Linear regression (continued…)

Reminders/Comments

- Assignment #2 is released
  - You do not have to use python; but I will continue to give you python code as a starting point
  - Note: python can have unexpected behavior; even if it runs, do not assume it is doing exactly what you expect
  - Python is pass by assignment (kinda like pass by reference)
- Thought questions due next week
  - t1.rtf under files, to see shared questions from Thought Questions 1
- Some notations added to notations document
- References list provided under files
- Posted Matlab example

Clarifications

- Matrix computations from matrix cookbook
- Norm notation
- Regularization and MAP

A further motivator for the (sometimes grueling) material

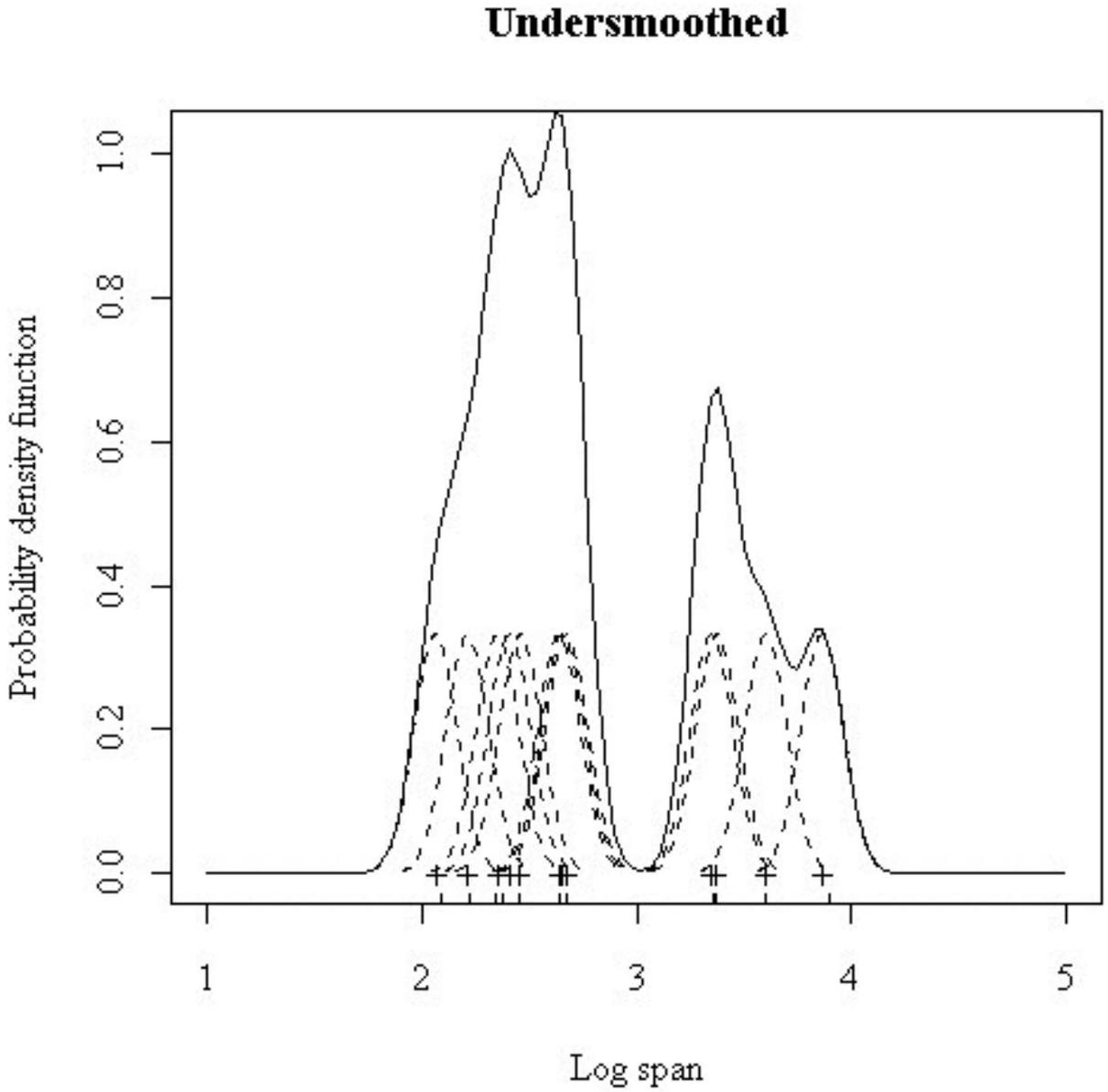
- Recent graduate, who took this course, working on predicting user actions on online ads

