# Linear Regression

SWC Neuroinformatics 2024

Lior Fox
Gatsby Unit, UCL

# Intro

Linear regression is ***everywhere***

The practicalities are endless

It is worthwhile taking time to go through the basics

# Outline

- The Least Squares solution in 1 and multiple dimensions
- Model complexity, Bias-Variance tradeoff, and regularization
- Probabilistic / Bayesian interpretation of linear regression
- Some examples

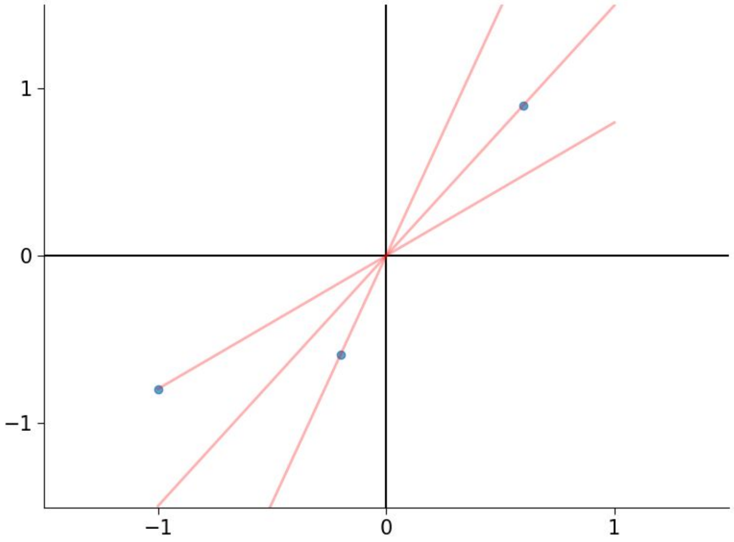
# The problem

- Given a sample ($x^i$, $y^i$), find a good way to describe the dependence of *y* on *x*.

- Immediate questions:
    - What makes a fit “good”?
    - What type of dependences are we willing to consider?

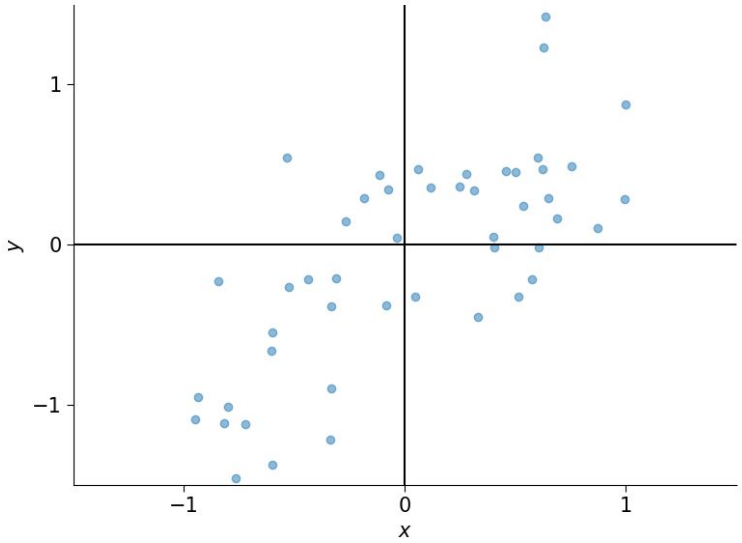
- Keep in mind: we would like our model to be able to predict *unseen* future data

- Restrict to **linear** model: **y ~ bx**
    - If there was a single point, we could fit perfectly
    - What shall we do if there are multiple points?



# Least Squares

- Instead of trying to perfectly fit each point, minimise the sum of squared errors
- The minimisation is over the parameter, ***b***
- Why squared error?
  - Mathematically convenient
  - The “right” thing under some assumptions (later)
- Deriving the solution



# Multiple predictors

- Each **x** is now a vector $[x_1\ x_2\ \ldots\ x_p]^T$
- Linear mapping x ↦ y is parameterised by a vector of coefficients, $\mathbf{y} \sim \mathbf{x}^T\mathbf{b}$

Deriving the solution

Interpretation:

- The analogy to the 1D case
- The predictions for **y** are $\mathbf{Xb} = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$ : the *orthogonal projection* of **y** onto the space spanned by the input vectors

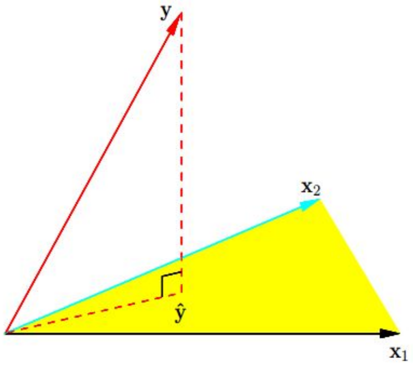


Fig source: The Elements of Statistical Learning

# Adding an offset

- By positing **y ~ bx** we have constrained the regression line to pass through the origin
- We can instead assume **y ~ bx + c** to allow an offset (aka ‘bias’, ‘intercept’)

$$b^* = \mathrm{Cov}[x,y]/\mathrm{Var}[x] \qquad c^* = E[y] - bE[x]$$

- The optimal (minimising MSE) line goes through the sample average
- Therefore we can alternatively *center* the data, and consider the homogeneous model
  - Note that this recovers the “full” solution, as **Cov[x,y]=E[xy]** and **Var[x]=E[x$^2$]**

- Alternatively, we can view **y ~ bx + c** as a multiple predictors case
- For that, we redefine the examples, with mapping x ↦ [x, 1].
- **Exercise**: show the solutions match: $[b^*\ c^*]^T = (X^TX)^{-1}X^Ty$ , after the remapping

# The asymmetry in regression

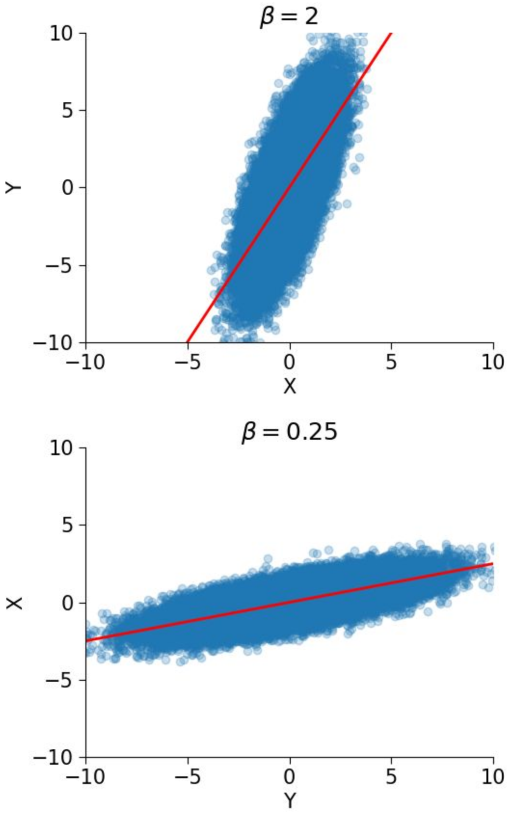
- Regressing **y ~ bx** and **x ~ ay** isn't the same – in general, a ≠ 1/b

- This is due to the division by Var[x]!

$$b_{y\sim x} = E[xy]/E[x^2]$$
$$b_{x\sim y} = E[xy]/E[y^2]$$

- Changes the role of the "noise"

*In this example: $x \sim N(0,1)$, $y=2x+e$, $e \sim N(0,2^2)$*



# Inference and hypothesis testing

- So far we didn’t commit too much to the true data distribution
- If we add some assumptions, we can say more about the result

In particular, we assume that:

- observations **y** were generated by an **unknown** linear model
- additive gaussian noise
- independent (and equal variance) noise between different observations

Mathematically, assume that $\mathbf{y}=\mathbf{X}\boldsymbol{\beta}+\boldsymbol{\varepsilon}$ , with $\boldsymbol{\varepsilon} \sim \mathbf{N}(0,\sigma^2\mathbf{I})$

# Inference and hypothesis testing

We can now work out the distribution of the estimator $\boldsymbol{\beta}$ – this is a simple exercise in manipulating gaussian distributions:

$$
\begin{aligned}
\boldsymbol{\beta} &= (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \\
&= (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T(\mathbf{X}\boldsymbol{\beta}+\boldsymbol{\varepsilon}) \\
&= (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{X}\boldsymbol{\beta} + (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\boldsymbol{\varepsilon} \\
&= \boldsymbol{\beta} + (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\boldsymbol{\varepsilon}
\end{aligned}
$$

And therefore, we have $\boldsymbol{\beta} \sim \mathbf{N}(\boldsymbol{\beta},\ (\mathbf{X}^T\mathbf{X})^{-1}\boldsymbol{\sigma}^2)$

- Unbiased
- Lowest variance among all linear unbiased estimators (*Gauss Markov Theorem)*

# Inference and hypothesis testing

- Now that we have the distribution of **β** we can perform hypothesis testing
- For example, we might want to test whether $\beta_i \neq 0$:
  - Write the distribution of $\beta_i$ under the null, i.e., under $\beta_i=0$, and compare the observed value
  - In our case, this would be a simple Z-test (since we assumed known variance)
  - If the variance is unknown, it can be estimated, resulting in a *t*-distribution for the standardised $\beta_i$

- The assumptions allow the usage of many standard statistical tools (t-test, F-test, etc)

- Can be slightly relaxed if we are willing to use nonparametric tests (e.g., bootstrap)

# Rethinking the “linear” in linear regression

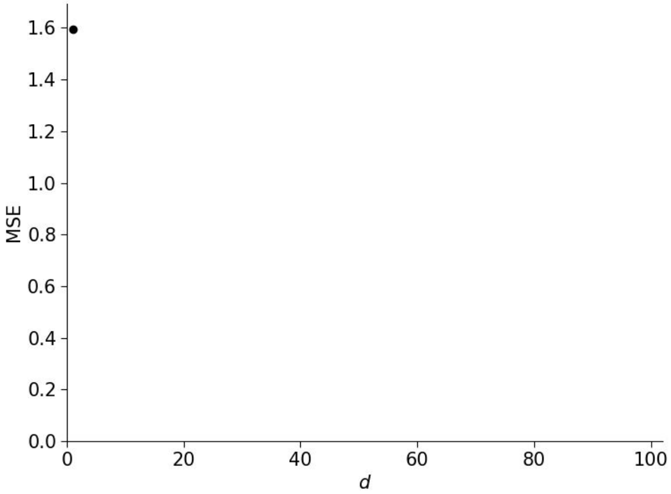
- The important thing was linearity **in the parameters**
- We can add more predictors, which are functions of the original variables, resulting in non-linear functions of those

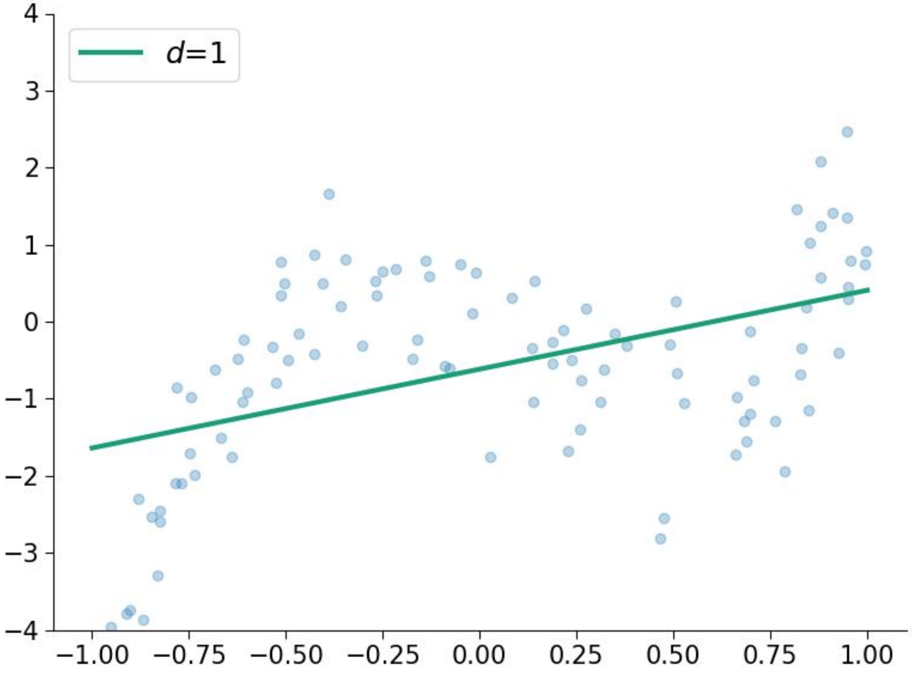
**Example**: polynomial regression

- Suppose we have 1D inputs x. We map $\mathbf{x} \mapsto [\mathbf{x}\ \ \mathbf{x}^2\ \ \mathbf{x}^3 \ldots \mathbf{x}^d]^T$, and solve for **b**
- Then effectively we have a model $\mathbf{y \sim b_1x + b_2x^2 + \ldots + b_dx^d}$ – a *d* degree polynomial

