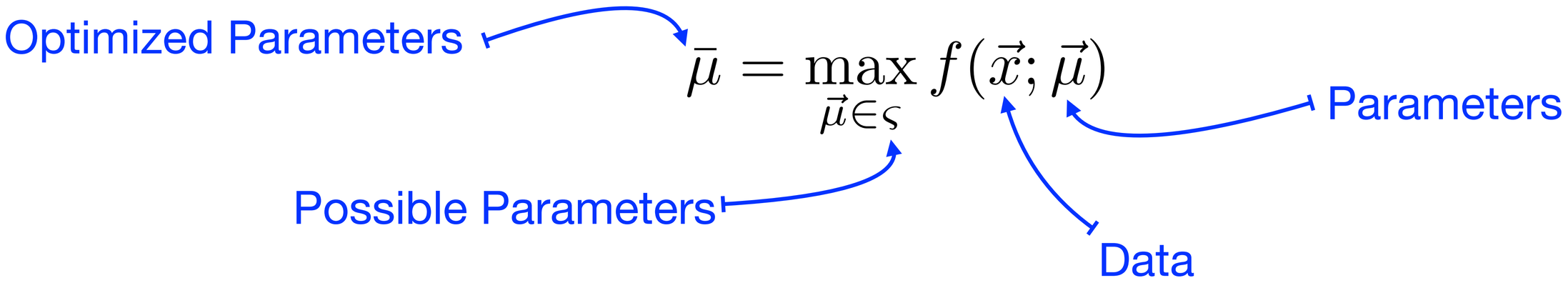
Optimization Overview

General Observations

- In a sense, we humans are constantly trying to optimize things
- Microeconomics tries to capture this
- In certain industries, especially with copious data, the work can become more mathematical, statistical and rigorous
- Even in these convenient cases, the mathematics and computer science is tricky

Optimization



Two Essential Questions

$$\bar{\mu} = \max_{\vec{\mu} \in \varsigma} f(\vec{x}; \vec{\mu})$$

- What *f* are we trying to optimize?
- How will we search for an optimum?

Trivial Example

$$\bar{\mu} = \min_{\mu \in \mathbb{R}} \sum_{i=1}^{N} f(\vec{x}_i; \vec{\mu})_\mu |^2$$

Bonus Question: What if the square wasn't there?

- We can view the *mean* as an optimization problem
- What value of μ minimizes the sum of squared distances to our data?
- Simple calculus obtains our answer

$$\frac{d}{d\mu} \sum_{i=1}^{N} (x_i - \mu)^2 = -2 \sum_{i=1}^{N} (x_i - \mu) = 0$$

$$\sum_{i=1}^{N} (x_i - \mu) = 0 \implies \mu = \frac{1}{N} \sum_{i=1}^{N} x_i$$

Not So Trivial Example

$$\hat{\beta} = \min_{\vec{\beta} \in \mathbb{R}^K} \sum_{i=1}^{N} \left(x_i^0 - f(\vec{x}; \vec{\beta}) \cdot \vec{x}_i^{[0]} \right)^2$$

- We can also view *linear regression* as an optimization problem
- What value of $\vec{\beta}$ minimizes the sum of squared distances to our data?
- After some symbol-jiggling, we find ourselves in the domain of linear algebra

$$\hat{\beta} = (X^* X)^{-1} X^* Y$$

$x^{[0]}$, independent variables

x^0, dependent variables

Business as Optimization

- Ultimately, nearly any aspect of business can be viewed as a problem in microeconomics, via constrained optimization of choice mechanisms
    - Office size and location
    - Inventory size and replenishment schedule
    - Advertising campaign style
- We consider these choices to be within the domain of mathematical finance in a more specific set of cases:
    - Market data and a population of market prices
    - Similar objective functions among disparate entities
    - Explicit or near-explicit treatment of errors and randomness
    - Algorithms and computability

Economic Risk and Reward

- Economically we want good outcomes *g*, with little risk of things going wrong *w*
- Notionally, a linear equation would look like finding an optimum of

$$c_1 g - c_2 w$$

- We might be inclined to define *g* as expected outcome, and *w* as its standard deviation
- The mathematical economics is more convenient if we use variance instead, and divide both constants by c_1

$$g - \lambda \sigma_g^2$$

- A Taylor expansion near the optimum shows that these are equivalent up to constants

Portfolio Optimization

- This concept suggests optimization in a common case: allocating resources with weights $\overrightarrow{w}$ among many correlated uses (*Kantorovich*)
- The continuous weight single-period version with gaussian outcomes described by μ, Σ yields

$$g - \lambda\sigma^2 = w^* \cdot \vec{\mu} - \lambda w^* \cdot \Sigma \cdot w$$

- This simple version has a solution $\overrightarrow{w} = \frac{1}{\lambda}\Sigma^{-1}\mu$, but if we introduce higher statistical moments, constraints, discreteness, or cointegration that is not correlation, it becomes a more generic optimization problem

Parameter Fitting

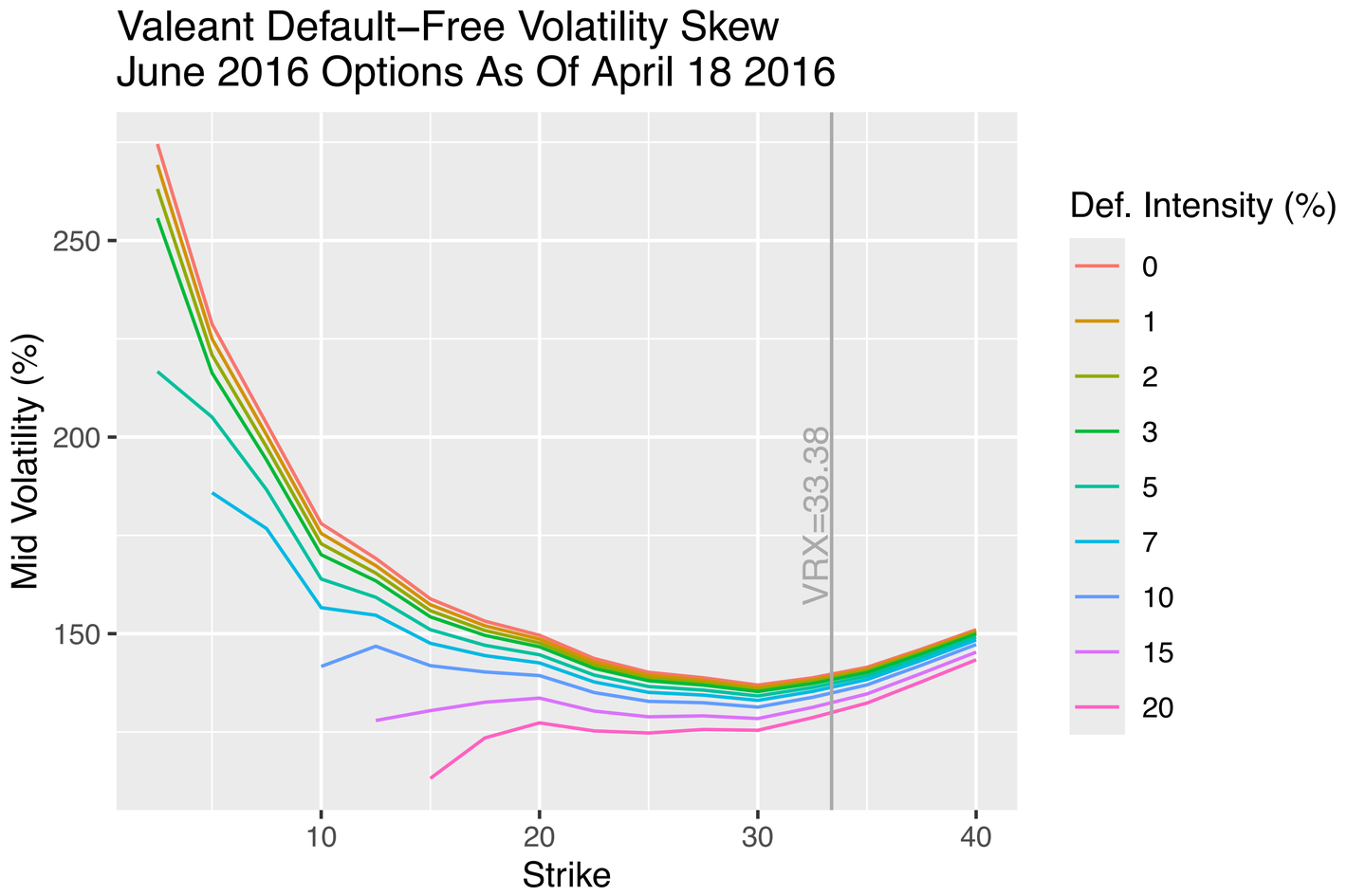
- It is common to represent a large data set by means of a fitted model of its elements
- Simplest version: describing a whole dataset by its mean.
- We also saw this with linear regression, where we optimized the fit according to a sum of squared errors. Here the $\vec{\beta}$ summarizes the dataset.
- In these cases, the objective function *f* is not directly economic
- Squares in the objective function terms are common, due to their convenient behavior with respect to computing derivatives (gradients)

