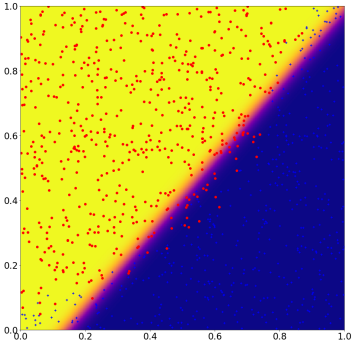
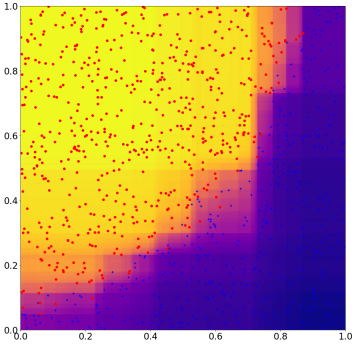
Machine Learning 1.09: Regularisation

Tom S. F. Haines
T.S.F.Haines@bath.ac.uk

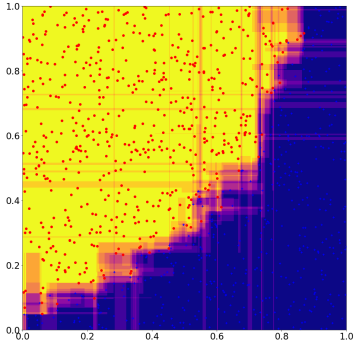
Underfitting & overfitting



- Underfitting
- Logistic regression



- Balanced
- Tuned random forest
- (scikit learn, `min_impurity_decrease=0.008`, `n_estimators=512`)



- Overfitting
- Badly tuned random forest
- (scikit learn, default parameters)

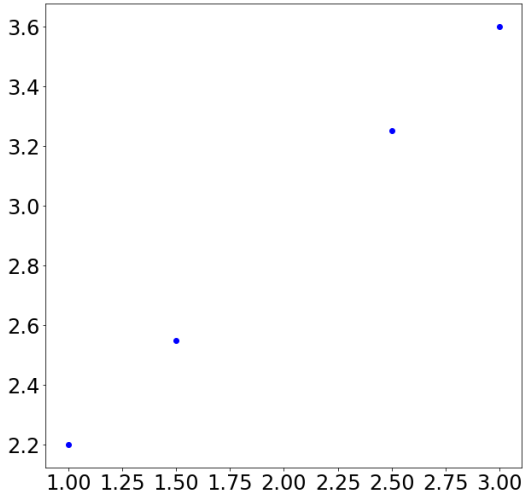
Regularisation

- This lecture is about **regularisation**
- Fixes overfitting
- Effectively "extra information"

UNIVERSITY OF BATH

Extra information

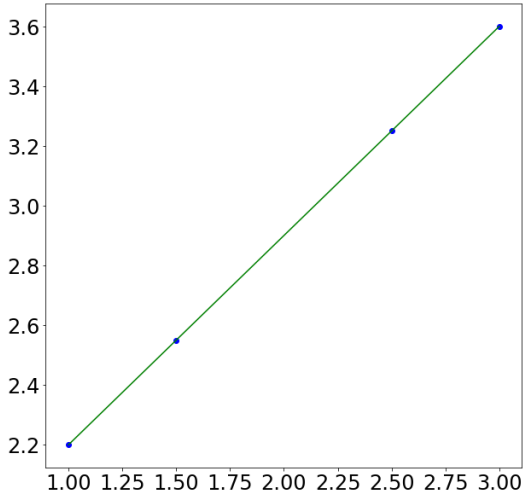
- 1D regression, 4 points



UNIVERSITY OF BATH

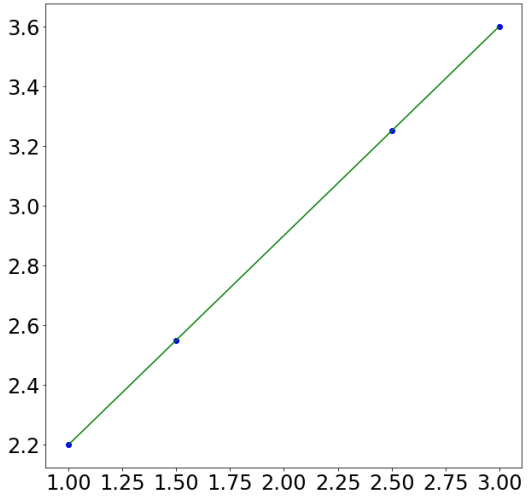
Extra information

- 1D regression, 4 points
- Linear solution obvious – to us!