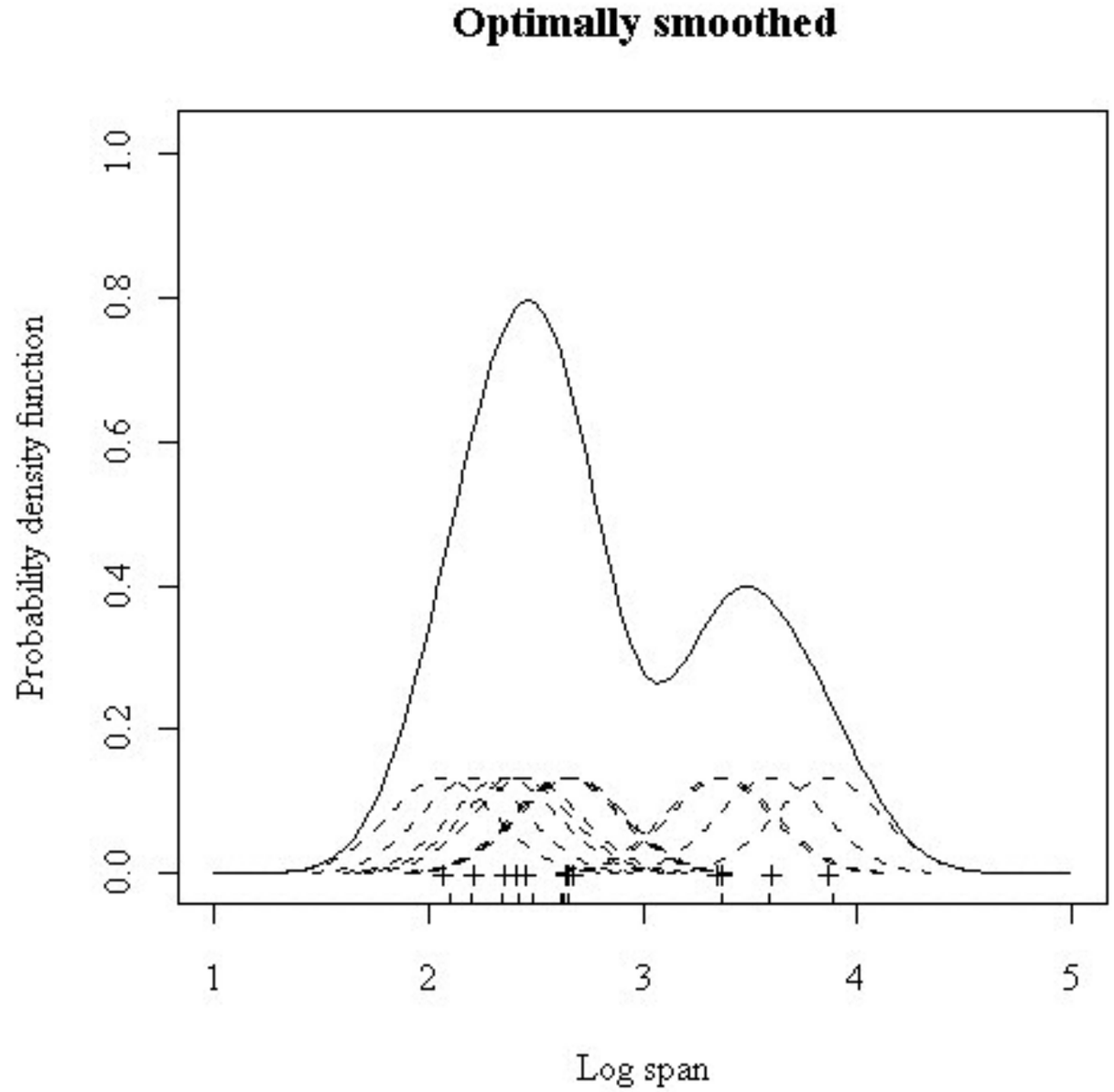
- Their comments:
  - "All the machine learning knowledge I use here is covered in the B555 course; I think those topics are essential for everyone who wants to become a data scientist in industry"
  - "I have done projects on modifying training algorithms, like adding priors or changing the loss functions"
  - "Although there are various packages to choose and one don't need to implement a model from scratch, knowing the details helps in parameter tuning and feature engineering and it also makes me creative in finding new ideas"

