Introduction to modeling

By Marcelo Mattar
http://mattarlab.com

Part I: What are models?

Let's consider 3 very different models:

- Cosine tuning model of neuronal activity
- Hodgkin & Huxley single-neuron spike generation model
- Reinforcement learning model 😍

What are the many reasons why they are helpful?

Cosine tuning (CT)

$$\frac{f(s) - f_0}{f_{max}} = \cos(s - s_p)$$

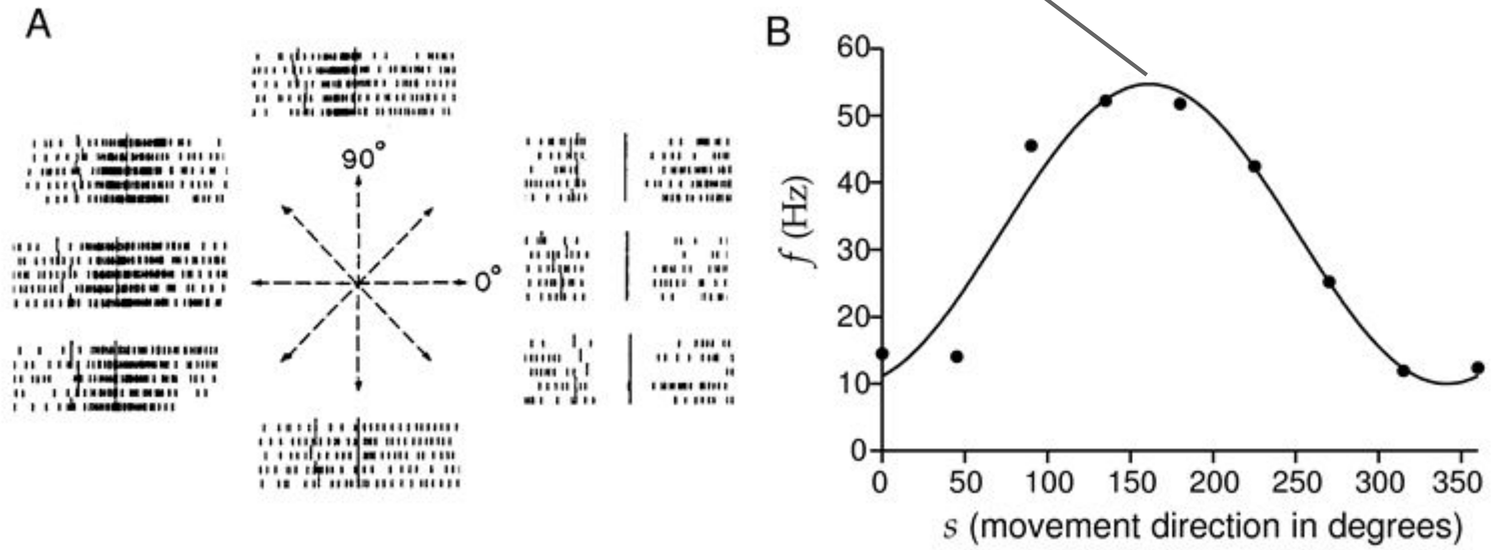


Figure from Dayan & Abbott, 2001

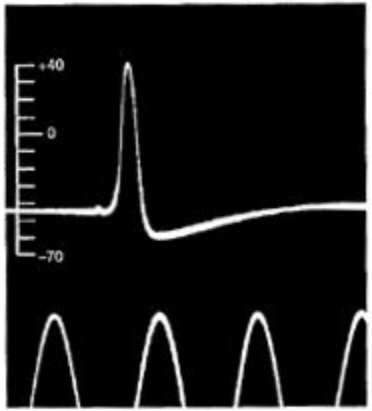
Why is CT helpful?

- It provides a really nice compact summary (**description**) of data
- It generalizes across movements
- We can use this for applications, e.g. brain-machine interfaces (prosthetics)

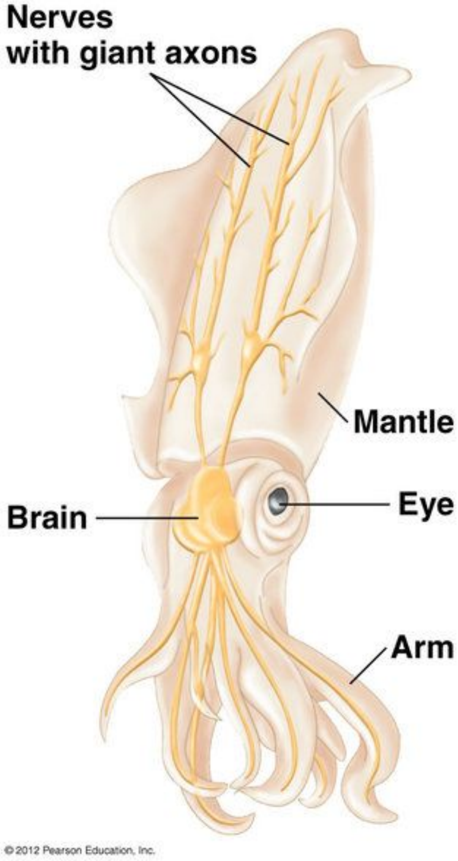
But…

- It's purely descriptive
- No consideration **how/why** cosine tuning arises
- Thus, provides limited insight

Hodgkin & Huxley (HH)



Hodgkin & Huxley, 1939



Longfin Inshore Squid (loligo pealeii)

https://www.scirp.org/journal/paperinformation.aspx?paperid=23595

Hodgkin & Huxley

$$C \cdot \frac{dV}{dt} = -g_K n^4 \cdot (V - E_K) - g_{Na} m^3 h \cdot (V - E_{Na}) - g_L \cdot (V - E_L) + I(t)$$

Electrochemical gradients

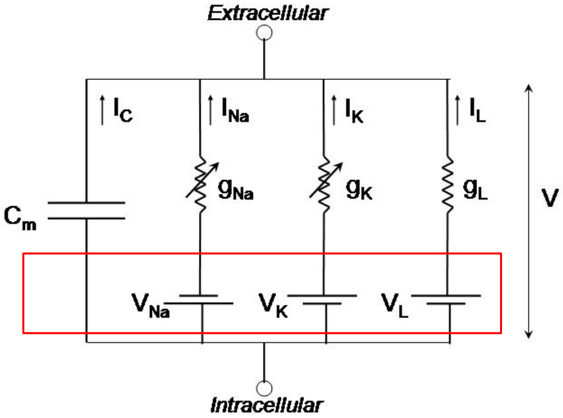


Figure from scholarpedia

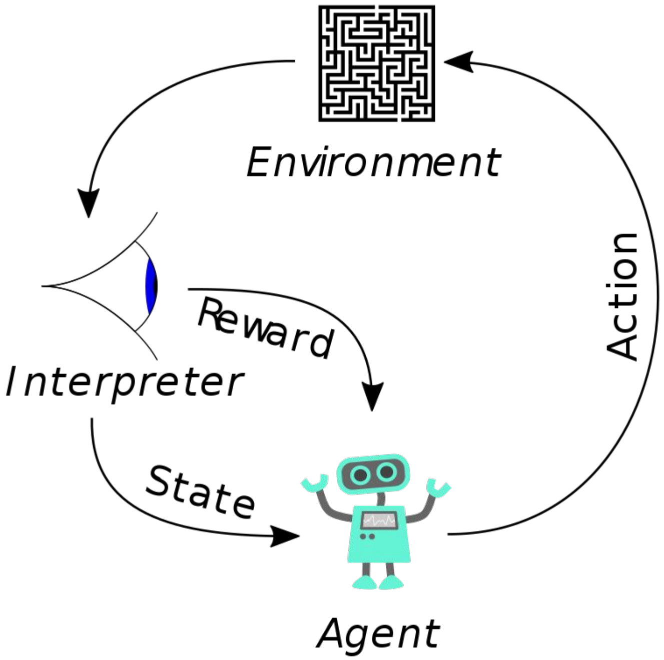
Why is HH helpful?

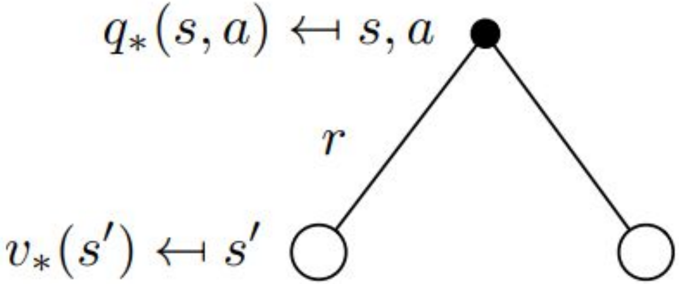
- Provides a **mechanism** for the generation of action potentials
- Synthesizes large amounts of neural data
- Describes variables that are not easily measurable (latent)
  - E.g. Channel opening/closing, currents for individual ions
- Allows for studying effect of interventions
  - E.g. How would a K channel blocker affect spiking?
- You can make real predictions
  - E.g. conditions controlling timing of action potential onset (threshold, refractory period)

But…

- No idea **why** ion channels open/close (molecular mechanism)

Reinforcement learning (RL)





$$q_*(s,a) = \mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a v_*(s')$$

Why is RL helpful?

- Provides a **normative** benchmark of what is best in theory
  - Including predictions for optimal behavior (under given assumptions)
- Allows to synthesize large amounts of behavioral and neural data
- Describes variables that are not easily measurable (latent)
  - E.g. reward prediction error
- Has inspired new technology
  - E.g. RL is central to modern AI

SUMMARY - Why models are great?

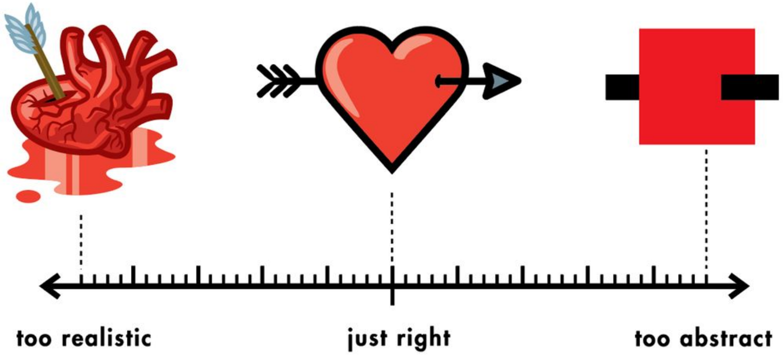
- Knowledge synthesis (CT, HH, RL)
- Identifying hidden assumptions, hypotheses, unknowns (HH)
- Mechanistic insights (HH)
- Retrieve latent information (HH, RL)
- A testbench for medical interventions (HH)
- A testbench for behavioral interventions (RL)
- Guidance in designing experiments ➔ quantitative predictions (HH, RL)
- Inspire new technologies (CT, RL)

What are models?

Models are an abstraction of reality!

Models are partial, imperfect descriptions of the universe, developed by science for our understanding of the universe that is otherwise too complex to grasp by the limits of the human mind…

Rosenblueth & Wiener (1945)

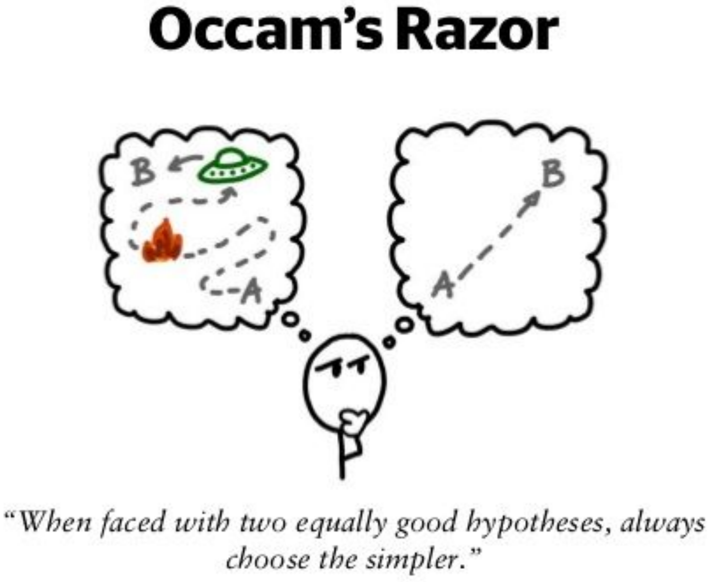


https://computersciencewiki.org/index.php/Abstraction

What are models?

How do we find the right level of abstraction?

- Keep it as simple as possible, but as detailed as needed
- This is determined by our question, hypotheses and model goals! ➜ more during day 2



What are models?

Models allow for **understanding** and **control** (Rosenblueth & Wiener, 1945)

Insights not directly accessible by experiments / data

Interventions, e.g. experimental, clinical

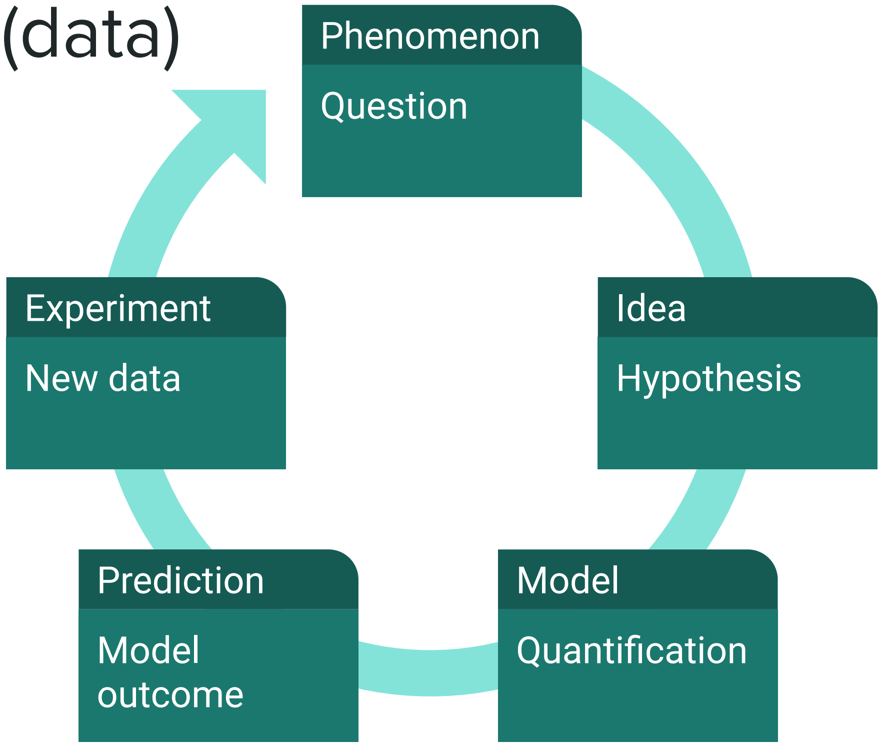
Requires model validation ➜ experiments!

In other words, a model is a **Hypothesis**!

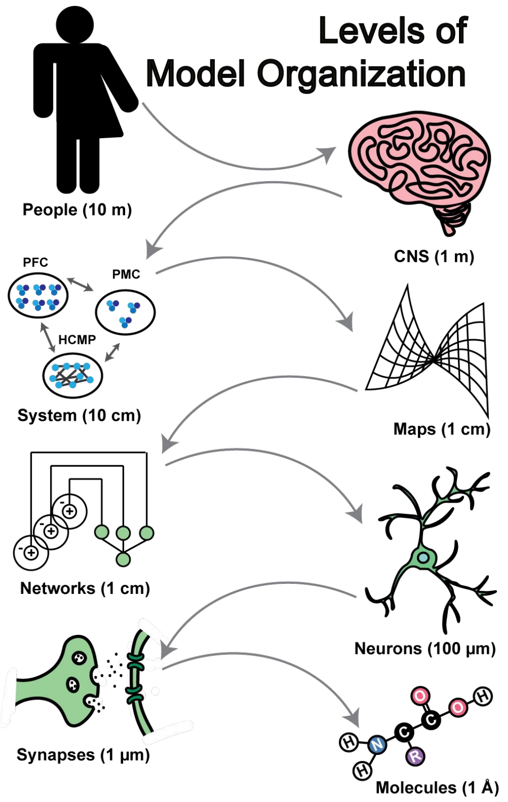
Models & experiments (data)

Cycle of discovery

- Models inspire experiments
- Data constrain models & provide new questions

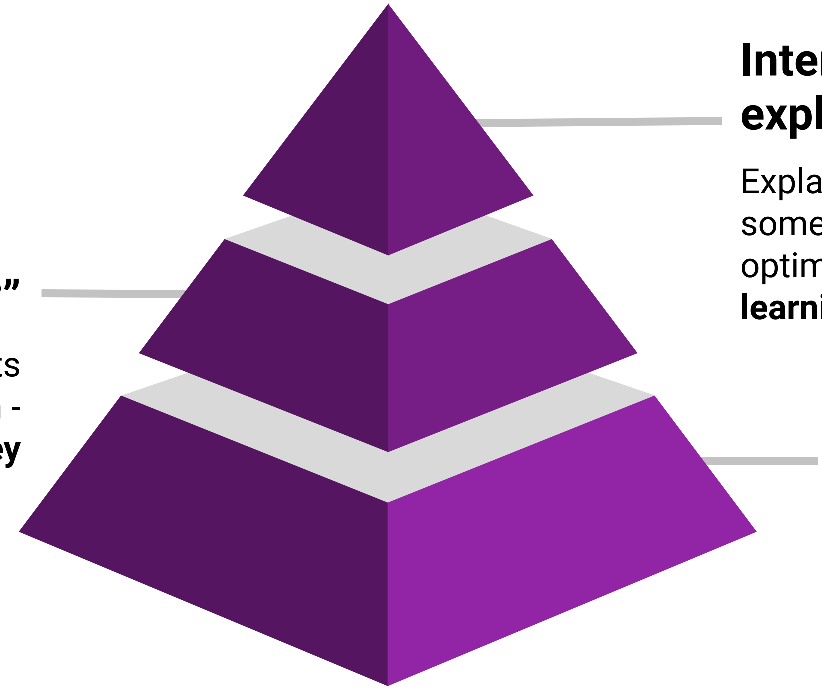


Classifying models



Trappenberg (2009), *Fundamentals of Computational Neuroscience*

The model universe



Interpretive / explanatory - “why?”

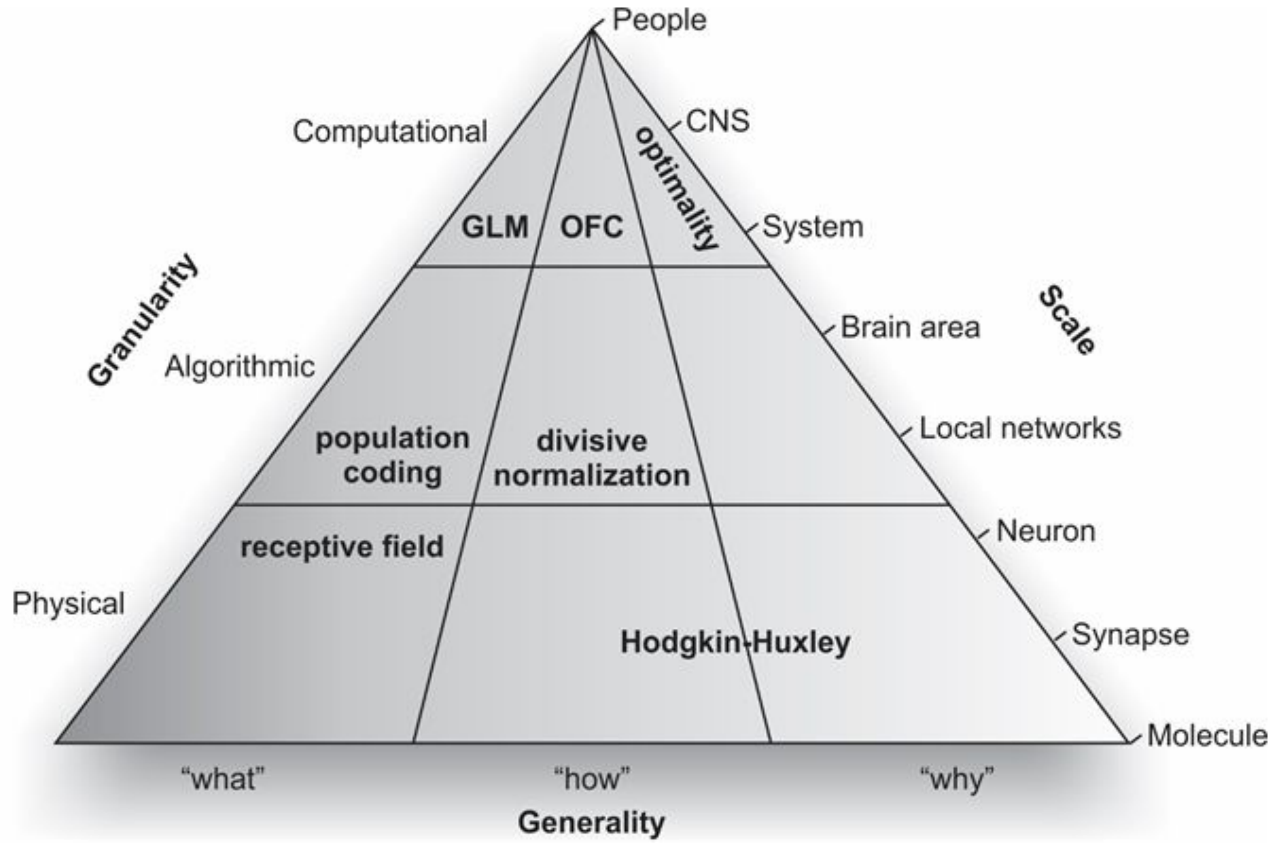
Explain why the brain does something, e.g. because it’s optimal - **e.g. Reinforcement learning**

Mechanistic - “how?”

Show how neural circuits perform complex function - **e.g. Hodgkin & Huxley**

Descriptive - “what?”

Compact summary of large amounts of data - **e.g. Cosyne tuning**



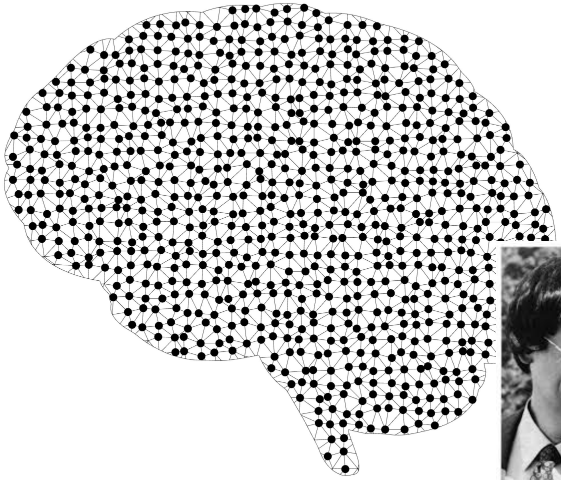
Model classifications

Different models help answer **different, parallel questions**

Marr’s 3 levels of analysis

Brain: hierarchy of complexities

- (“What” is outside of Hierarchy)
- Computational level - 1
    - What is the **objective** of the system?
    - How close is it to **optimal**?
- Algorithmic level - 2
    - What are the **data structures**?
    - What are the **approximations**?
    - What is the **runtime**?
- Implementation level - 3
    - What is the **hardware**?
        - Neurons? Synapses? Molecules?



David Marr (1945-1980)

image credit: Wikipedia

Summary

- Different models allow answering different questions
- Everyone has a different idea of what’s a “good” model

- “Good” is what best answers the question with minimal assumptions and highest explanatory power
- Thus **model diversity** is good!

Part II: Model fitting

Two important questions in modeling

1) Models have parameters

How should we set those, and how can we understand our uncertainty about them?

2) We have multiple models

Which models explain reality better?

Arguably almost all of neuroscience is about finding good models

Fitting and choosing models

Fitting models

- Why we fit models
- Linear models (a "what" model)

How to fit models

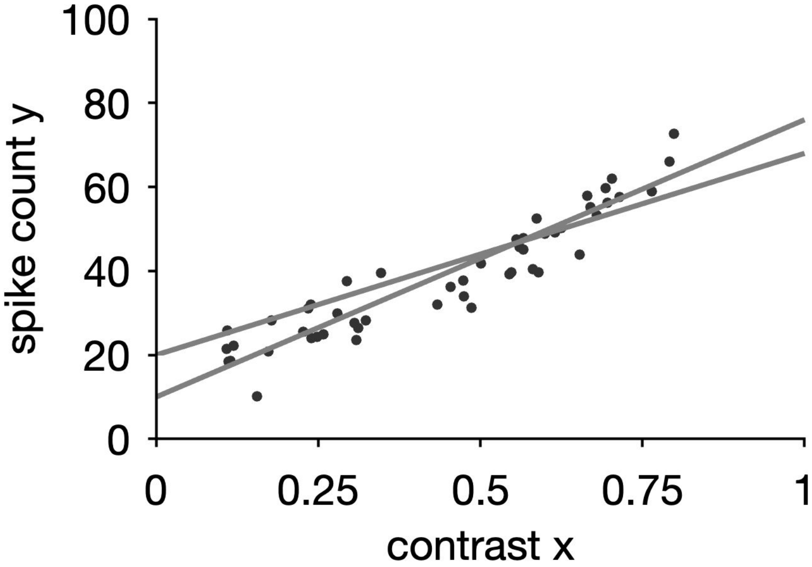
- Fitting models by minimizing errors, or by maximizing likelihood
- Duality between minimizing squared error and maximizing Gaussian likelihood

Assessing model fits

- Bootstrapping to assess parameter uncertainty
- Comparing models

Why we fit models & linear models

A simple linear model ("what" model)



Simple model

spike count $\sim$ increases linearly with contrast

$$y \approx \theta_0 + \theta_1 x$$

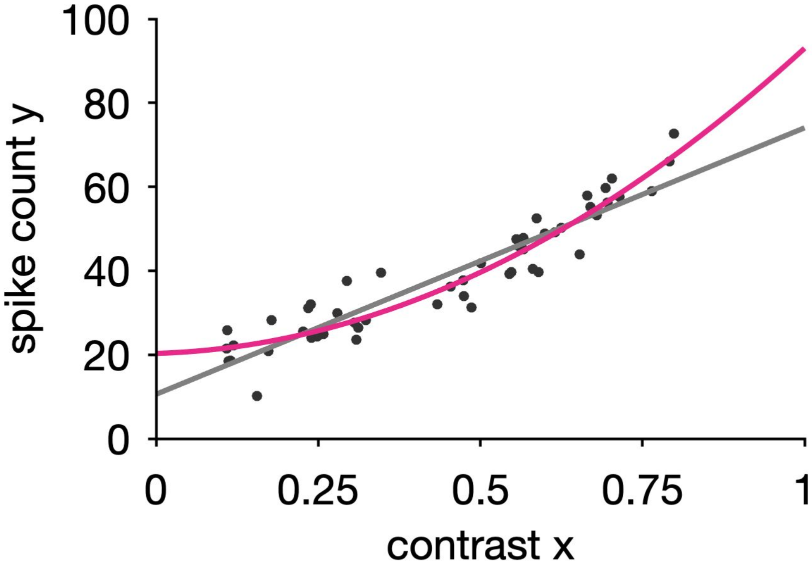
intercept slope

What is the best set of parameters?

How do we measure goodness-of-fit?

How do we find the best-fitting parameters?

Purpose of model fitting



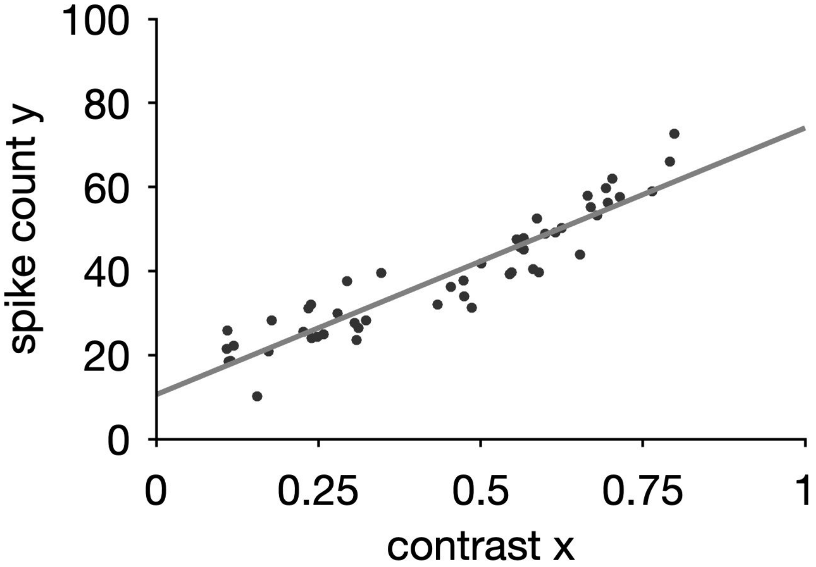
Validation: generate new data
check on held-out data

Prediction: behavior outside of data

Interpret: e.g., spike count ~ contrast? ($\theta_0 \neq 0$?)
(aka no intercept; simple models only)

Compare: fits across different models

Linear model can be more complex



spike count ~ increases linearly with contrast

$$y \approx \theta_0 + \theta_1 x$$

intercept slope

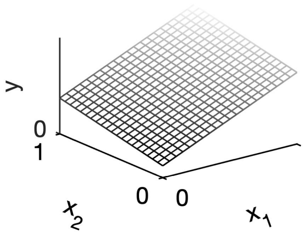
Linear models in general

Assume multiple inputs, one for each stimulus feature (e.g., orientation, contrast, etc.)

$$\boldsymbol{x} = (x_1, x_2, \dots)^T$$

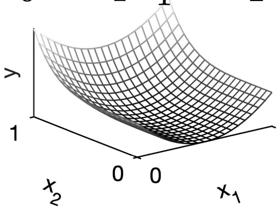
(Simple) linear model
defines (hyper)plane in ***x***

$$y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \dots$$



Can be non-linear in inputs

e.g., $y = \theta_0 + \theta_1 x_1^2 + \theta_2 x_2^4 + \dots$



More generally,

$$y = \sum_i \theta_i \phi_i(\boldsymbol{x}) = \boldsymbol{\theta}^T \boldsymbol{\phi}(\boldsymbol{x})$$

linear in parameters ***θ***,
not (necessarily) inputs ***x***

$$\boldsymbol{\phi}(\boldsymbol{x}) = \begin{pmatrix} 1 \\ \phi_1(\boldsymbol{x}) \\ \phi_2(\boldsymbol{x}) \\ \vdots \end{pmatrix}$$

How to fit models

Two philosophies for fitting models (linear & otherwise)

Models as functions

$$y = f\left(x; \theta\right)$$

Aim: find model with small errors

noise from some distribution

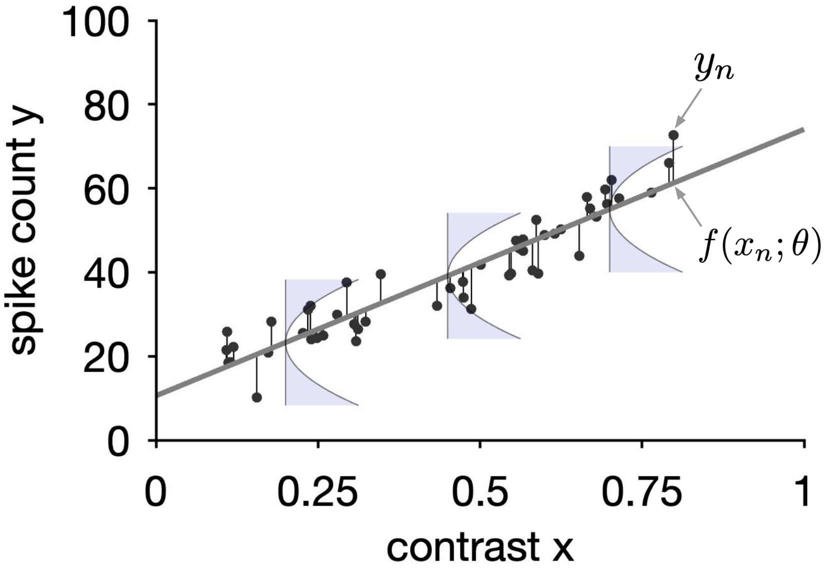
Models as generators

$$y_{\text{measured}} = f\left(x; \theta\right) + \eta$$

Aim: find model that assigns high probability to the data

Supports richer set of statements about models!

Fitting models by minimizing squared errors



Mean squared error (MSE)

Average squared difference between data and model prediction

$$\mathrm{MSE}(\theta) = \frac{1}{N}\sum_{n=1}^{N}\left(y_n - f(x_n;\theta)\right)^2$$

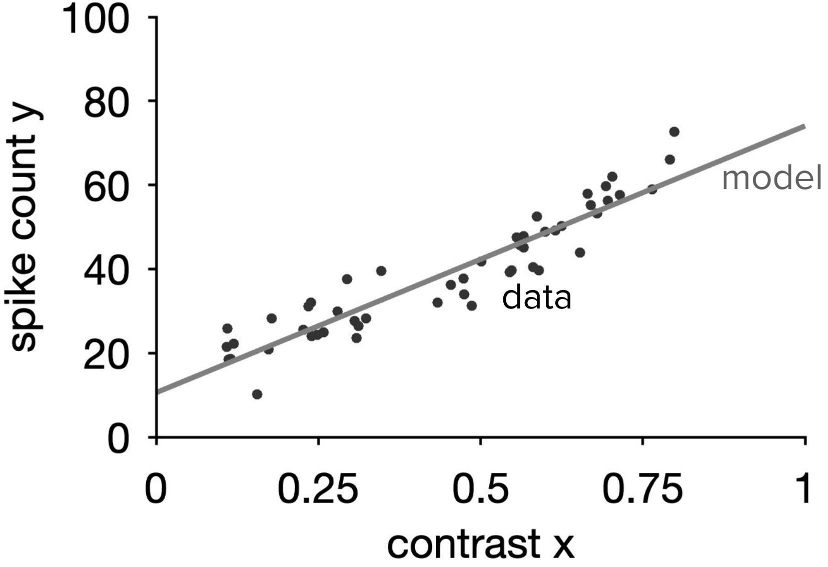
measured

model prediction

Best-fitting parameters

$$\hat{\theta}_{\mathrm{MSE}} = \operatorname*{argmin}_{\theta} \mathrm{MSE}(\theta)$$

Generative perspective on model fitting



Generative perspective
Model assumed to "generate" observed data

data $\sim$ model prediction + noise

what we can't control
(e.g., measurement noise)

what we don't care about
(e.g., deviation from mean firing rate)

Likelihood function

$$p\left(\text{data}|\text{parameters }\theta\right) = \mathcal{L}\left(\theta|\text{data}\right)$$

"How likely is data for given parameters?"

Fitting models by maximum likelihood

Aim of maximum likelihood (ML) fits
Find parameters that make data most likely

$$\hat{\theta}_{\mathrm{ML}} = \underset{\theta}{\operatorname{argmax}} \, \mathcal{L}\left(\theta|\mathrm{data}\right) = \underset{\theta}{\operatorname{argmax}} \log \mathcal{L}\left(\theta|\mathrm{data}\right)$$

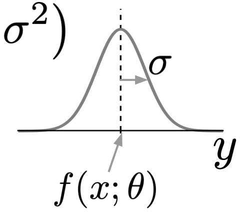
ML for independent trials
If trials are independent, then $\mathcal{L}\left(\theta|\mathrm{data}\right) = \prod_n \mathcal{L}\left(\theta|\mathrm{data}_n\right)$
As a result,

$$\hat{\theta}_{\mathrm{ML}} = \underset{\theta}{\operatorname{argmax}} \prod_n \mathcal{L}\left(\theta|\mathrm{data}_n\right) = \underset{\theta}{\operatorname{argmax}} \sum_n \log \mathcal{L}\left(\theta|\mathrm{data}_n\right)$$

Example: Maximum likelihood with Gaussian noise

Gaussian noise with variance σ^2

$$y = f(x;\theta) + \eta \quad \Leftrightarrow \quad p(y|x,\theta) = \mathcal{L}(\theta|x,y) = \mathcal{N}\left(y|f(x;\theta), \sigma^2\right)$$



trials are independent

$$\log \mathcal{L}(\theta|X,Y) = \sum_n \log \mathcal{L}(\theta|x_n, y_n)$$

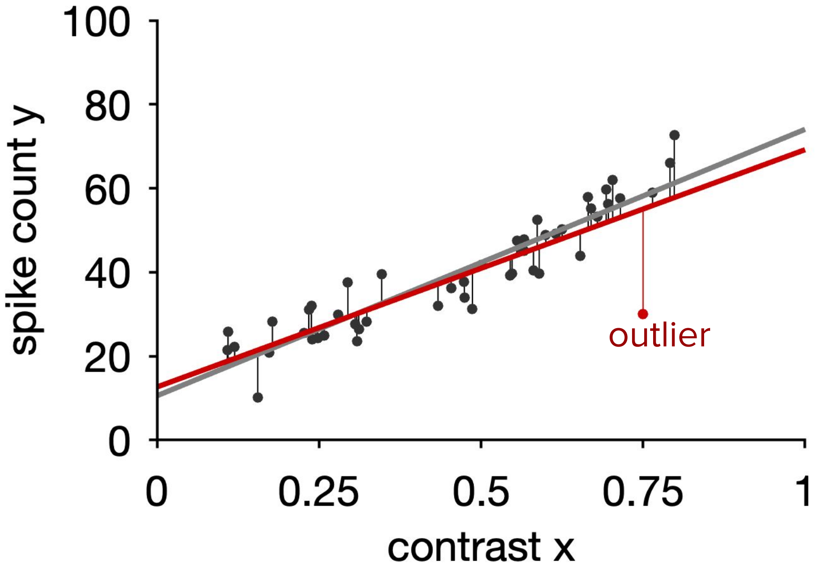
$$= -\frac{N}{2\sigma^2}\frac{1}{N}\sum_n (y_n - f(x_n;\theta))^2 + \text{const.} \quad = -\frac{N}{2\sigma^2}\boxed{\mathrm{MSE}(\theta)} + \text{const.}$$

linear model with Gaussian noise

independent of $\boldsymbol{\theta}$

maximizing likelihood with Gaussian noise = minimizing mean squared error

Gaussian noise: sensitivity to outliers



Gaussian noise: quadratic error function

- Larger errors weigh more strongly
- Fits sensitive to outliers

Fitting linear models

Linear model

$$y = f\left(\boldsymbol{x};\boldsymbol{\theta}\right) + \eta = \boldsymbol{\theta}^T \boldsymbol{\phi}\left(\boldsymbol{x}\right) + \eta$$

Log-likelihood with Gaussian noise

$$\log \mathcal{L}\left(\boldsymbol{\theta}|\boldsymbol{X},\boldsymbol{y}\right) = -\frac{N}{2\sigma^2}\frac{1}{N}\sum_n \left(y_n - \boldsymbol{\theta}^T \boldsymbol{\phi}\left(\boldsymbol{x}_n\right)\right)^2 + \text{const.}$$

Properties

- Single most important statistical ("what") model
- Likelihood quadratic in $\boldsymbol{\theta}$ (concave function) → easy to find best-fitting parameters
- Analytic expression for ML estimate (see tutorial)

What we have learned

Two philosophies for fitting models

Minimizing error
Maximizing likelihood

Minimizing mean squared error == maximizing likelihood with Gaussian noise

Squared error makes fit sensitive to outliers

Applied to linear model

Easy to find best-fitting parameters,
computable by analytical expression

Assessing model fits

Parameter uncertainty

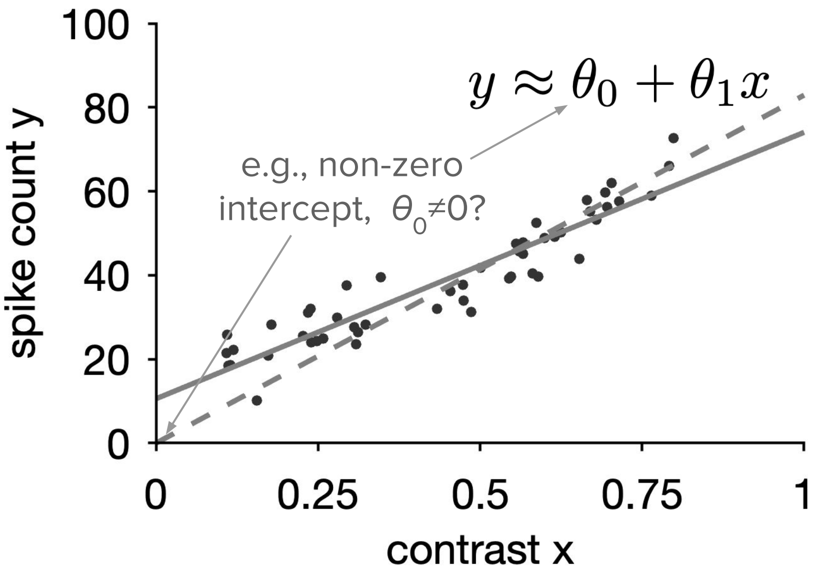
Limited data → multiple parameter values ***θ*** might explain the data about equally well. Reflects *inherent uncertainty* about best-fitting parameters.

Example uses

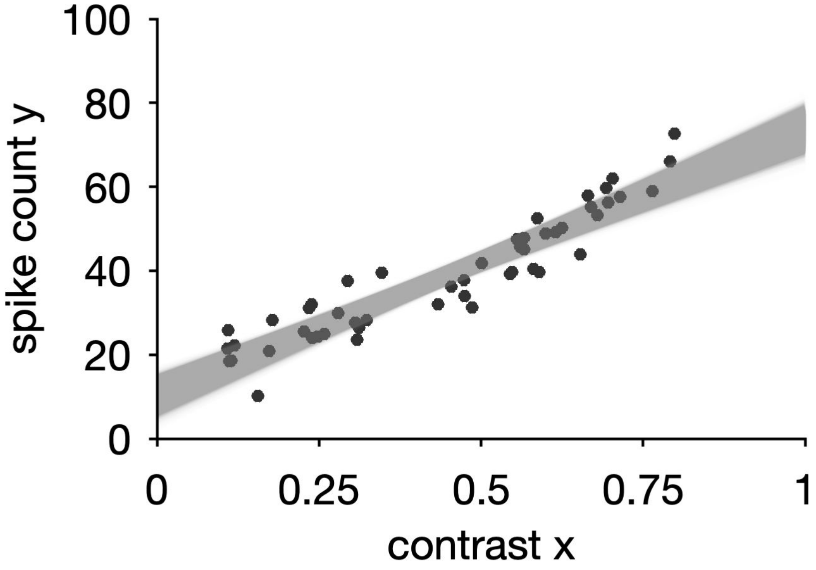
- How well does data constrain parameters?
- Are parameters significantly non-zero (i.e., relevant)?

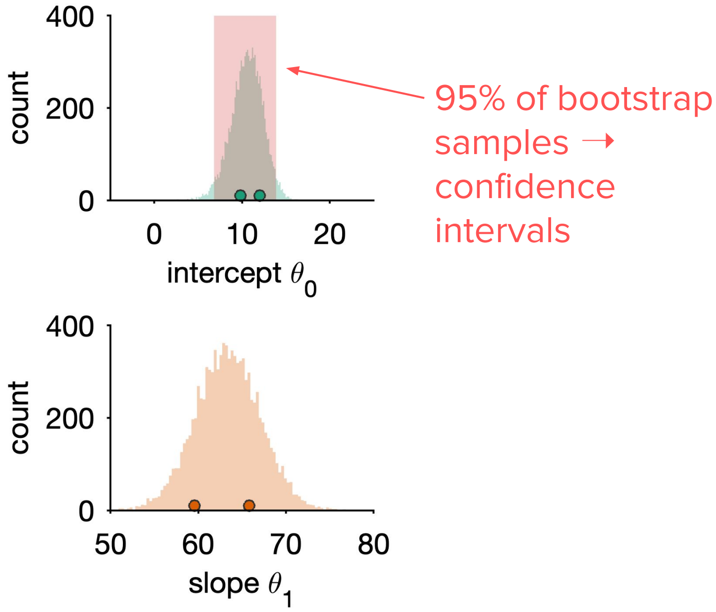
Linear models can assess uncertainty through standard statistics (not discussed further).

Generally can assess parameter uncertainty through *bootstrapping*.

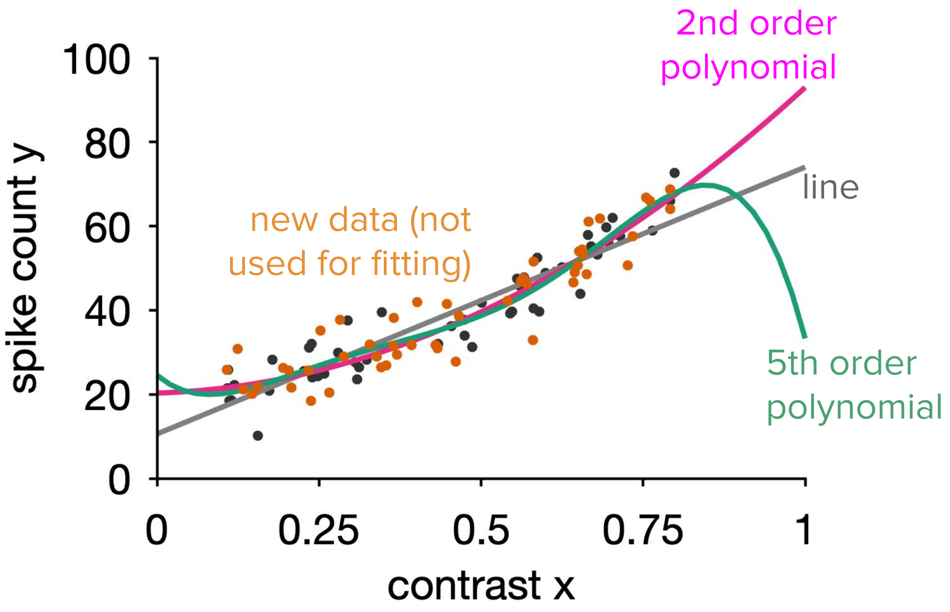


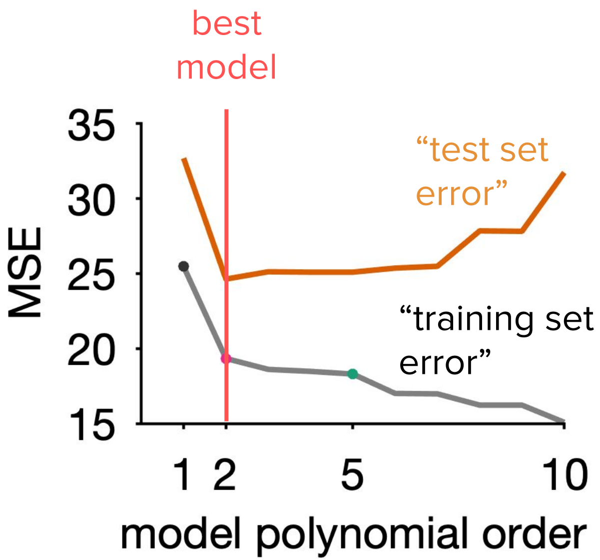
Assessing uncertainty by bootstrap



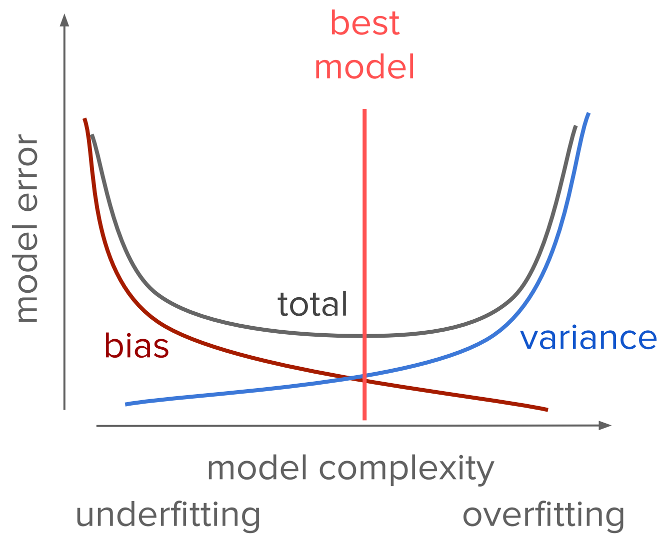


Fitting & comparing multiple models





Bias-variance trade-off



Bias
Low model complexity: systematic deviation from structure underlying data (underfitting)

Variance
High model complexity: capturing variability beyond the structure underlying data (i.e., noise; overfitting)

Total error = bias + variance

Best model: balances bias & variance

Two philosophies for comparing models

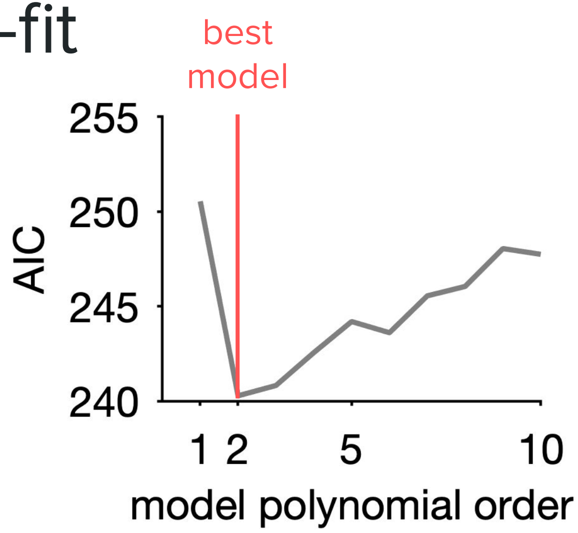
Goodness of fit (popular in statistics)	Compute likelihood of fitted model, and correct for number of parameters, compare goodness of fits. Good models use few parameters to produce good fits
Cross validation (popular in machine learning)	Fit model to some data (training set), then check how well it predicts new data (test set).

Model comparison by goodness-of-fit

Example: Akaike Information Criterion (AIC)
(lower is better)

$$\mathrm{AIC} = 2k - 2\log\mathcal{L}\left(\hat{\theta}_{\mathrm{ML}}|X,Y\right)$$

number of parameters



Pros Easy to compute

Cons Strong assumptions about the model's structure

Alternatives

Other information criteria: BIC / DIC / …, differ in how they measure model complexity

Bayesian model comparison: implicit complexity penalty by averaging over model parameters

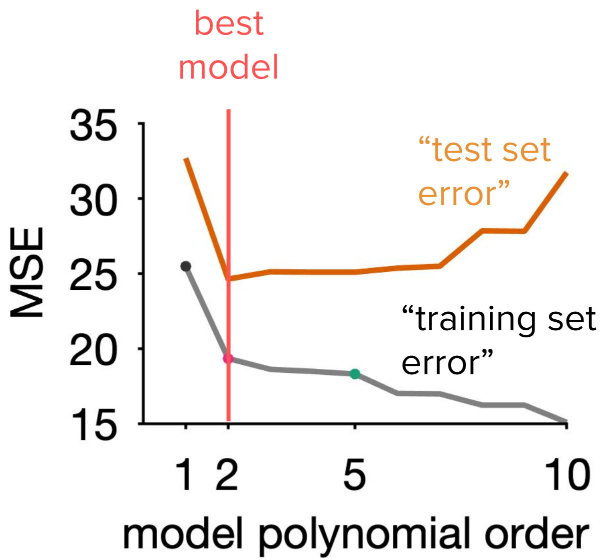
…

Model comparison by cross-validation

Compare models by prediction error on held-out data

Pros Minimal assumptions about data
Widely applicable

Cons Requires lots of data
Computationally expensive
Little sensitivity to small model differences



What we have learned

Limited data makes model parameters uncertain

Assessing uncertainty by bootstrapping

Provides measure of uncertainty
Allows computing confidence intervals

Two philosophies for model comparison

Goodness-of-fit
Cross-validation