Classic Parameter Fitting: Black-Scholes Skew

- In derivatives pricing, we have a "classical" model where, given a set of assumptions, we can write some asset prices $P(K)$ indexed by *K* as $BS(K, \sigma(K, \vec{\mu}))$ where this function is fairly nonlinear
- We might take $\sigma(K, \vec{\mu})$ to be parabolic in which case it has 3 parameters, or a spline with a few more, or something even more complex
- We also have *N* observed bid prices $B_i = B(K_i)$, ask prices $A_i = A(K_i)$, and primary weights w_i
- We end up with a fairly complex optimization to perform, not least because there are constraints on permissible values of $\vec{\mu}$

$$\bar{\mu} = \min_{\vec{\mu} \in \varsigma} \sum_{i=1}^{N} \frac{w_i}{A_i - B_i} \left(BS(K_i, \sigma(K_i, \vec{\mu})) - \frac{1}{2}(B_i + A_i) \right)^2$$

Example: Parameter Effects



Fitted model outputs by default intensity. Note how changing the intensity parameter alters predictions across the entire strike scale. Only an uncommonly specific set of curves can be precisely reproduced by this mathematical construction.

High Parameter Dimensionality

- Once we find ourselves using modern machine learning, parameter counts increase vastly
  - Decision tree ensembles (random forests, boosted trees)
  - Neural networks
- The calculus and matrix algebra can become unwieldy
- This leads directly to the question of just *how* we will optimize

Layer (type)	Output Shape	Param #	Connected to
California (InputLayer)	(None, 8)	0	–
dense_16 (Dense)	(None, 25)	225	California[0][0]
dropout_14 (Dropout)	(None, 25)	0	dense_16[0][0]
leaky_re_lu_14 (LeakyReLU)	(None, 25)	0	dropout_14[0][0]
dense_17 (Dense)	(None, 25)	650	leaky_re_lu_14[0…
dropout_15 (Dropout)	(None, 25)	0	dense_17[0][0]
leaky_re_lu_15 (LeakyReLU)	(None, 25)	0	dropout_15[0][0]
dense_18 (Dense)	(None, 25)	650	leaky_re_lu_15[0…
dropout_16 (Dropout)	(None, 25)	0	dense_18[0][0]
leaky_re_lu_16 (LeakyReLU)	(None, 25)	0	dropout_16[0][0]
dense_19 (Dense)	(None, 25)	650	leaky_re_lu_16[0…
dropout_17 (Dropout)	(None, 25)	0	dense_19[0][0]
leaky_re_lu_17 (LeakyReLU)	(None, 25)	0	dropout_17[0][0]
dense_20 (Dense)	(None, 25)	650	leaky_re_lu_17[0…
dropout_18 (Dropout)	(None, 25)	0	dense_20[0][0]
leaky_re_lu_18 (LeakyReLU)	(None, 25)	0	dropout_18[0][0]
dense_21 (Dense)	(None, 25)	650	leaky_re_lu_18[0…
dropout_19 (Dropout)	(None, 25)	0	dense_21[0][0]
leaky_re_lu_19 (LeakyReLU)	(None, 25)	0	dropout_19[0][0]
dense_22 (Dense)	(None, 25)	650	leaky_re_lu_19[0…
dropout_20 (Dropout)	(None, 25)	0	dense_22[0][0]
leaky_re_lu_20 (LeakyReLU)	(None, 25)	0	dropout_20[0][0]
dense_23 (Dense)	(None, 1)	26	leaky_re_lu_20[0…
dense_24 (Dense)	(None, 1)	26	leaky_re_lu_20[0…
concatenate_1 (Concatenate)	(None, 2)	0	dense_23[0][0], dense_24[0][0]

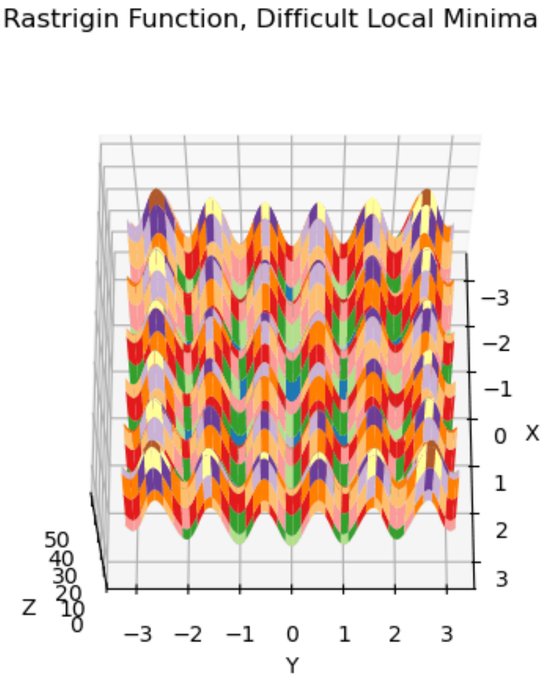
Fitting: Difficulty

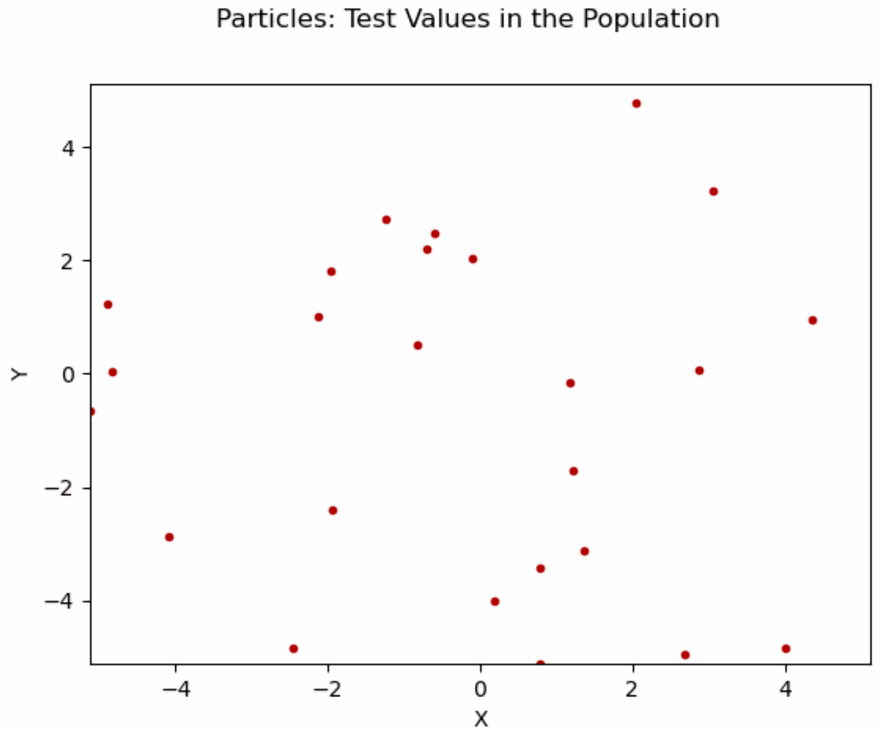
- We have talked about objective functions to fit, but aside from near-trivial cases, not yet about *how* to fit them
- In general, optimization problems are "hard" according to a rigorous definition from computer science
- **Worse, even the cases that are "easy" in a computer-scientific sense are computationally devilish in practice**
- Among the biggest of issues: computing $f()$ or its gradients can cost a lot of resources. We have to be parsimonious with our calculations
- This puts us in the domain of *economic* optimization: optimal choice in constructing our optimization runs themselves

Fitting: Taxonomy of Techniques

- Naive: grid searches
- When we have continuous variables, we can apply the methods of calculus
  - Newton's method and quasi-Newtonian methods, secants, BFGS
  - Seeking near-zero gradients: gradient descent, coordinate descent
- For trickier cases we fall back to some other choices
  - "Trying stuff out" via Monte Carlo and particle filter methods
  - Optimization of proxies: Gaussian processes, fitting on simplified interpolations of our objective

