text{noise}}} - \lambda\frac{\theta^2}{2}\right)$$

- LASSO:

Lik.: $y \sim \mathbf{N}(x^T\theta, \sigma^2_{\text{noise}})$
Prior: $\theta \sim \mathbf{Laplace}(0, p/\lambda)$

Posterior (Assume IID) → (proportional)

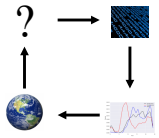
$$\prod_{I=1}^{n} \exp\left(-\frac{(y - x^T\theta)^2}{2\sigma^2_{\text{noise}}} - \lambda\sum_{k=1}^{p}|\theta_k|\right)$$

Logistic Regression
Classification with Linear Models

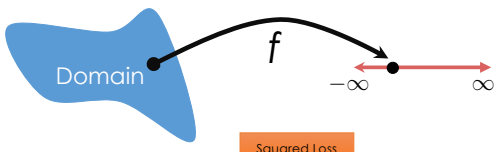
Slides by:

Joseph E. Gonzalez

jegonzal@cs.berkeley.edu



Recap: *Least Squares Regression*

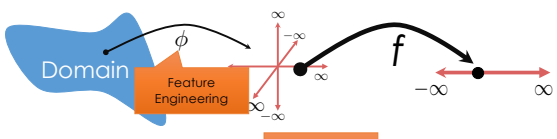


Squared Loss

$$\hat{\theta} = \arg\min_{\theta} \frac{1}{n}\sum_{i=1}^{n}(y_i - f_\theta(x_i))^2 + \lambda\mathbf{R}(\theta)$$

Regularization

Recap: *Least Squares Regression*

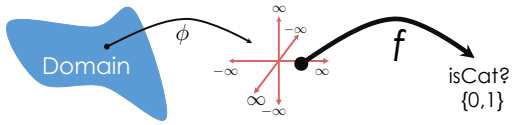


Squared Loss

$$\hat{\theta} = \arg\min_{\theta} \frac{1}{n}\sum_{i=1}^{n}(y_i - f_\theta(x_i))^2 + \lambda\mathbf{R}(\theta)$$

Regularization

Classification

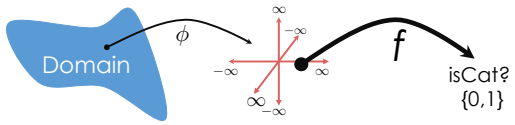


Can we just use least squares?

$$\hat{\theta} = \arg\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} (y_i - f_\theta(x_i))^2 + \lambda \mathbf{R}(\theta)$$

Python Demo!

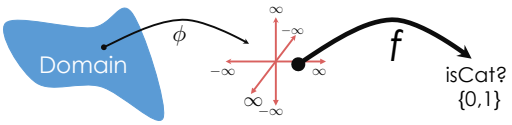
Classification



Can we just use least squares?

$$\hat{\theta} = \arg\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} (y_i - f_\theta(x_i))^2 + \lambda \mathbf{R}(\theta)$$

Classification



Can we just use least squares?

$\hat{\theta} = \arg\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} (y_i - f_\theta(x_i))^2 + \lambda \mathbf{R}(\theta)$

- Yes … (can be easy to compute)
- Need a decision function (e.g., f(x) > 0.5) …
- Difficult to interpret model …

Returning to the method of Maximum Likelihood

Recall: Least Squares Regression

Maximum Likelihood formulation

- Independently and identically distributed (IID)

$$y_i = x_i^T \theta + \epsilon_i \quad \text{Gaussian Noise } \mathbf{N}(0, \sigma^2_{\text{noise}}) \quad \Rightarrow \quad y_i \sim \mathbf{N}\left(x_i^T \theta, \sigma^2_{\text{noise}}\right)$$

- Probability of the data

$$\log \mathcal{L}(\theta) \propto -\sum_{i=1}^{n} \left(y_i - x_i^T \theta\right)^2$$

Let's apply this reasoning to classification

Classification: Maximum Likelihood Formulation

What would be a good model for the observation y_i?

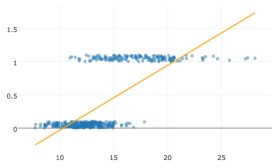
$$y_i \sim \mathbf{Bern}(p_i)$$

How do we define the probability for each observation?

$$p_i \stackrel{?}{=} x_i^T\theta$$

- Not bounded between [0,1]

How could we transform our line?