Thought question

- What I understood about the difference between parametric and non-parametric models, is that in parametric models we are trying to learn some parameters given the training data, and then the model will only depend on the parameters values. However, in non-parametric situation the model depends mainly on the training data. Does this imply that the parametric models generalize better than the non-parametric models?
  - Generalize —> the notion that the model will predict well on new (test) data that it was not trained on
  - We have seen this with overfitting —> if model overfits to the training data, it may not perform well on new data, does not generalize well
  - Regularization and priors is an approach to avoid overfitting, since does not solely rely on data (introduced expert bias, e.g. to simplicity)
  - Parametric models chosen by expert, non-parametric mostly by data, so typically non-parametric more likely to overfit
  - Non-parametric methods powerful; simply need to keep this property in mind

Example of non-parametric





Non-linear regression

- f(x) = y, where we need to learn the (continuous) function f
- If we do not have a lot of interesting features in x, then unlikely that a linear function will be enough
  - e.g., imagine predicting tomorrow's temperature using only todays
- Several representations that augment vector x based on results that state that they can represent any function
  - e.g. Stone-Weierstrass theorem states that every continuous function on finite interval can be closely approximated by a polynomial
- Other basis (e.g. radial basis function network and wavelets) have other justifications for representation properties

Linear regression for non-linear problems

$$f(x) = w_0 + w_1 x, \longrightarrow f(x) = \sum_{j=0}^{p} w_j x^j,$$

	$\mathbf{X}$		$\mathbf{\Phi}$		
1	x_1	$\rightarrow$	$\phi_0(x_1)$	...	$\phi_p(x_1)$
	x_2		...	...	...
	...		...	...	...
n	x_n		$\phi_0(x_n)$	...	$\phi_p(x_n)$

Figure 4.3: Transformation of an $n \times 1$ data matrix $\mathbf{X}$ into an $n \times (p+1)$ matrix $\mathbf{\Phi}$ using a set of basis functions ϕ_j, $j = 0, 1, \ldots, p$.

$$\mathbf{w}^* = \left(\mathbf{\Phi}^\top \mathbf{\Phi}\right)^{-1} \mathbf{\Phi}^\top \mathbf{y}.$$