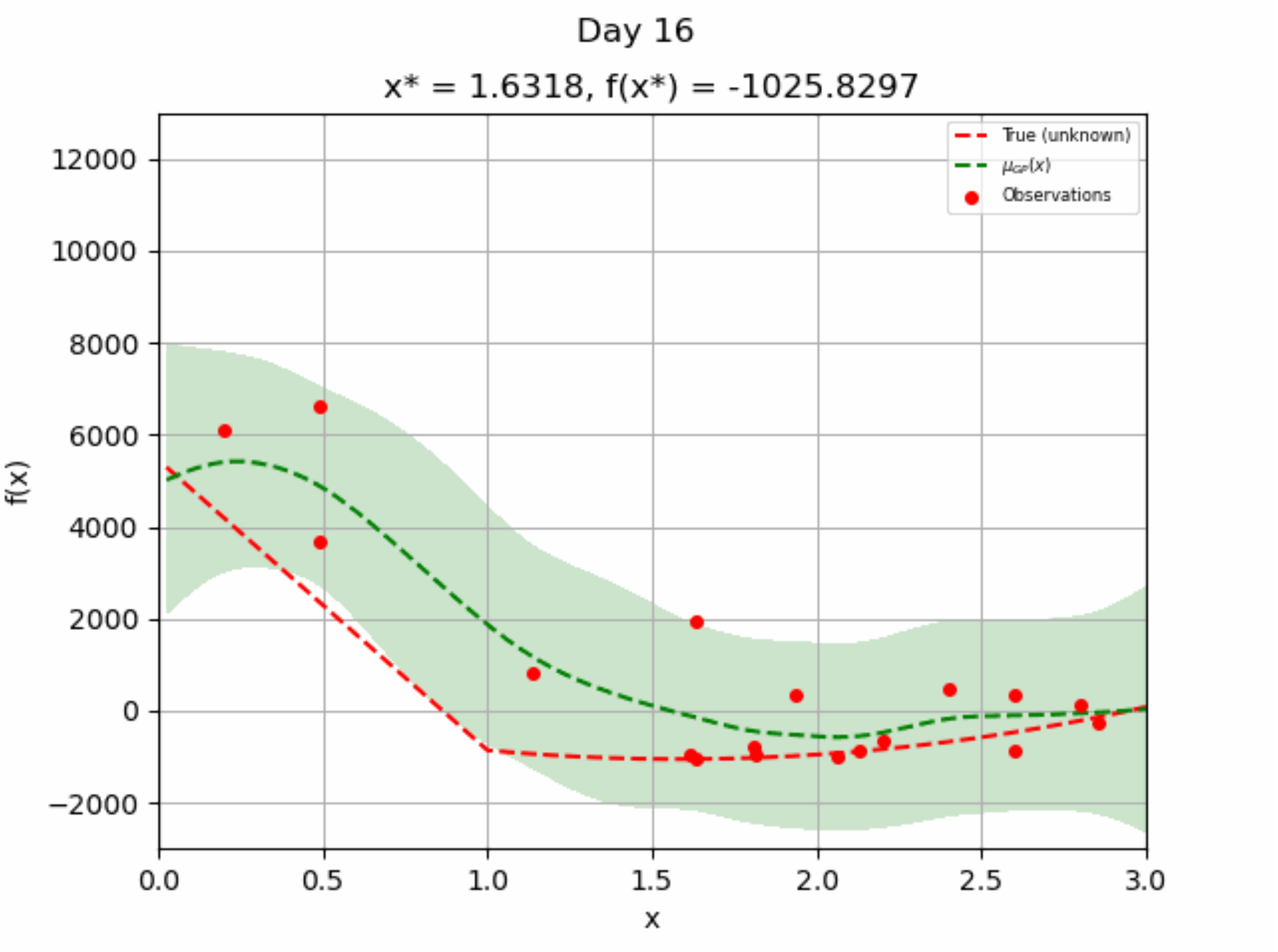
Particle Filters For Tricky Cases





Gaussian Processes For Expensive Experiments

- Bayesian optimization by gaussian process (and their relatives)
- Can maximize expected value or minimize "regret"
- Here, we show a toy experiment
- Negative PL versus investment threshold *x*



Cautionary Tale: Economies Change

- As market structure changes, our objective function changes with it
- We start out with a clear global minimum near 0.7 on day 1
- By day 4 it is a local, not a global, minimum
- By day 7 the only local minimum is near 0.3
- Yet, no daily change in shape was large