What would be a good model for the observation y_i?

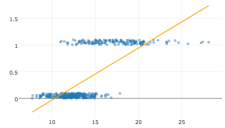
$$y_i \sim \mathbf{Bern}(p_i)$$

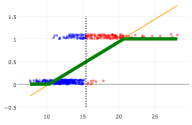
How do we define the probability for each observation?

$$p_i \stackrel{?}{=} x_i^T\theta$$

- Not bounded between [0,1]

How could we transform our line?





Truncate the line?

- P(Malignant | width = 10) = 0?
  - probably not … weird tail behavior
- Not smooth → Computational issues …

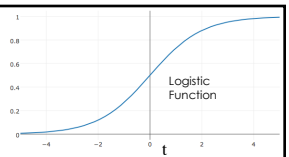
What does the data look like?

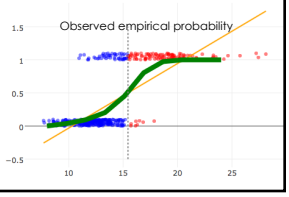
Python Demo!

Logistic Function

$$\sigma(t) = \frac{1}{1+e^{-t}}$$

- Activation function in ML
- Inverse link function in stats.
- Smoothly interpolates between 0 and 1
- Similar shape to what we observe in the data

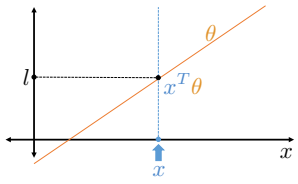




The Logistic Regression Model

Single Data Point Model (x, y)

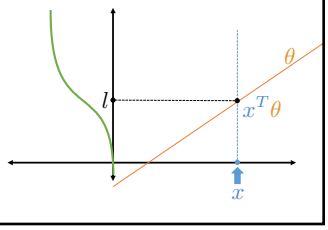
$l = x^T\theta$



The Logistic Regression Model

Single Data Point Model (x, y)

$l = x^T\theta$

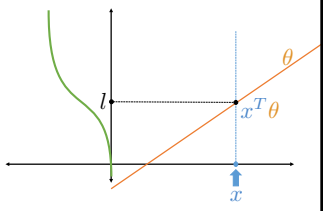


The Logistic Regression Model

Single Data Point Model (x, y)

$$l = x^T\theta$$

$$p = \sigma(l) = \frac{1}{1+\exp(-l)}$$



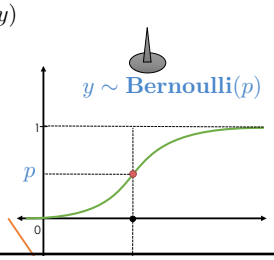
The Logistic Regression Model

Single Data Point Model (x, y)

$$l = x^T\theta$$

$$p = \sigma(l) = \frac{1}{1+\exp(-l)}$$

$$y \sim \textbf{Bernoulli}(p)$$



The Logistic Regression Model

Single Data Point Model (x, y)

$$l = x^T\theta$$

$$p = \sigma(l) = \frac{1}{1+\exp(-l)}$$

$$y \sim \textbf{Bernoulli}(p)$$

$$\Rightarrow y \sim \textbf{Bernoulli}\left(\sigma\left(x^T\theta\right)\right)$$

$$\mathbf{P}(y \mid x) = p^y(1-p)^{(1-y)} = \sigma\left(x^T\theta\right)^y\left(1-\sigma\left(x^T\theta\right)\right)^{1-y}$$

How do we find the "best" θ?

Maximum Likelihood Method

What do we need?

The Logistic Regression Model

What is the likelihood of an ***IID*** dataset?

$$\mathcal{L}(\theta) = \prod_{i=1}^{n} \mathbf{P}(y_i \mid x_i) = \prod_{i=1}^{n} \sigma\left(x_i^T\theta\right)^{y_i}\left(1-\sigma\left(x_i^T\theta\right)\right)^{(1-y_i)}$$

Steps of Maximum Likelihood

1. Define likelihood of the data ✓
2. Maximize Likelihood
   1. Take the derivative
   2. Set equal to 0
   3. Solve

First, let's simplify the likelihood expression by taking the **log.**

The Logistic Regression Model

$$\mathcal{L}(\theta) = \prod_{i=1}^{n} \sigma\left(x_i^T\theta\right)^{y_i}\left(1-\sigma\left(x_i^T\theta\right)\right)^{(1-y_i)}$$

Taking the log:

$$\log\mathcal{L}(\theta) = \sum_{i=1}^{n} y_i \log\sigma\left(x_i^T\theta\right) + (1-y_i)\log\left(1-\sigma\left(x_i^T\theta\right)\right)$$

Collecting terms:

$$\log\mathcal{L}(\theta) = \sum_{i=1}^{n} y_i \log\frac{\sigma\left(x_i^T\theta\right)}{1-\sigma\left(x_i^T\theta\right)} + \log\left(1-\sigma\left(x_i^T\theta\right)\right)$$

$$\log\mathcal{L}(\theta) = \sum_{i=1}^{n} y_i \log\sigma\left(x_i^T\theta\right) + (1-y_i)\log\left(1-\sigma\left(x_i^T\theta\right)\right)$$

Collecting terms:

$$\log\mathcal{L}(\theta) = \sum_{i=1}^{n} y_i \log\frac{\sigma\left(x_i^T\theta\right)}{1-\sigma\left(x_i^T\theta\right)} + \log\left(1-\sigma\left(x_i^T\theta\right)\right)$$

$$\log\frac{\sigma\left(x_i^T\theta\right)}{1-\sigma\left(x_i^T\theta\right)} \overset{\text{Defn. of } \sigma}{=} \log\frac{\frac{1}{1+\exp\left(-x_i^T\theta\right)}}{1-\frac{1}{1+\exp\left(-x_i^T\theta\right)}}$$

$$\log \mathcal{L}(\theta) = \sum_{i=1}^{n} y_i \log \frac{\sigma(x_i^T\theta)}{1-\sigma(x_i^T\theta)} + \log\left(1-\sigma(x_i^T\theta)\right)$$

$$\log \frac{\sigma(x_i^T\theta)}{1-\sigma(x_i^T\theta)} \underset{\text{Defn. of } \sigma}{=} \log \frac{\frac{1}{1+\exp(-x_i^T\theta)} \times (1+\exp(-x_i^T\theta))}{1-\frac{1}{1+\exp(-x_i^T\theta)} \times (1+\exp(-x_i^T\theta))}$$

$$\underset{\text{Alg.}}{=} \log \frac{1}{1+\exp(-x_i^T\theta)-1}$$

$$\underset{\text{Alg.}}{=} \log \exp(x_i^T\theta) \underset{\text{Alg.}}{=} x_i^T\theta$$

$$\log \mathcal{L}(\theta) = \sum_{i=1}^{n} y_i \log \frac{\sigma(x_i^T\theta)}{1-\sigma(x_i^T\theta)} + \log\left(1-\sigma(x_i^T\theta)\right)$$

$$\log \frac{\sigma(x_i^T\theta)}{1-\sigma(x_i^T\theta)} = x_i^T\theta = \log \frac{\mathbf{P}(y=1\,|\,x)}{\mathbf{P}(y=0\,|\,x)}$$ **Log odds**

- **Logistic activation function** implies that $x^T\theta$ defines the log odds:

$$\text{Log Odds} = x_i^T\theta = 0 \quad \Rightarrow \quad \mathbf{P}(y=1\,|\,x) = \mathbf{P}(y=0\,|\,x)$$
$$\text{Log Odds} = x_i^T\theta > 0 \quad \Rightarrow \quad \mathbf{P}(y=1\,|\,x) > \mathbf{P}(y=0\,|\,x)$$
$$\text{Log Odds} = x_i^T\theta < 0 \quad \Rightarrow \quad \mathbf{P}(y=1\,|\,x) < \mathbf{P}(y=0\,|\,x)$$

$$\log \mathcal{L}(\theta) = \sum_{i=1}^{n} y_i x_i^T\theta + \log\left(1-\sigma(x_i^T\theta)\right)$$

$$1-\sigma(x_i^T\theta) \underset{\text{Defn. of } \sigma}{=} 1-\frac{1}{1+\exp(-x_i^T\theta)} \underset{\text{Alg.}}{=} \frac{\exp(-x_i^T\theta)}{1+\exp(-x_i^T\theta)}$$

$$\underset{\text{Alg.}}{=} \frac{1}{1+\exp(x_i^T\theta)} \underset{\text{Defn. of } \sigma}{=} \sigma(-x_i^T\theta)$$

$$\log \mathcal{L}(\theta) = \sum_{i=1}^{n} y_i x_i^T\theta + \log\left(1-\sigma(x_i^T\theta)\right)$$

Useful identity

$$1-\sigma(x_i^T\theta) = \sigma(-x_i^T\theta)$$

$$= \sum_{i=1}^{n} y_i x_i^T\theta + \log\sigma(-x_i^T\theta)$$

$$\log \mathcal{L}(\theta) = \sum_{i=1}^{n} y_i x_i^T\theta + \log\sigma(-x_i^T\theta)$$ Simplified!

Steps

1. Define likelihood of the data ✓
2. Maximize Likelihood
   1. Take the derivative
   2. Set equal to 0
   3. Solve

$$\log \mathcal{L}(\theta) = \sum_{i=1}^{n} y_i x_i^T\theta + \log\sigma(-x_i^T\theta)$$ Simplified!

Steps

1. Define likelihood of the data ✓
2. Maximize Likelihood
   1. Take the derivative
   2. Set equal to 0
   3. Solve

$$\nabla_\theta \log \mathcal{L}(\theta) = \sum_{i=1}^{n} \nabla_\theta y_i x_i^T\theta + \nabla_\theta \log\sigma(-x_i^T\theta)$$

$$\nabla_\theta \log \mathcal{L}(\theta) = \sum_{i=1}^{n} \nabla_\theta y_i x_i^T \theta + \nabla_\theta \log \sigma\left(-x_i^T \theta\right)$$

Gradient $a^T\theta = a$

$$= \sum_{i=1}^{n} y_i x_i + \nabla_\theta \log \sigma\left(-x_i^T \theta\right)$$

Chain Rule

$$= \sum_{i=1}^{n} y_i x_i + \frac{1}{\sigma\left(-x_i^T \theta\right)} \nabla_\theta \sigma\left(-x_i^T \theta\right)$$

$$\nabla_\theta \log \mathcal{L}(\theta) = \sum_{i=1}^{n} y_i x_i + \frac{1}{\sigma\left(-x_i^T \theta\right)} \nabla_\theta \sigma\left(-x_i^T \theta\right)$$

Useful Identity

$$\frac{\partial}{\partial t}\sigma(t) = \frac{\partial}{\partial t}\frac{1}{1+e^{-t}} \overset{\text{Chain Rule}}{=} \frac{-1}{(1+e^{-t})^2}\frac{\partial}{\partial t}\left(1+e^{-t}\right)$$

$$\overset{\text{Chain Rule}}{=} \frac{e^{-t}}{(1+e^{-t})^2} \overset{\text{Alg.}}{=} \left(\frac{1}{1+e^{-t}}\right)\left(\frac{e^{-t}}{1+e^{-t}}\right)$$

$$\overset{\text{Alg.}}{=} \left(\frac{1}{1+e^{-t}}\right)\left(\frac{1}{e^{t}+1}\right) \overset{\text{Defn. of } \sigma}{=} \sigma(t)\sigma(-t)$$

$$\nabla_\theta \log \mathcal{L}(\theta) = \sum_{i=1}^{n} y_i x_i + \frac{1}{\sigma\left(-x_i^T \theta\right)} \nabla_\theta \sigma\left(-x_i^T \theta\right)$$

Useful Identity

$$\frac{\partial}{\partial t}\sigma(t) = \sigma(t)\sigma(-t)$$

$$= \sum_{i=1}^{n} y_i x_i - \frac{\sigma\left(-x_i^T \theta\right)}{\sigma\left(-x_i^T \theta\right)} \sigma\left(x_i^T \theta\right) x_i$$

$$= \sum_{i=1}^{n} \left(y_i - \sigma\left(x_i^T \theta\right)\right) x_i$$

$$\nabla_\theta \log \mathcal{L}(\theta) = \sum_{i=1}^{n} \left(y_i - \sigma\left(x_i^T \theta\right)\right) x_i$$

Steps

1. Define likelihood of the data ✓
2. Maximize Likelihood
   1. Take the derivative ✓
   2. Set equal to 0
   3. Solve

$$\nabla_\theta \log \mathcal{L}(\theta) = \sum_{i=1}^{n} \left(y_i - \sigma\left(x_i^T \theta\right)\right) x_i = 0$$

Steps

1. Define likelihood of the data ✓
2. Maximize Likelihood
   1. Take the derivative ✓
   2. Set equal to 0 ✓
   3. Solve

$$\nabla_\theta \log \mathcal{L}(\theta) = \sum_{i=1}^{n} \left(y_i - \sigma\left(x_i^T \theta\right)\right) x_i = 0$$

Steps

1. Define likelihood of the data ✓
2. Maximize Likelihood
   1. Take the derivative ✓
   2. Set equal to 0 ✓
   3. Solve

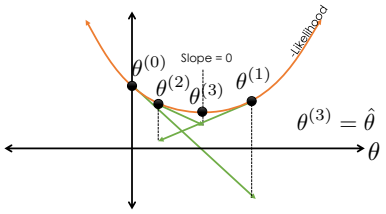
$$\sum_{i=1}^{n} \sigma\left(x_i^T \theta\right) x_i = \sum_{i=1}^{n} y_i x_i$$

$$\sum_{i=1}^{n} \frac{x_i}{1+\exp\left(x_i^T \theta\right)} = \sum_{i=1}^{n} y_i x_i$$

Gradient Descent

Stuck! ➔ No Analytic Solution!

Gradient Descent Intuition



Simple, fast, and works well in high dimensions.

The Gradient Descent Algorithm

$\theta^{(0)} \leftarrow$ random initial vector

For τ from 0 to convergence

$$\theta^{(\tau+1)} \leftarrow \theta^{(\tau)} - \rho(\tau)\nabla_\theta(-\log \mathcal{L}(\theta))\Big|_{\theta^{(\tau)}}$$ Evaluated at $\theta^{(\tau)}$

- $\rho(\tau)$ is the learning rate (e.g., $1/\tau$)
- Converges when gradient is ≈ 0

Stochastic Gradient Descent

- For many learning problems the gradient is a sum:

$$\nabla_\theta \log \mathcal{L}(\theta) = \sum_{i=1}^{n} \left(y_i - \sigma\left(x_i^T\theta\right)\right) x_i$$

  - For large n this can be costly
- What if we approximated the gradient by looking at a few random points:

$$\nabla_\theta \log \mathcal{L}(\theta) \approx \frac{n}{|\mathcal{B}|} \sum_{i\in\mathcal{B}} \left(y_i - \sigma\left(x_i^T\theta\right)\right) x_i$$

- What if we approximated the gradient by looking at a few random points:

$$\nabla_\theta \log \mathcal{L}(\theta) \approx \frac{n}{|\mathcal{B}|} \sum_{i\in\mathcal{B}} \left(y_i - \sigma\left(x_i^T\theta\right)\right) x_i$$

Batch Size

Batch of Random Points

- This is a reasonable estimator for the gradient
  - Unbiased …
- Often batch size is one!
- A key ingredient in the recent success of deep learning

Recap: Logistic Regression

- Defined a model: $y \sim \mathbf{Bernoulli}\left(\sigma\left(x^T\theta\right)\right)$
- Constructed a likelihood function:

$$\mathcal{L}(\theta) = \prod_{i=1}^{n} \sigma\left(x_i^T\theta\right)^{y_i} \left(1 - \sigma\left(x_i^T\theta\right)\right)^{(1-y_i)}$$

- Computed the gradient of the log-likelihood

$$\nabla_\theta \log \mathcal{L}(\theta) = \sum_{i=1}^{n} \left(y_i - \sigma\left(x_i^T\theta\right)\right) x_i$$

- Computed the gradient of the log-likelihood

$$\nabla_\theta \log \mathcal{L}(\theta) = \sum_{i=1}^{n} \left(y_i - \sigma\left(x_i^T\theta\right)\right) x_i$$

  - Realized we couldn't solve for 0 .. ☹
- Fall back to **(Stochastic) Gradient Descent**

$\theta^{(0)} \leftarrow$ random initial vector

For τ from 0 to convergence

$\mathcal{B} \leftarrow$ select k random indices

$$\theta^{(\tau+1)} \leftarrow \theta^{(\tau)} - \rho(\tau)\frac{n}{k} \sum_{i\in\mathcal{B}} \left(\sigma\left(x_i^T\theta\right) - y_i\right) x_i$$