Regularization intuition

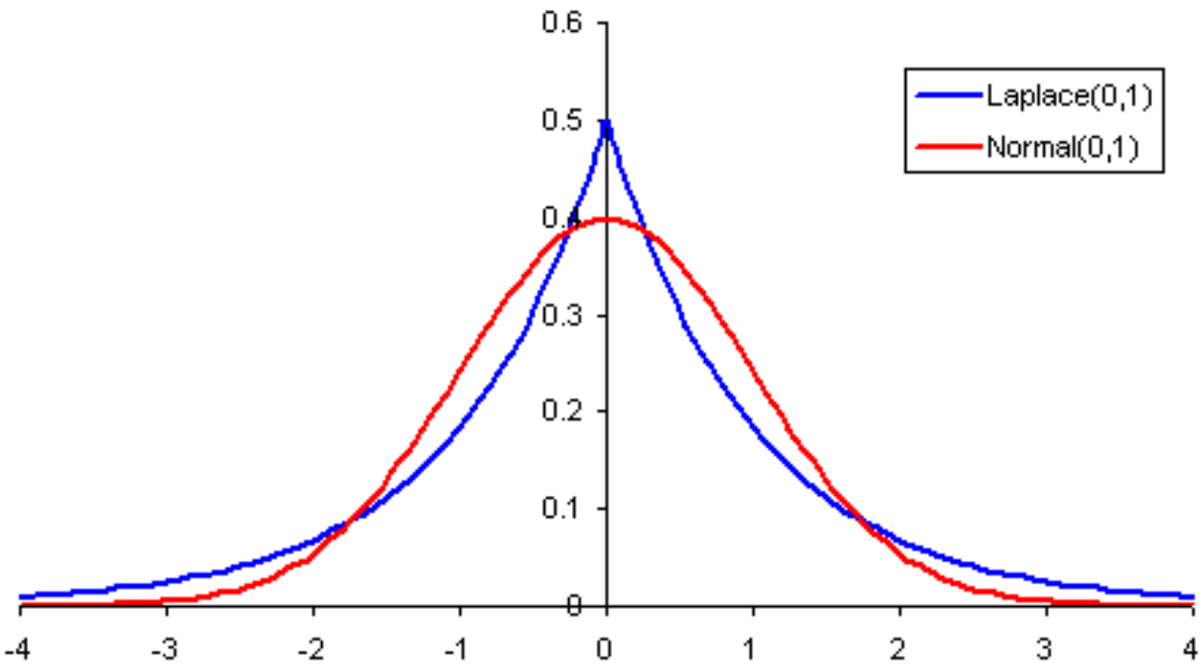


Figure 4.5: A comparison between Gaussian and Laplace priors. The Gaussian prior prefers the values to be near zero, whereas the Laplace prior more strongly prefers the values to equal zero.

Regularization intuition

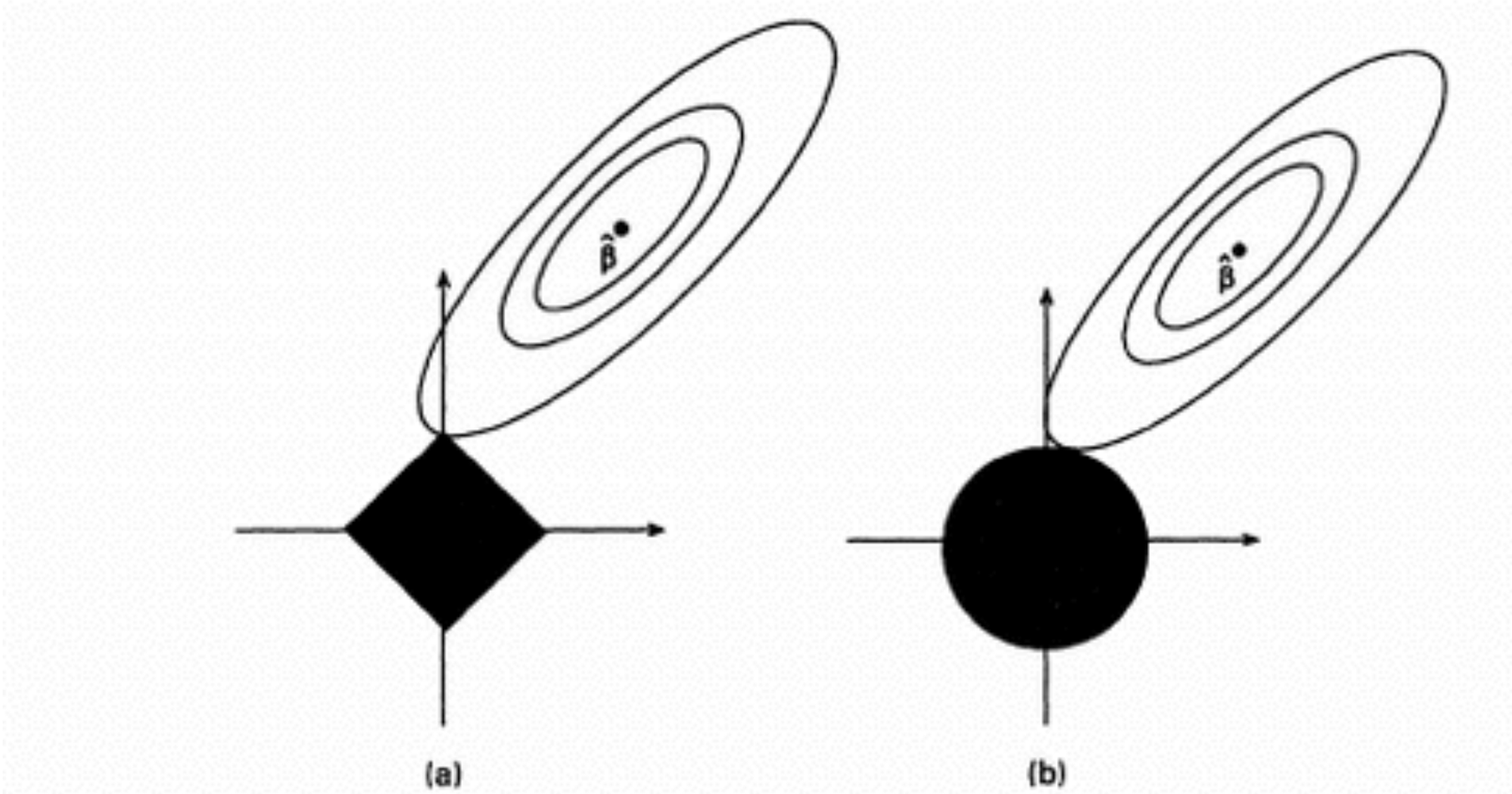


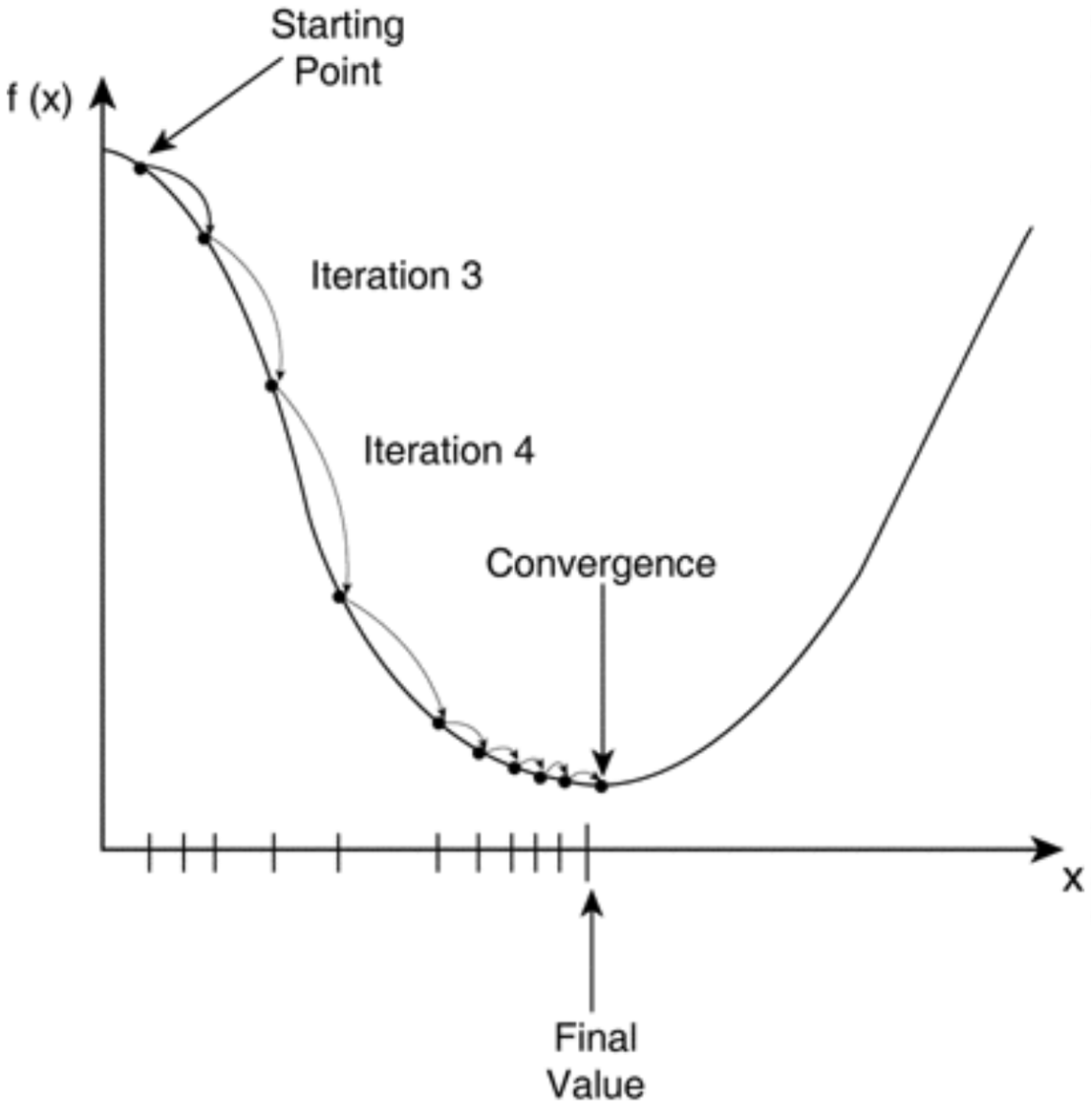
Fig. 2. Estimation picture for (a) the lasso and (b) ridge regression