Extra information

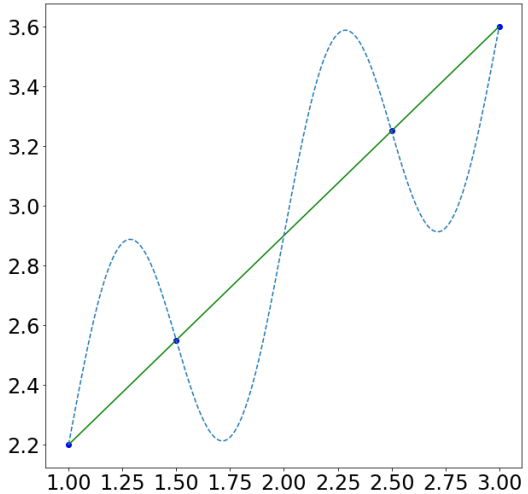
- 1D regression, 4 points
- Linear solution obvious – to us!
- Assume general model – anything goes



Extra information

- 1D regression, 4 points
- Linear solution obvious – to us!

- Assume general model – anything goes
- e.g. a sine curve

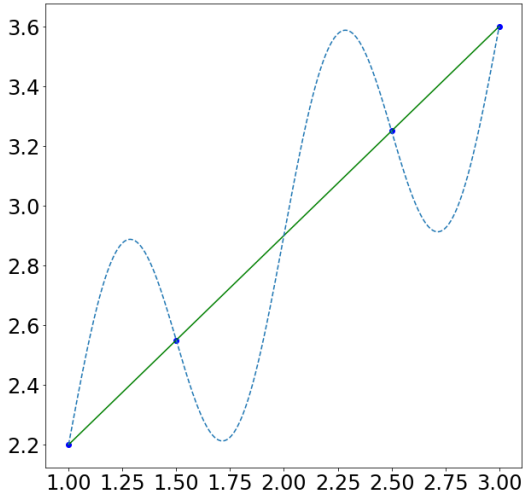


Extra information

- 1D regression, 4 points
- Linear solution obvious – to us!

- Assume general model – anything goes
- e.g. a sine curve

- Perfect match at known points...
- **Identical cost to a straight line**!

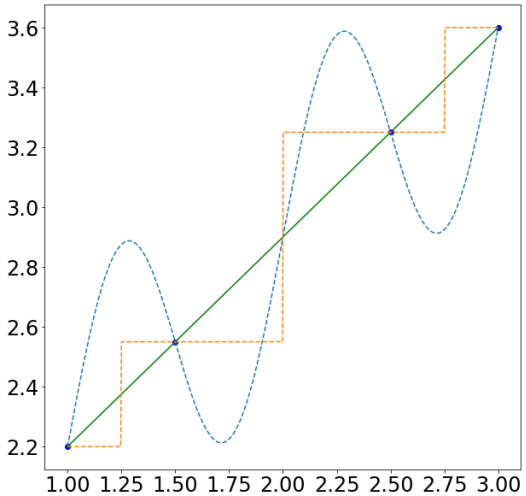


Extra information

- 1D regression, 4 points
- Linear solution obvious – to us!

- Assume general model – anything goes
- e.g. a sine curve

- Perfect match at known points...
- **Identical cost to a straight line**!



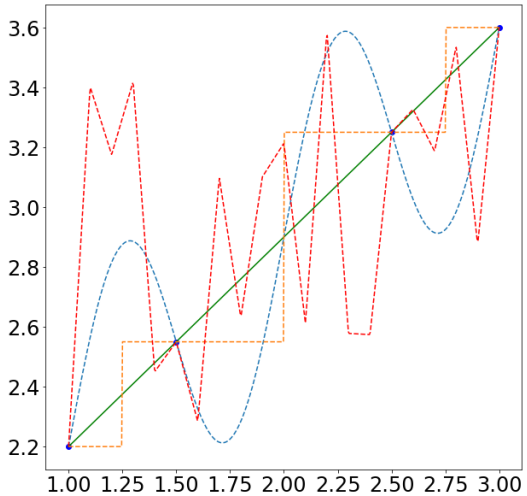
UNIVERSITY OF BATH

Extra information

- 1D regression, 4 points
- Linear solution obvious – to us!