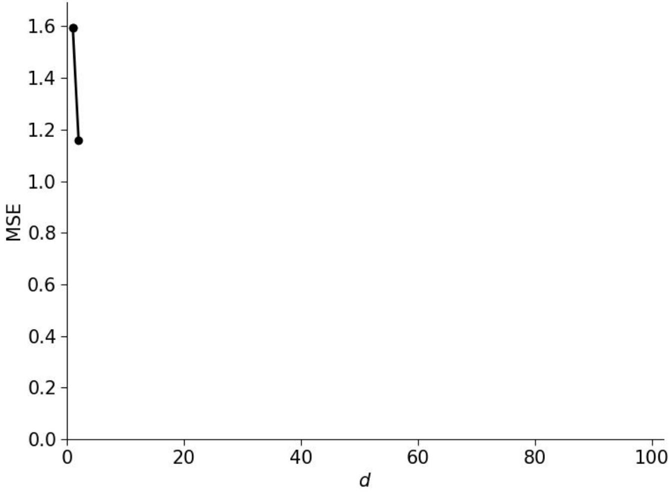
- What happens as we increase **d**?
  - We are making the model less and less constrained
  - Since poly(d) $\subset$ poly(d+1), we can only decrease the error by taking larger and larger **d**
  - Is this the right thing to do?

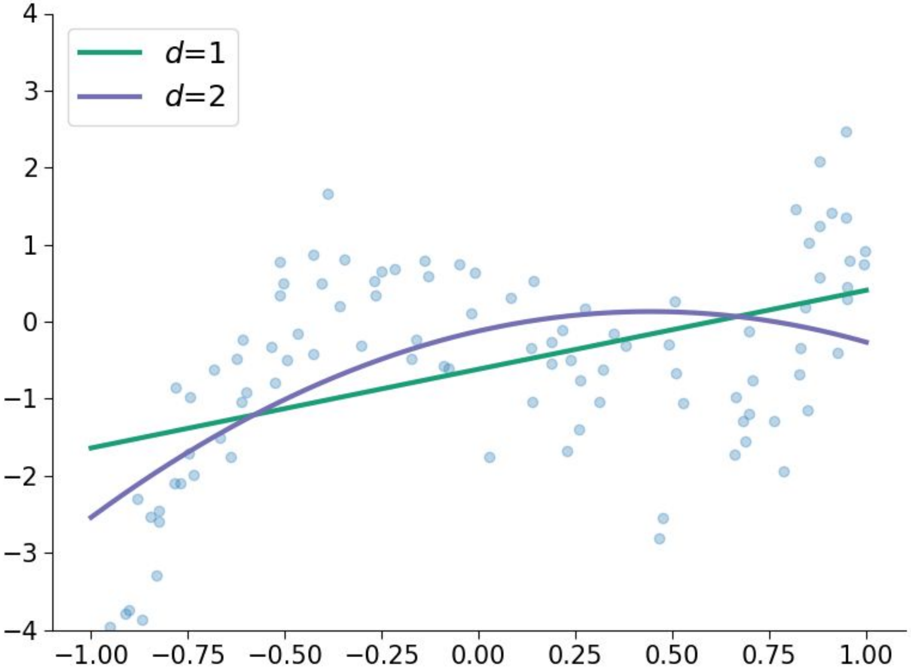
# Model complexity



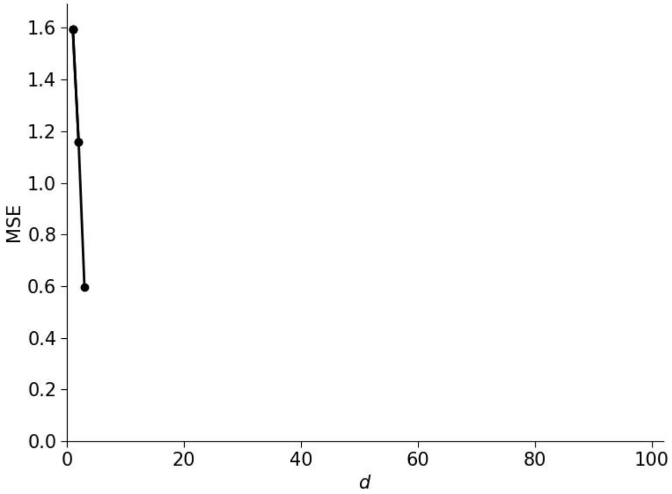


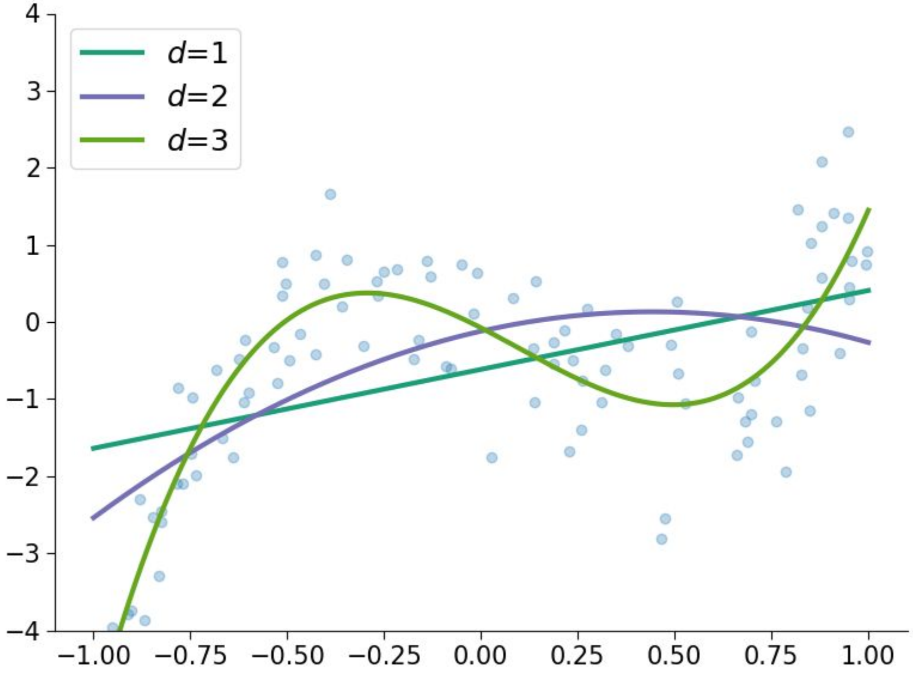
# Model complexity



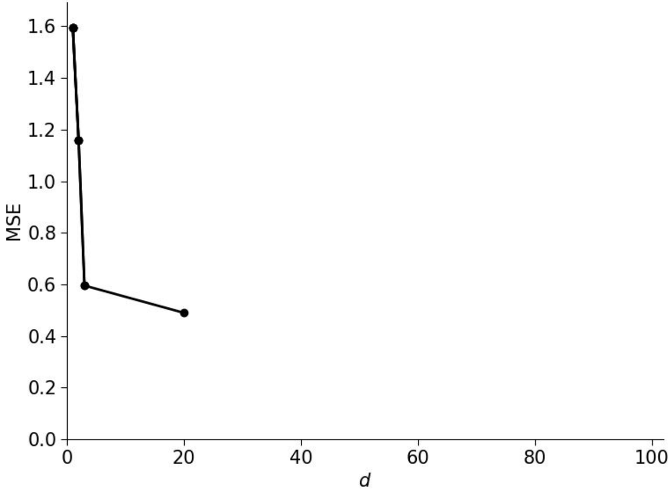


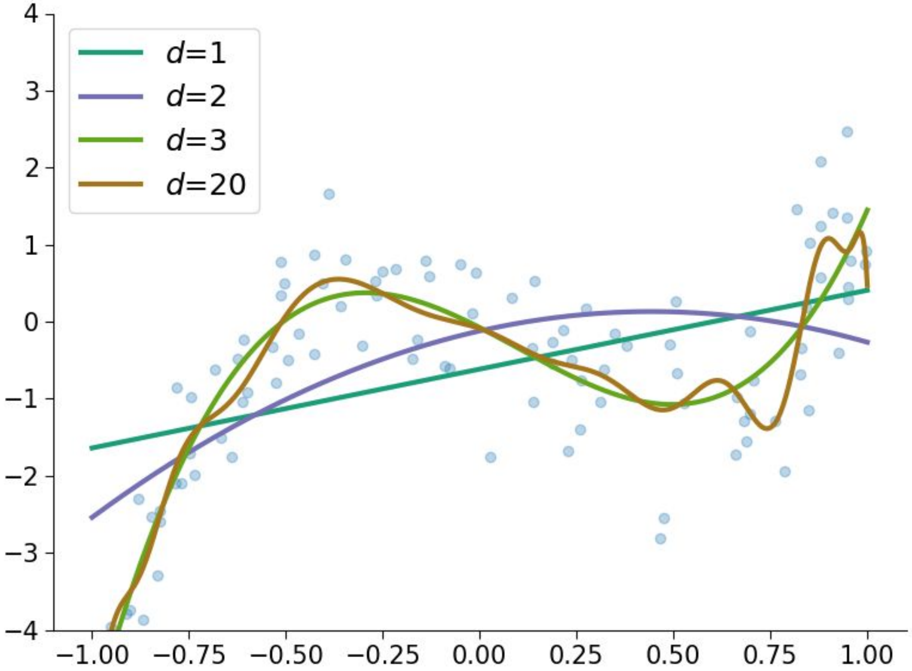
# Model complexity



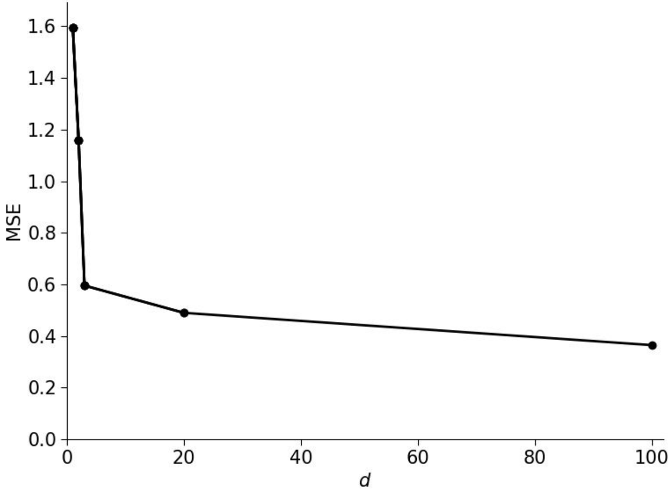


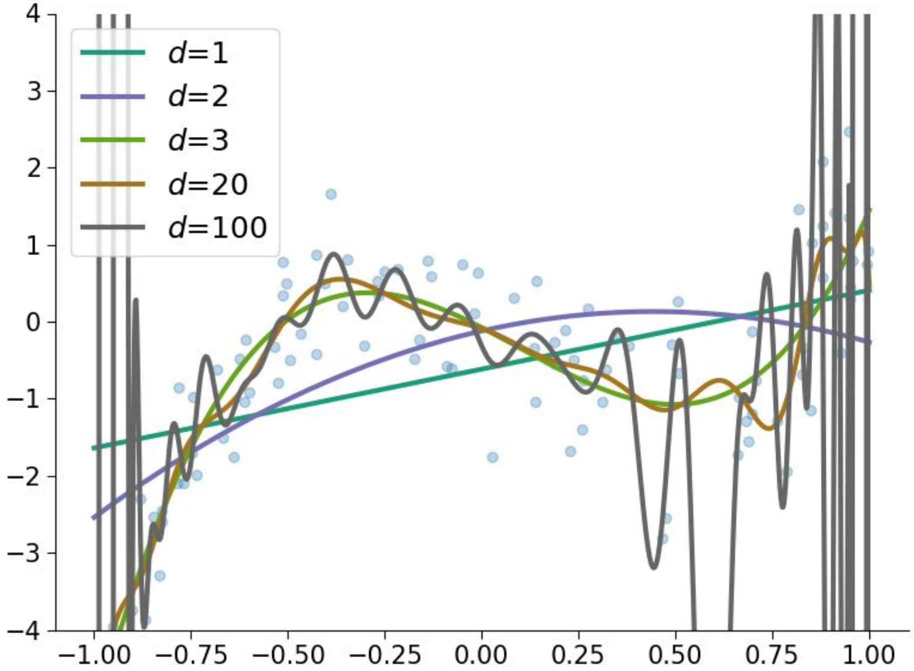
# Model complexity



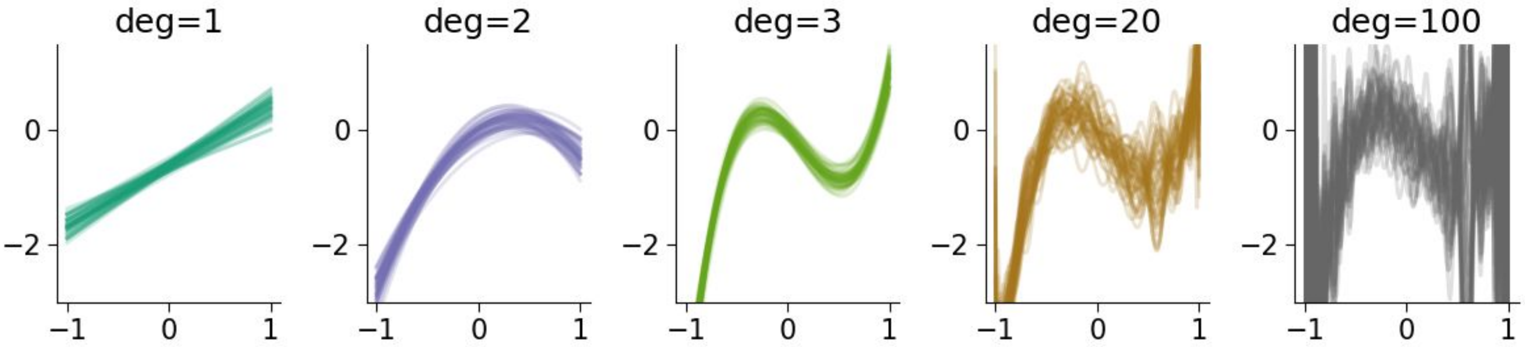


# Model complexity





# Model complexity



# The bias-variance tradeoff

The complex models have:

- Low **bias** – they will fit whatever sample we give them
- High **variance** – different samples will result in very different predictions

The simple models have:

- High **bias** – they will systematically differ from the observations
- Low **Variance** – different samples result in similar predictions

The expected generalization error depends on both bias and variance

# Bias-Variance decomposition of MSE

- Assume the true data is generated by **$y = f(x) + \varepsilon$**
- We estimate $\hat{f}$ based on a sample $(x^1,y^1, \ldots , x^N,y^N)$
- what is the expected error on a **new** datapoint **$x$**?

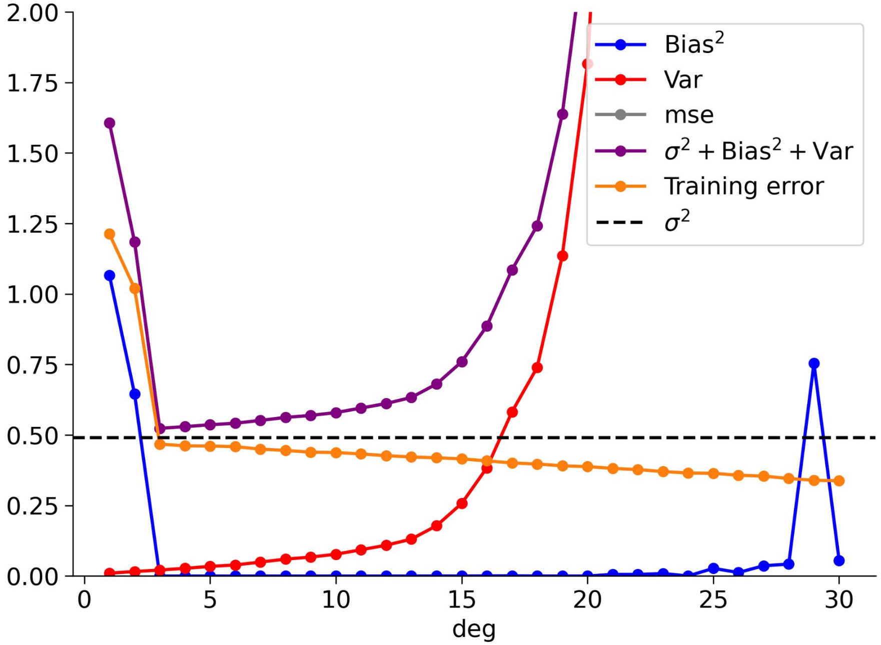
$$
\begin{aligned}
E[(y-\hat{f})^2] &= E[(f + \varepsilon - \hat{f})^2] \\
&= E[\varepsilon^2 + f^2 - 2\hat{f}f + \hat{f}^2] && (\ \varepsilon \text{ uncorrelated with } \hat{f}\ ) \\
&= \sigma^2 + f^2 - 2fE[\hat{f}] + Var[\hat{f}] + E^2[\hat{f}] && (\ E[X^2]=Var[X]+E^2[X]) \\
&= \sigma^2 + (E[\hat{f}]-f)^2 + Var[\hat{f}]
\end{aligned}
$$

The total expected error: Irreducible error + Bias squared + Variance

# The polynomial regression example, again

- Training error keep decreasing
- Generalization error diverges

Making the model more and more complex was **not** the right thing to do



# Regularization

- It is sometimes beneficial to explicitly constraint model complexity, even if we make it biased as a result. This is known as regularization

- The typical way: constraint the magnitude of **β**, “shrinking” parameters towards 0
- Solve a different optimisation problem. For example:

$$\boldsymbol{\beta} = \min\{(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})^T(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}) + \lambda\boldsymbol{\beta}^T\boldsymbol{\beta}\ \} \qquad \text{(Ridge regression)}$$

$$\boldsymbol{\beta} = (\mathbf{X}^T\mathbf{X}+\lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y} \qquad \text{(Solution)}$$

- Alternatively, could use $L_1$ (rather than $L_2$) norm, resulting in the Lasso objective

# Regularization – Ridge and Lasso

- Both methods shrink the parameters, but there are some differences
- Ridge will shrink magnitude of all parameters without setting them to 0
- Lasso will result in a sparse solution, setting small parameters to 0

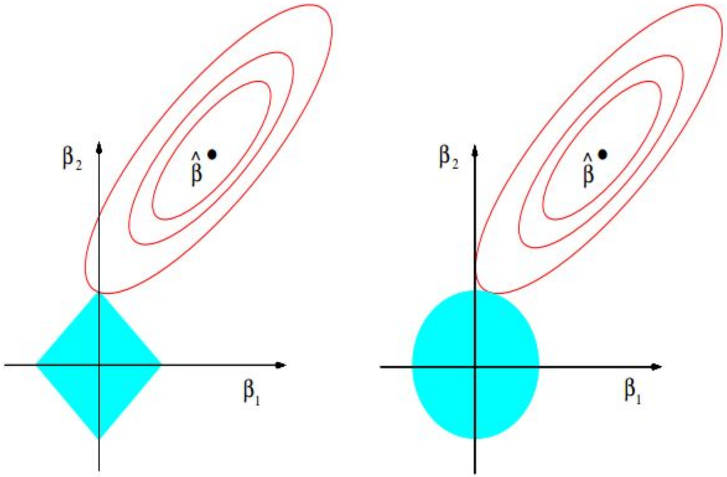


Fig source: The Elements of Statistical Learning

# Ridge regression – geometric interpretation

- Recall – the SVD decomposition $\mathbf{X=U\Sigma V^T}$
- Plugging into the (ordinary) solution:

$$\mathbf{Xb} = \mathbf{X(X^TX)^{-1}X^Ty} = \mathbf{UU^Ty}$$

- The orthogonal projection onto the column space of **X**, as we've seen before
- The ridge regression solution is

$$\mathbf{Xb} = \mathbf{X(X^TX+\lambda I)^{-1}X^Ty} = \mathbf{U\Sigma(\Sigma^2+\lambda I)^{-1}\Sigma U^Ty}$$

- Same projection, but each component is scaled down by $d_i^2/(d_i^2+\lambda)$
- PCs with small variances are shrinked more

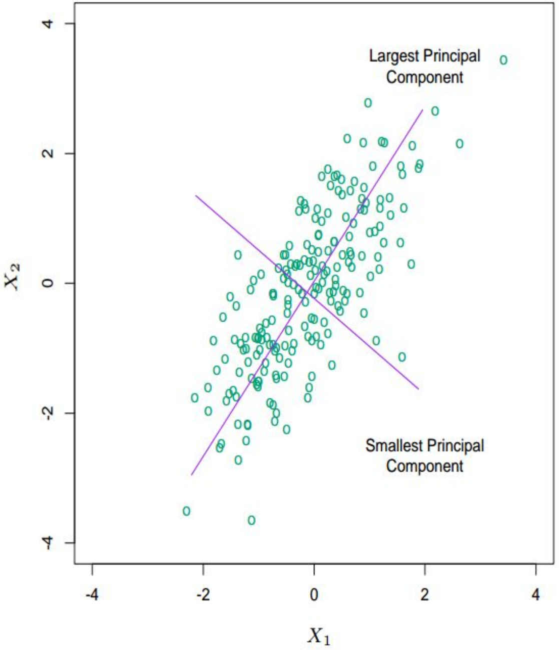


Fig source: The Elements of Statistical Learning

# Probabilistic view of linear regression

- Back to the basics: why minimise the **square error**?
- If we assume that the data was generated by an unknown linear model with additive gaussian noise, this is equivalent to **maximising the likelihood**:

$$\begin{aligned} \log P(y_1, \dots, y_N|b) &= \log P(y_1|b) + \dots + \log P(y_N|b) \\ &= -0.5 * \sigma^{-2}(y_1 - x_1\, b)^2 - \dots - 0.5 * \sigma^{-2}(y_N - x_N\, b)^2 + C \end{aligned}$$

- Therefore, the **maximum-likelihood estimator** for β is the least square estimator

# Ridge regression – Bayesian interpretation

- We use the data to update our belief about the true parameter.
- But what if we already have some prior belief? We use Bayes' theorem:

$$p(\beta \mid D) \propto p(\beta)\ p(D|\beta)$$

- For example, we might assume a-priori that **$\beta \sim N(0,\tau I)$**. Then, the log posterior is:

$$\mathrm{logp}(\beta \mid D) = C\ \ -0.5\ \tau^{-1}\beta^T\beta - 0.5\sigma^{-2}(y-X\beta)^T(y-X\beta)$$

- Maximizing this is the same as solving ridge regression, with **$\lambda = \sigma^2/\tau$**
- Consider the extreme cases:

**"Uniform"/flat prior**: Large **$\tau$**, resulting in **$\lambda \rightarrow 0$**, reducing to OLS

**Strong prior:** Small **$\tau$**, resulting in **$\lambda \rightarrow \infty$**, retaining $\beta=0$

# Towards Bayesian Linear Regression

- The Gaussian case is special because of conjugacy
    - posterior remains a gaussian with updated parameters

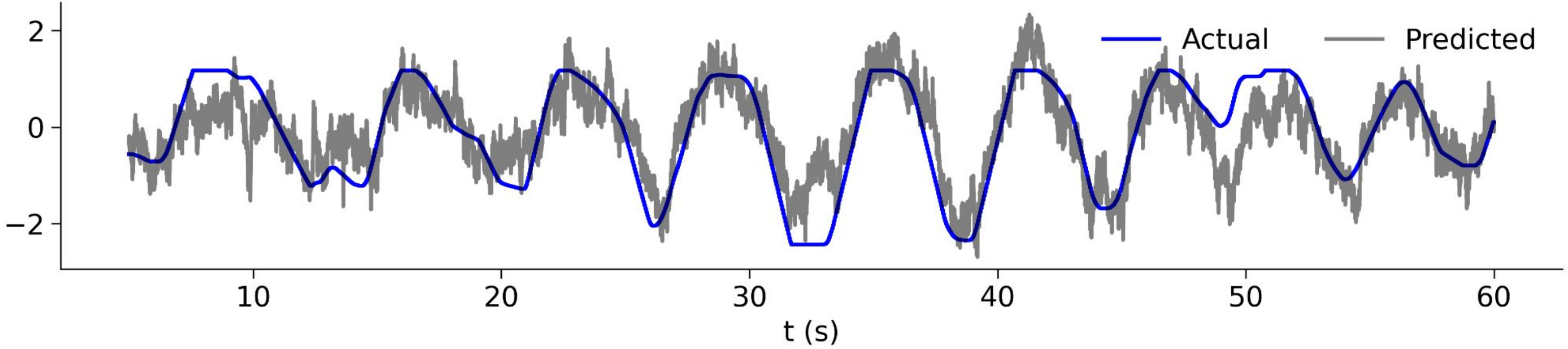
- In this case, the MAP is also the posterior mean

- The general case (non-zero prior mean, general prior covariance) is also tractable

- Serves as the basic intuition/building blocks for many more Bayesian models

# Example 1: Decoding

- We try to predict stimulus/behavior from neural data
- Example: subjects using  joystick to move a cursor, following target location
- Predict the (standardised) cursor y-location from ECoG data

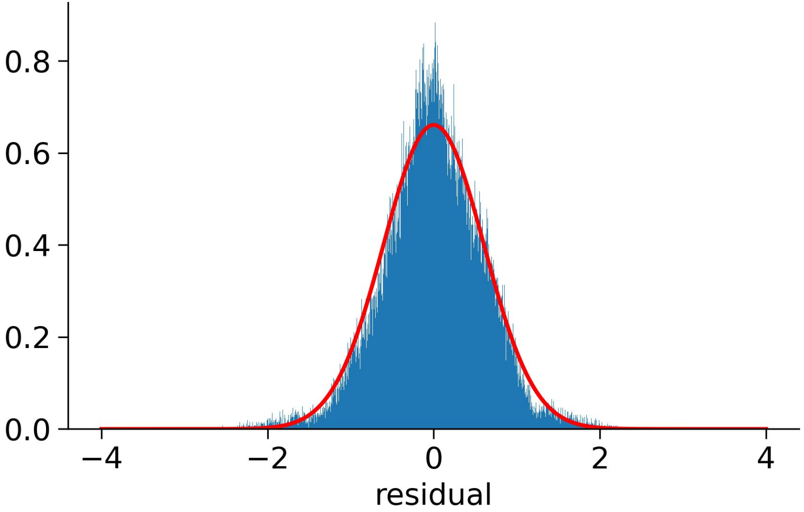


***Ethics statement****: All patients participated in a purely voluntary manner, after providing informed written consent, under experimental protocols approved by the Institutional Review Board of the University of Washington (#12193). All patient data was anonymized according to IRB protocol, in accordance with HIPAA mandate. It was made available through the library described in “A Library of Human Electrocorticographic Data and Analyses” by Kai Miller [Reference], freely available at https://searchworks.stanford.edu/view/zk881ps0522*

Data: https://exhibits.stanford.edu/data/catalog/zk881ps0522 ; paper: Shalck et al. 2007, https://iopscience.iop.org/article/10.1088/1741-2560/4/3/012

# Example 1: Decoding

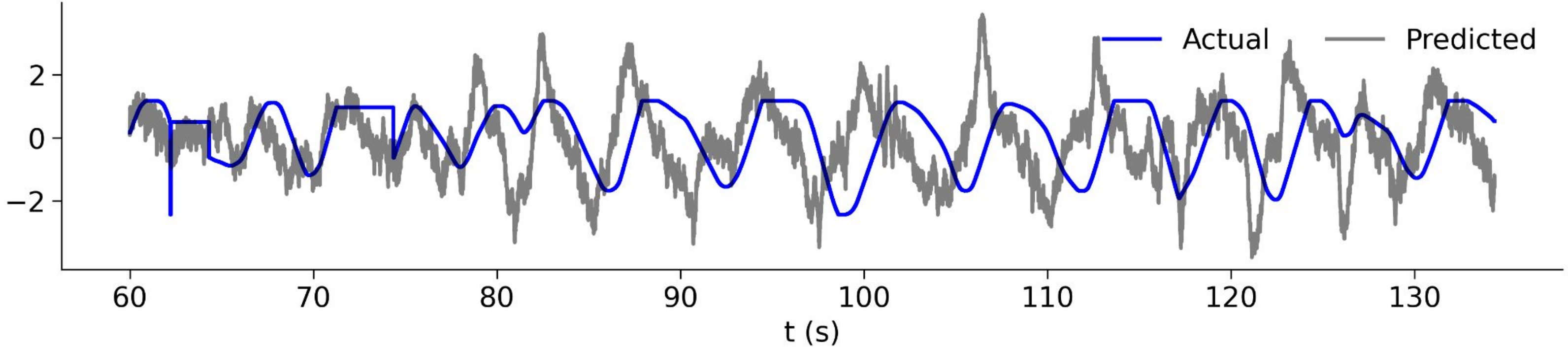
- We try to predict stimulus/behavior from neural data
- Example: subjects using  joystick to move a cursor, following target location
- Predict the cursor location from ECoG data



Data: https://exhibits.stanford.edu/data/catalog/zk881ps0522 ; paper: Shalck et al. 2007, https://iopscience.iop.org/article/10.1088/1741-2560/4/3/012

# Example 1: Decoding

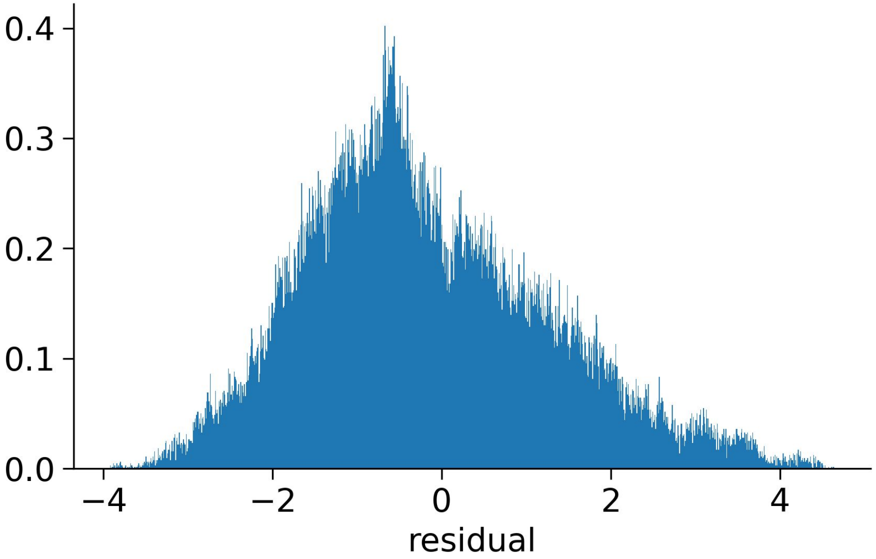
- What about test?



Data: https://exhibits.stanford.edu/data/catalog/zk881ps0522 ; paper: Shalck et al. 2007, https://iopscience.iop.org/article/10.1088/1741-2560/4/3/012

# Example 1: Decoding

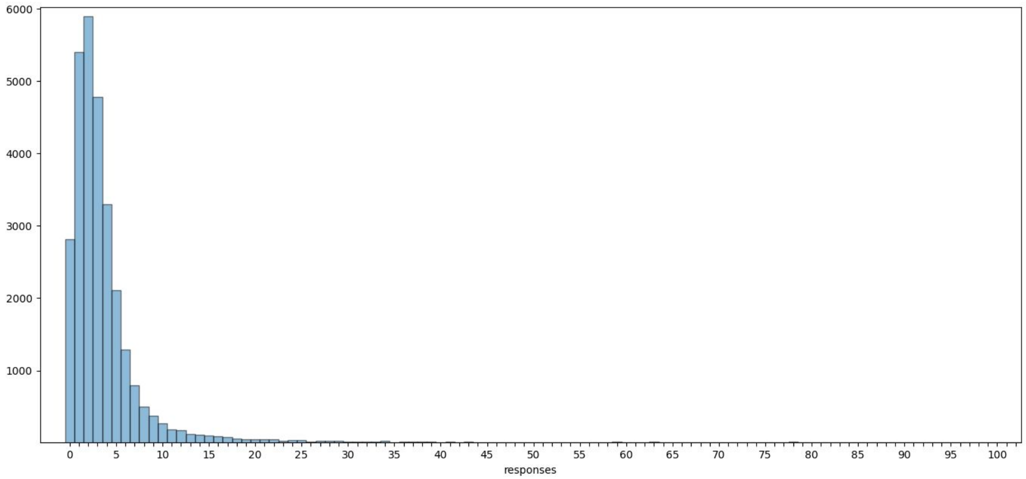
- What about test?



Data: https://exhibits.stanford.edu/data/catalog/zk881ps0522 ; paper: Shalck et al. 2007, https://iopscience.iop.org/article/10.1088/1741-2560/4/3/012

# Example 2: Encoding (Receptive Fields)

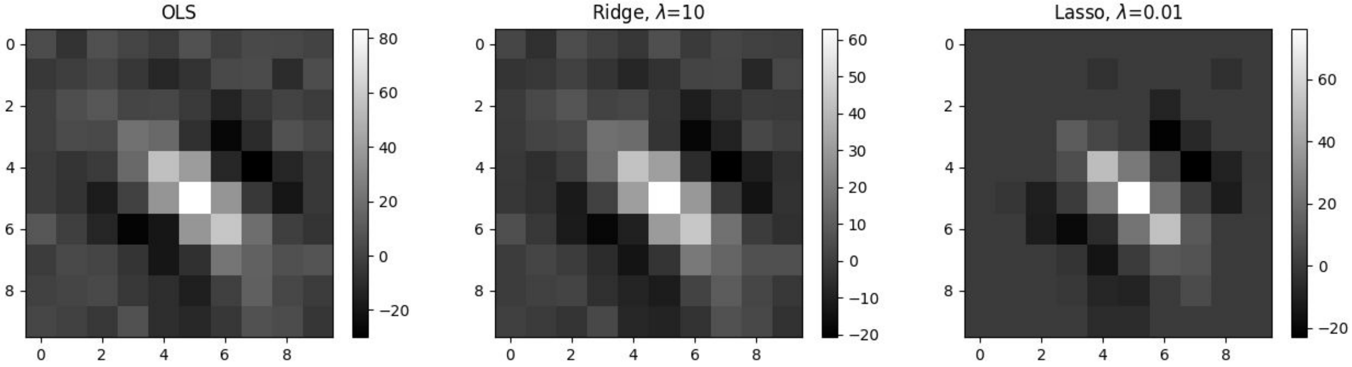
- Instead of neural→stimulus, we try to predict stimulus→neural
- Example: presented images, simple cell responses (spike counts)
- we really **shouldn’t** treat this as ordinary linear regression, but we will anyway



Data: https://www.gatsby.ucl.ac.uk/~rapela/projects/vrst/ ; paper: Rapela et al. 2006 https://jov.arvojournals.org/article.aspx?articleid=2192869

# Example 2: Encoding (Receptive Fields)

- Coefficients has the same dimensions as the input – in our case, it it’s an image
- A way of describing the fitted (linear) receptive field of the cell!



Data: https://www.gatsby.ucl.ac.uk/~rapela/projects/vrst/ ; paper: Rapela et al. 2006 https://jov.arvojournals.org/article.aspx?articleid=2192869

# Generalizations to different types of data

- The assumptions beyond the noise model:
  - Errors are symmetric around 0
  - Small errors are more likely than large errors

- Makes sense for continuous observations corrupted by measurement noise
- What if we have different type of observations?
  - Integers (e.g., number of spikes)
  - Binary outcomes (e.g., behavioral decision)
  - Categorical outcomes

- Different generalization exist (logistic regression, poisson regression, etc)
- Typically, cannot be solved in closed form (but can be optimised efficiently)

# Conclusions and further directions

- The model is almost always wrong

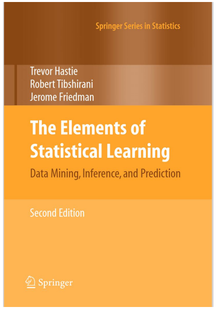
- Linear models are simple enough to analyze analytically. This makes them:
    - Extremely useful in their own right
    - Important for building intuition and understanding of more complicated models

- Ultimately, we built a model for **correlations** among predictors and outcomes
- Extra caution should be taken in interpreting the results as a “causal” story, particularly if fitted to observational data

- Many topics we haven’t covered
    - Cross-validation, GLMs, (fully) Bayesian linear regression, subset selection, …

# Sources and materials

**The Elements of Statistical Learning**, Hastie, Tibshirani, Friedman (available online)



Springer Series in Statistics

Trevor Hastie
Robert Tibshirani
Jerome Friedman

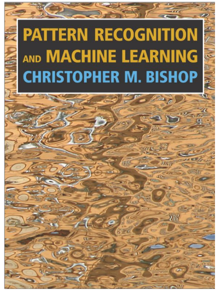
The Elements of Statistical Learning

Data Mining, Inference, and Prediction

Second Edition

Springer

**Pattern recognition and machine learning**, Bishop



PATTERN RECOGNITION AND MACHINE LEARNING

CHRISTOPHER M. BISHOP

Nueormatch academy tutorials [https://compneuro.neuromatch.io/tutorials/intro.html](https://compneuro.neuromatch.io/tutorials/intro.html)

**LOTS** of material online – but not everything is equally good