Whiteboard

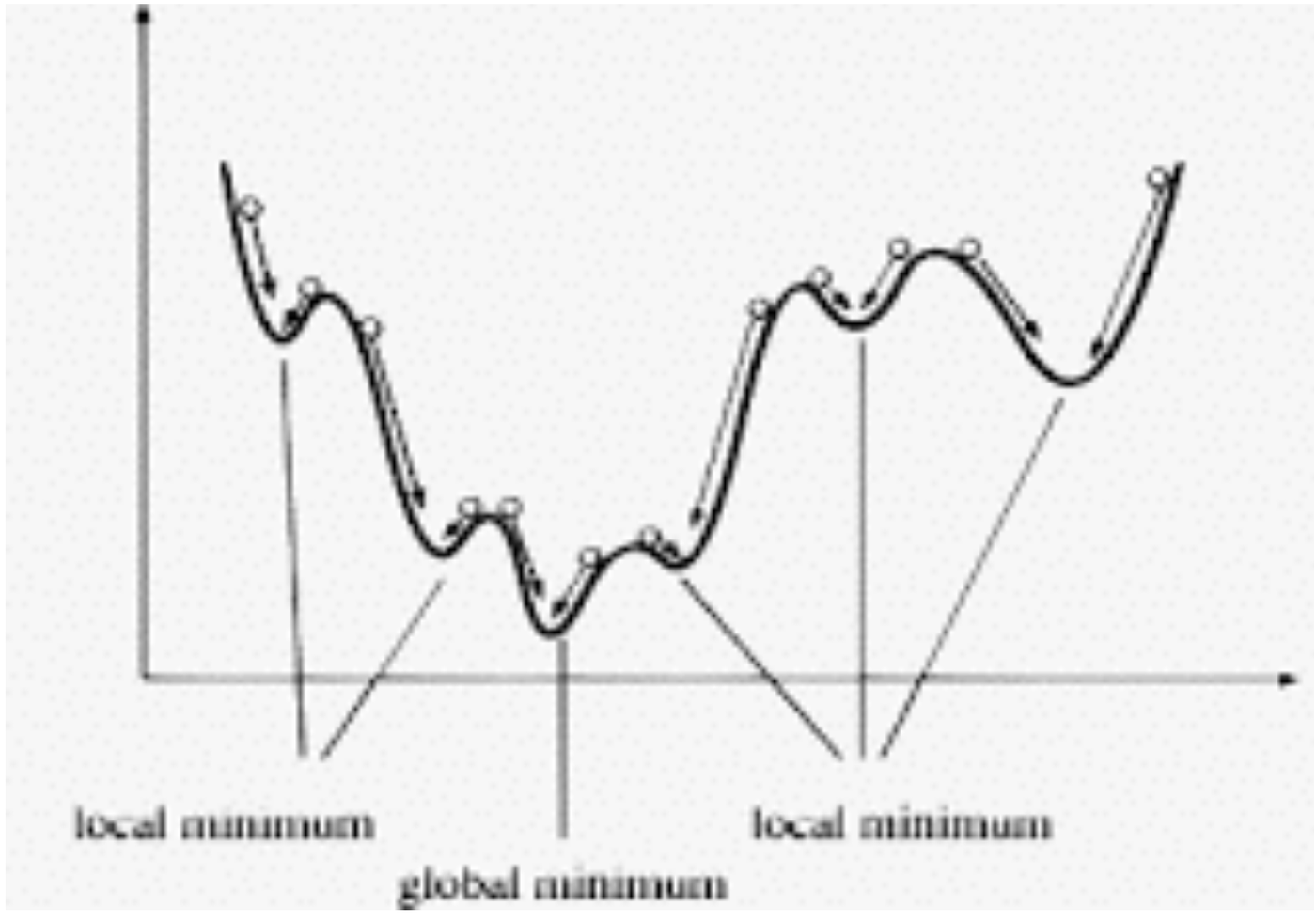
- Stability analysis with SVD
  - including bias-variance of solution
- Stochastic optimization for big datasets

SVD intuition

Gradient descent intuition



Convex function



Non-convex function