- Assume general model – anything goes
- e.g. a sine curve

- Perfect match at known points...
- **Identical cost to a straight line**!



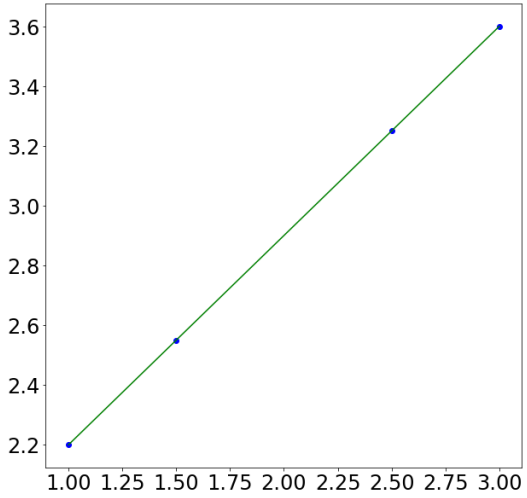
Extra information

- 1D regression, 4 points
- Linear solution obvious – to us!

- Assume general model – anything goes
- e.g. a sine curve

- Perfect match at known points...
- **Identical cost to a straight line**!

- Regularisation emphasises **simpler model**
  (straight line)
- **Common sense** for models!



Occam's razor

The simplest explanation is usually the correct one

Occam's razor

The simplest explanation is usually the correct one

- Idea can be traced back to Aristotle (384–322 BC)
- Ockham's version: "Plurality must never be posited without necessity"
  (translated from Latin – William of Ockham was a 13th century priest)
- Overfitting: An unjustifiably complex explanation

UNIVERSITY OF BATH

Reasons for regularisation

- Ill posed problem
- Overfitting

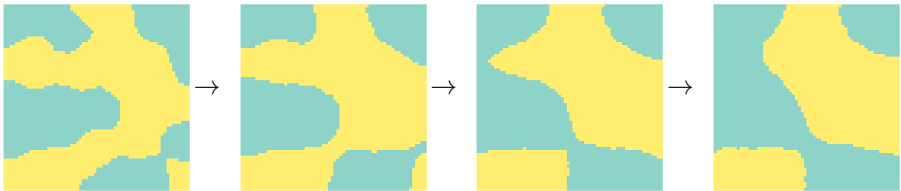
- Extra information
- Human understanding
- Easier optimisation

Often several at once

Reasons: Ill posed

- Ill posed: Multiple equally good solutions
- Just showed an example!

- Regularisation: Force selection, even if arbitrary
- Without: Optimisation can "drift" between solutions:

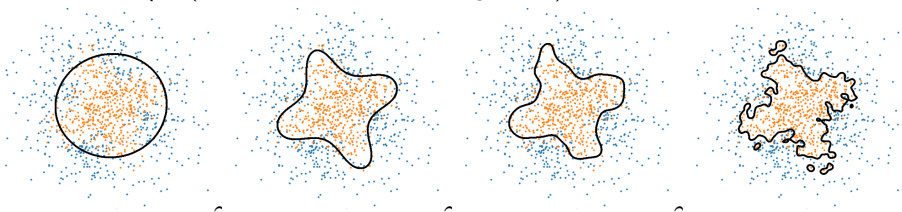


(colours represent different solutions, chasing each other around an image)

Reasons: Overfitting

- Already seen, but a reminder...
- Overfitting = Fitting to noise

- Another example (SVM, rbf kernel – will be taught later):



$\gamma = 0.01, C = 1$ $\gamma = 0.1, C = 1$ (about right) $\gamma = 1.0, C = 1$ $\gamma = 10.0, C = 1$

Reasons: Extra information

- Regularisation may reflect extra information
- Measured noise from a sensor

  (a seperate experiment)

  $\downarrow$

  Correct amount of regularisation to apply to signal

UNIVERSITY OF BATH

Reasons: Human understanding

- Goal: Learn $y = f_\theta(x)$
- Alternatively: Learn $y = f_\theta(z)$ and $z = f_\eta(x)$
  Subject to z being useful in some way, i.e. human interpretable

UNIVERSITY OF BATH

Reasons: Human understanding

- Goal: Learn $y = f_\theta(x)$
- Alternatively: Learn $y = f_\theta(z)$ and $z = f_\eta(x)$
  Subject to z being useful in some way, i.e. human interpretable

- Attribute learning:
  - $z = f_\eta(x)$ encodes: Has tail, black & white, four legs etc.
  - $x = f_\theta(z)$ encodes: Is zebra, is horse, is penguin.

UNIVERSITY OF BATH

Reasons: Human understanding

- Goal: Learn $y = f_\theta(x)$
- Alternatively: Learn $y = f_\theta(z)$ and $z = f_\eta(x)$
  Subject to z being useful in some way, i.e. human interpretable

- Attribute learning:
  - $z = f_\eta(x)$ encodes: Has tail, black & white, four legs etc.
  - $x = f_\theta(z)$ encodes: Is zebra, is horse, is penguin.

- Bit of a stretch calling this regularisation
- But similar goal: Prefer simpler model (as judged by a human)

UNIVERSITY OF BATH

Reasons: Human understanding

- Goal: Learn $y = f_\theta(x)$
- Alternatively: Learn $y = f_\theta(z)$ and $z = f_\eta(x)$
  Subject to z being useful in some way, i.e. human interpretable

- Attribute learning:
  - $z = f_\eta(x)$ encodes: Has tail, black & white, four legs etc.
  - $x = f_\theta(z)$ encodes: Is zebra, is horse, is penguin.

- Bit of a stretch calling this regularisation
- But similar goal: Prefer simpler model (as judged by a human)

- Notes:
  - "Sharing statistical strength": Learning to recognise black & white objects allows images of penguins to improve performance recognising zebras
  - Window into the black box
  - Attribute learning can also be uninterpretable

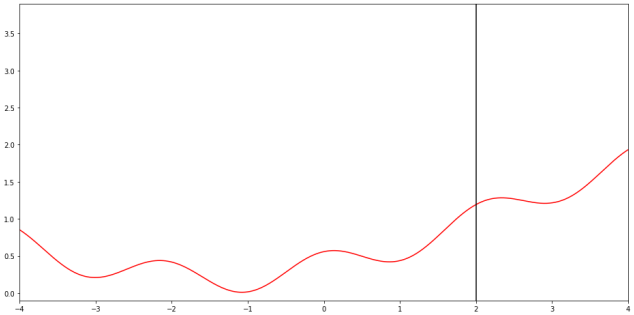
Reasons: Easier optimisation

- Already gave example of "drifting" between solutions
- Regularisation: Can smooth cost function $\rightarrow$ Fewer local minima
  (Also: Removes stationary points, accelerating convergence)

Reasons: Easier optimisation

- Already gave example of "drifting" between solutions
- Regularisation: Can smooth cost function $\rightarrow$ Fewer local minima
  (Also: Removes stationary points, accelerating convergence)

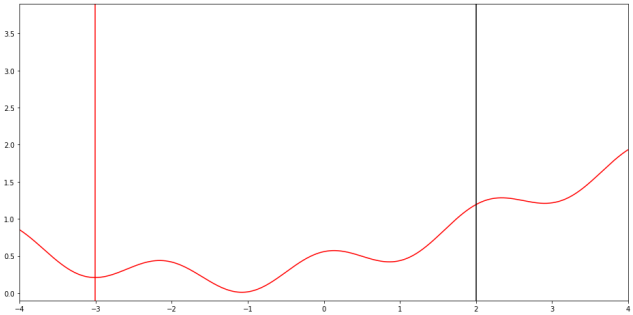
- Find minima of red function
  (starting at black vertical)



Reasons: Easier optimisation

- Already gave example of "drifting" between solutions
- Regularisation: Can smooth cost function $\rightarrow$ Fewer local minima
  (Also: Removes stationary points, accelerating convergence)

- Find minima of red function
  (starting at black vertical)
- Stops at $x = 3$,
  global optima is $x = 1$
  (using BFGS)

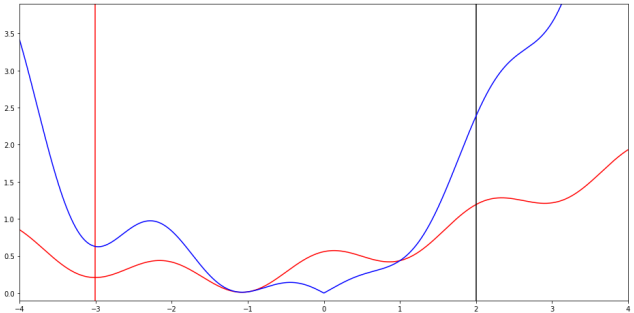


Reasons: Easier optimisation

- Already gave example of "drifting" between solutions
- Regularisation: Can smooth cost function → Fewer local minima
  (Also: Removes stationary points, accelerating convergence)

- Find minima of red function
  (starting at black vertical)
- Stops at $x = 3$,
  global optima is $x = 1$
  (using BFGS)

- Blue: L1 regularisation
  (pushing answer towards $x = 0$)

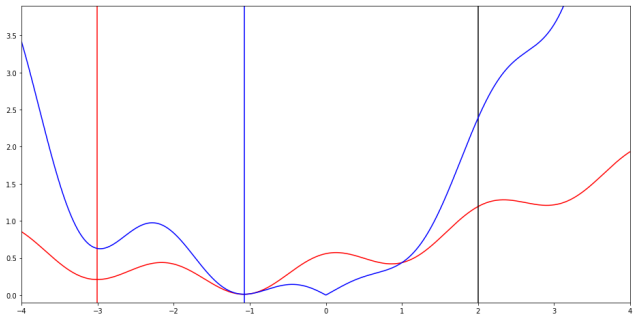


Reasons: Easier optimisation

- Already gave example of "drifting" between solutions
- Regularisation: Can smooth cost function → Fewer local minima
  (Also: Removes stationary points, accelerating convergence)

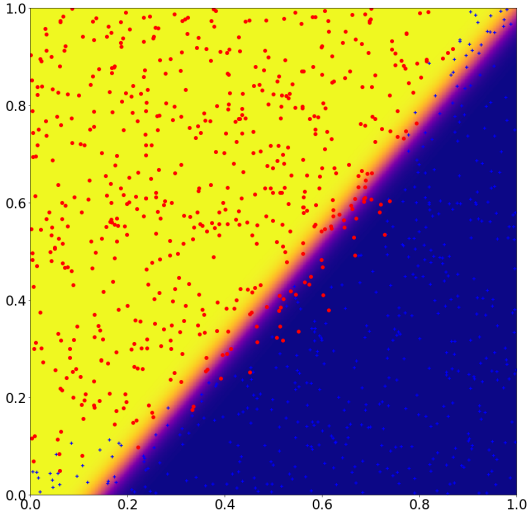
- Find minima of red function
  (starting at black vertical)
- Stops at $x = 3$,
  global optima is $x = 1$
  (using BFGS)

- Blue: L1 regularisation
  (pushing answer towards $x = 0$)
- Finds global optima



Aside: Model limits

- Model limits → Kinda like regularisation,
  e.g. Logistic regression only does straight lines
- But rarely the "right amount",
  and no hyperparameter to tune



Aside: Early stopping

- Model starts simple
- Gets more convoluted as optimisation runs
- So stop early!

UNIVERSITY OF BATH

Aside: Early stopping

- Model starts simple
- Gets more convoluted as optimisation runs
- So stop early!

- What does that even mean?
- Your regularisation is too weak - make it stronger!
- Stupid – avoid

Aside: Quantity I

- More data $\implies$ less regularisation required
- Infinite data $\implies$ no regularisation!
  (simple lookup – nearest neighbours)

Aside: Quantity I

- More data $\implies$ less regularisation required
- Infinite data $\implies$ no regularisation!
  (simple lookup – nearest neighbours)

- Hyperparameters control regularisation strength
- More/less data $\rightarrow$ Different hyperparameters

UNIVERSITY OF BATH

Aside: Quantity I

- More data $\implies$ less regularisation required
- Infinite data $\implies$ no regularisation!
  (simple lookup – nearest neighbours)

- Hyperparameters control regularisation strength
- More/less data $\rightarrow$ Different hyperparameters

- Model may have sweet spot:
  - Not enough data $\rightarrow$ Fail
  - Too much data $\rightarrow$ Stop improving (underfit)

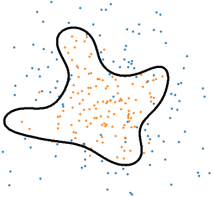
Aside: Quantity II

exemplars $= 16$, $\gamma = 0.5$,
accuracy $= 83.5\%$

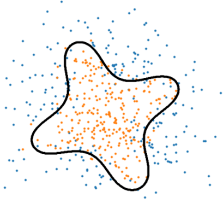
exemplars $= 32$, $\gamma = 0.5$,
accuracy $= 83.4\%$

exemplars $= 64$, $\gamma = 0.5$,
accuracy $= 87.5\%$

exemplars $= 128$, $\gamma = 0.25$,
accuracy $= 86.5\%$

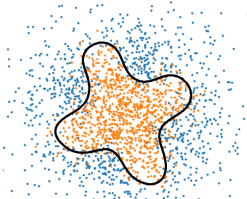


exemplars $= 256$, $\gamma = 0.25$,
accuracy $= 87.2\%$



exemplars $= 512$, $\gamma = 0.1$,
accuracy $= 89.1\%$

exemplars $= 1024$, $\gamma = 0.1$,
accuracy $= 89.3\%$



exemplars $= 2048$, $\gamma = 0.25$,
accuracy $= 89.3\%$

Model kinds I

- Discussed **why**
- **How** depends on model kind...

UNIVERSITY OF BATH

Model kinds I

- Discussed **why**
- **How** depends on model kind...
- Non-probabilistic
  - Arbitrary loss functions
- Probabilistic
  - Maximum likelihood (ML) (no regularisation)
  - Maximum a posteriori (MAP)
  - Bayesian

Non-probabilistic

- Model fitting: Minimises loss function $L(\theta)$

$$L(\theta) = \sum_{i=1}^{n} (y_i - f_\theta(x_i))^2$$

Non-probabilistic

- Model fitting: Minimises loss function $L(\theta)$

$$L(\theta) = \sum_{i=1}^{n} (y_i - f_\theta(x_i))^2$$

- Regularise θ: Add term to encourage small parameters, e.g.

$$L(\theta) = \frac{1}{n} \sum_{i=1}^{n} (y_i - f_\theta(x_i))^2 + \lambda \sum_{j=1}^{k} \theta_k^2$$

- λ controls regularisation strength – hyperparameter

Ridge regression I

Also called Tikhonov regression

For each exemplar:

$$y_i = ax_i + b, \qquad \theta = [a, b]$$

(identical to linear regression)

Loss function:

$$L(\theta) = \sum_{i=1}^{n} (y_i - (ax_i + b)))^2 + \lambda(a^2 + b^2)$$

UNIVERSITY OF BATH

Ridge regression II

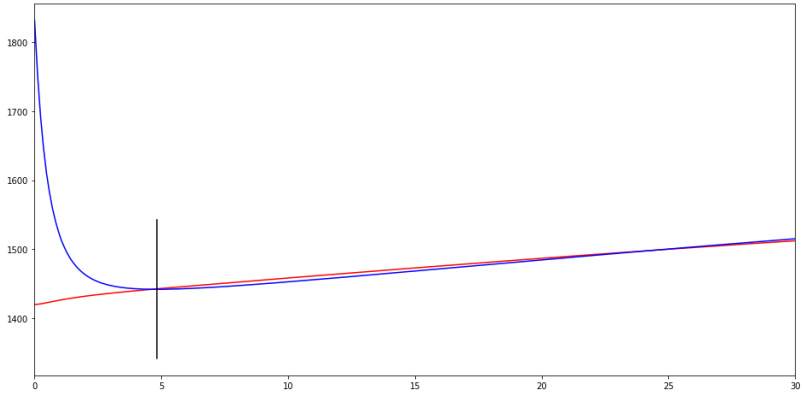
Also called Tikhonov regression

- Estimate diamond price given 9 features (carat, cut, colour, multiple for size)
- Linear model: Train RMSE $= 1420$; Test RMSE $= 1831$

Ridge regression II

Also called Tikhonov regression

- Estimate diamond price given 9 features (carat, cut, colour, multiple for size)
- Linear model: Train RMSE = 1420; Test RMSE = 1831
- Sweep: (x-axis = λ, y-axis = RMSE, blue = test, red = train)



- