Model fitting

By Jan Drugowitsch

neuromatch academy

About myself

Assistant Professor of Neurobiology
Harvard Medical School

Computational Neuroscience lab

- Bayesian computations in the brain
- Decision-making

Overview

Day 1 There are different kinds of useful models, and they all have parameters

Day 2 How to come up with models

- We have manually selected parameters that seemed to work
- We have compared the R^2 of 2 alternative models to see which one is better

Day 3 (today) How to fit these models and evaluate them

- How to correctly choose the best parameters → model fitting
- How to property evaluate how good a model is wrt. data and/or other models

Two central questions in science

1) Models have parameters
How should we set those?
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 (see Day 1)

Fitting (linear) models

Fitting models

- Purpose
- Linear models

How to fit models

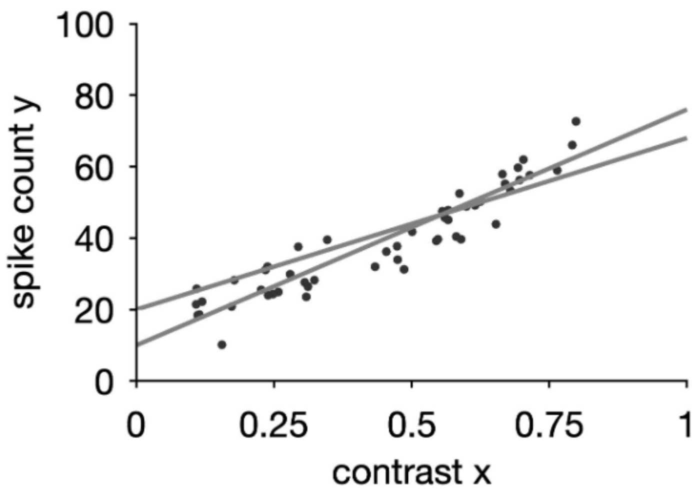
- Fitting models by minimizing errors, or by maximizing likelihood
- Duality between minimizing squared error and maximizing Gaussian likelihood

Assessing model fits

- Bootstrapping to assess parameter uncertainty
- Comparing models

Why we fit models & linear model

A simple linear model



Simple model

spike count ~ increases linearly with contrast

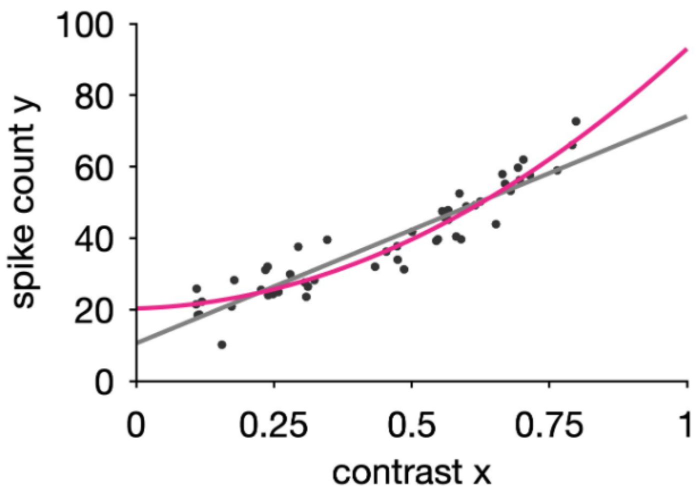
$$y \approx \underbrace{\theta_0}_{\text{intercept}} + \underbrace{\theta_1}_{\text{slope}} x$$

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



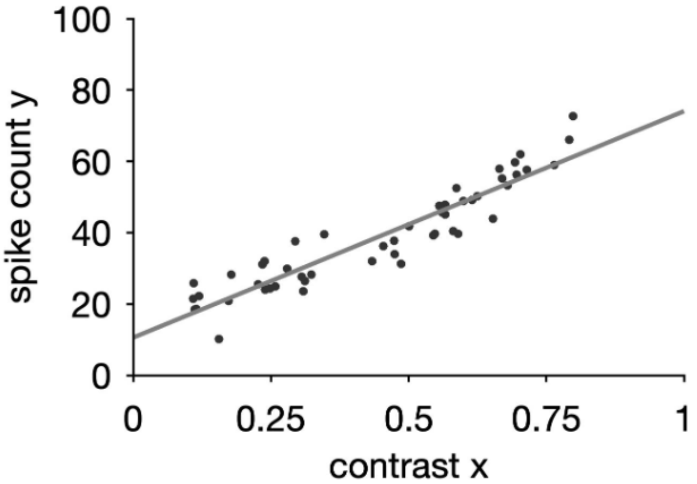
Validation: generate new data
check on held-out data

Prediction: behavior outside of data

Interpret: e.g., spike count ~ contrast? ($\theta_0 \neq 0$?)
(simple models only)

Compare: fits across different models

Linear model can be more complex



spike count ~ increases linearly with contrast

$$y \approx \underbrace{\theta_0}_{\text{intercept}} + \underbrace{\theta_1}_{\text{slope}} x$$

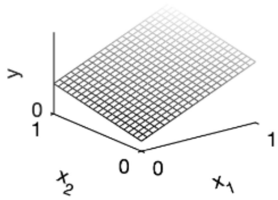
Linear models in general

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

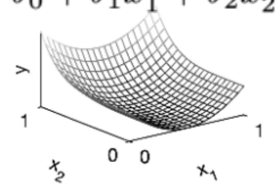
$$\boldsymbol{x} = (x_1, x_2, \dots)^T$$

(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

Models as functions (e.g., Day 2) $y = f(x;\theta)$

Aim: find model with small errors

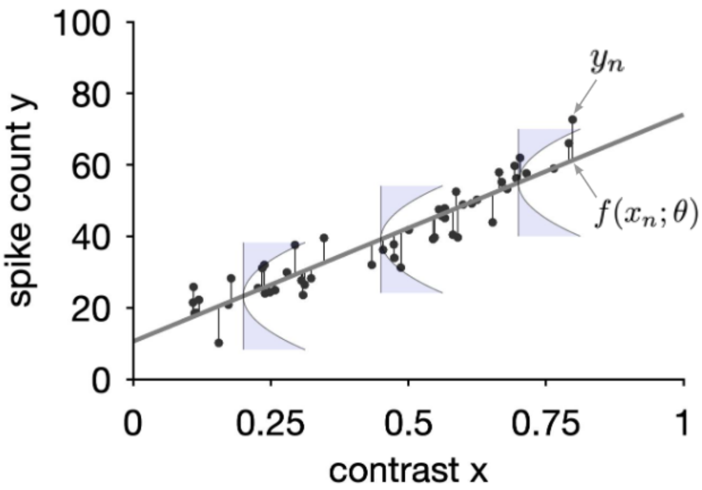
noise from some distribution

Models as generators $y_{\text{measured}} = f(x;\theta) + \eta$

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)

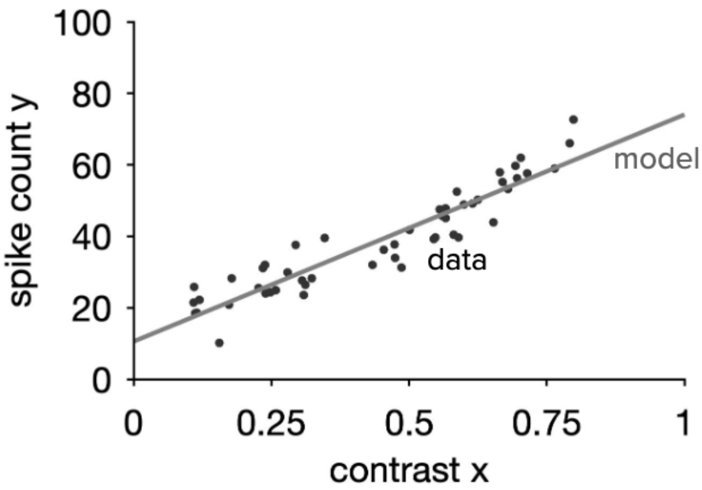
Average squared difference between data and model prediction

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

Best-fitting parameters

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

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

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

"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

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

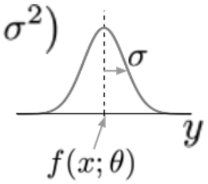
ML for independent trials
If trials are independent, then $\mathcal{L}\left(\theta|\mathrm{data}\right) = \prod_n \mathcal{L}\left(\theta|\mathrm{data}_n\right)$
As a result,

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

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

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

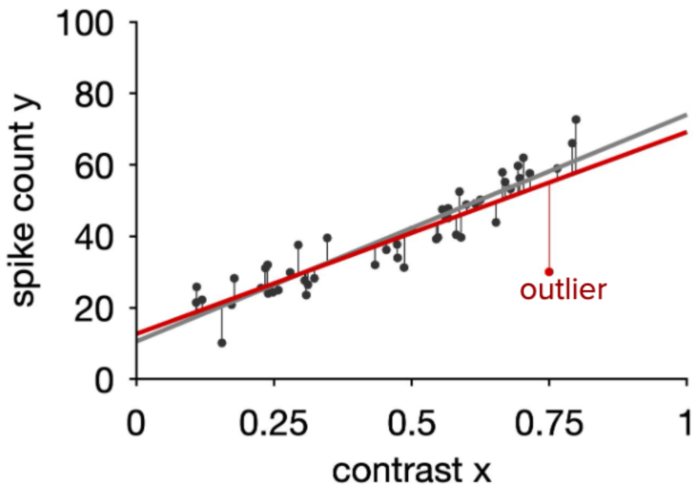
$$= -\frac{N}{2\sigma^2}\frac{1}{N}\sum_n (y_n - f(x_n;\theta))^2 + \text{const.} \quad = -\frac{N}{2\sigma^2}\boxed{\text{MSE}(\theta)} + \text{const.}$$

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 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

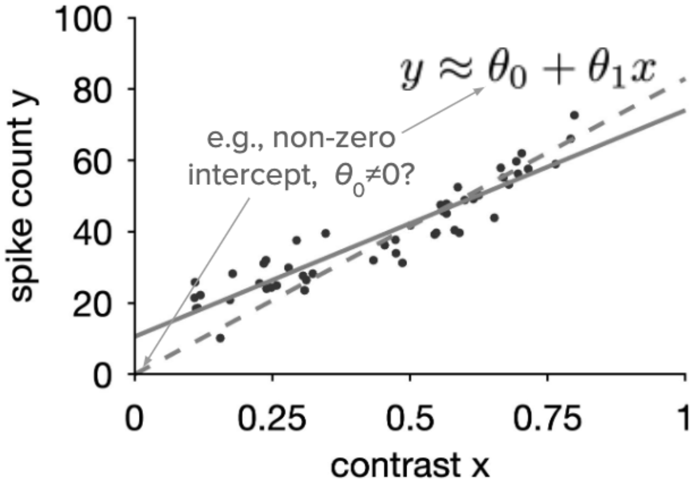
Limited data → multiple parameter values ***θ*** might explain the data about equally well. Reflects *inherent uncertainty* about best-fitting parameters.

Example uses

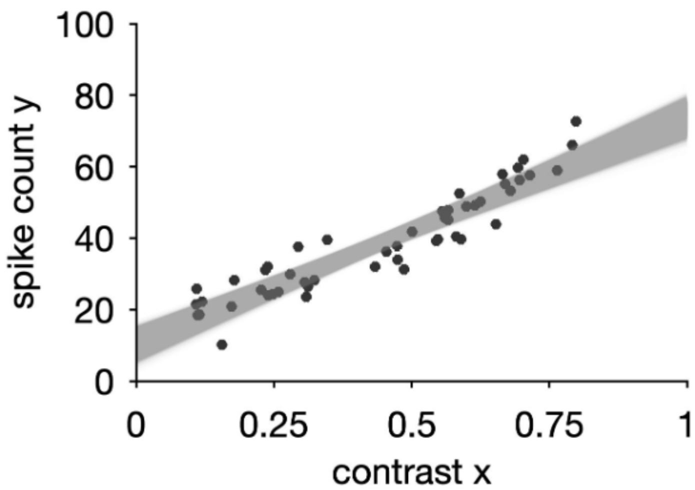
- How well does data constrain parameters?
- Are parameters significantly non-zero (i.e., relevant)?

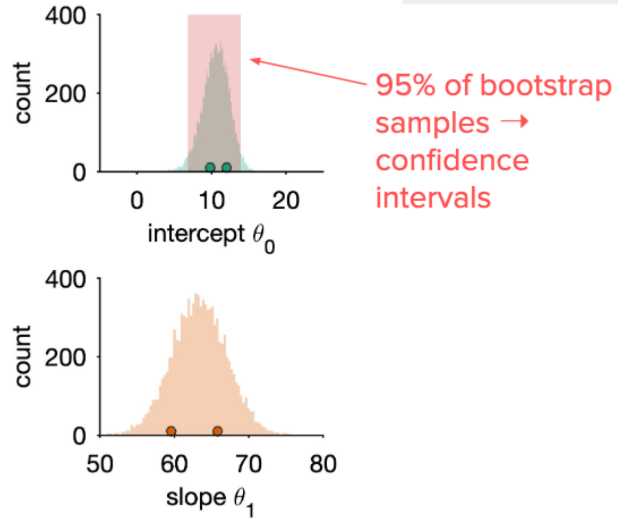
Linear models can assess uncertainty through standard statistics (not discussed further).

Generally assess parameter uncertainty through *bootstrapping*.

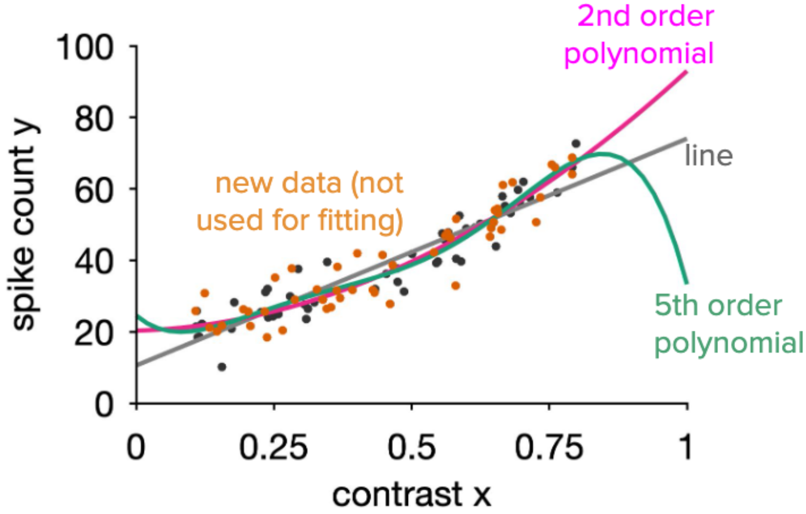


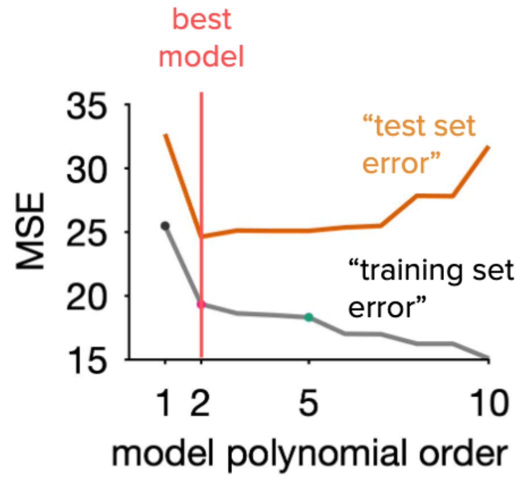
Assessing uncertainty by bootstrap



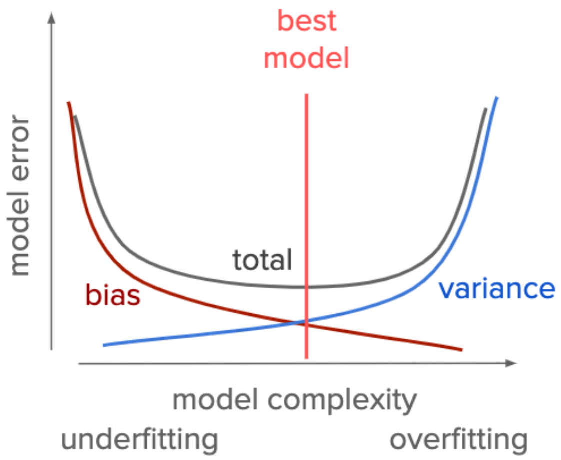


Fitting & comparing multiple models





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 (e.g, Day 2)

Cross validation
(popular in machine learning)

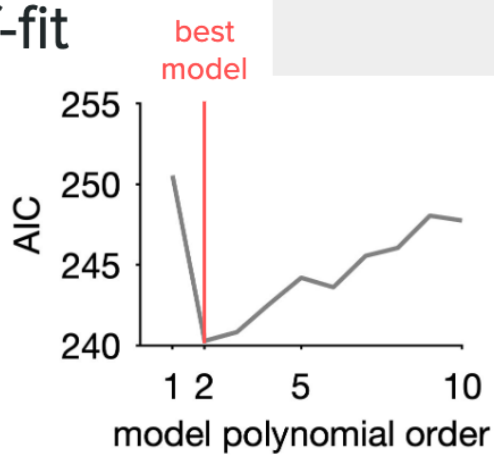
Fit model to some data (training set), then check how well it predicts new data (test set).

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

...

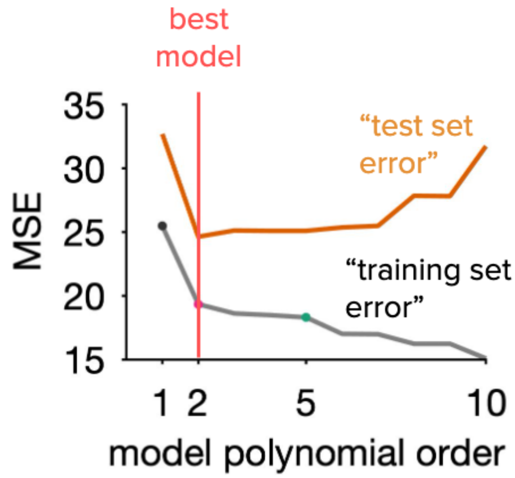
Model comparison by cross-validation

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

More details: today's tutorial



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

Enjoy!

neuromatch academy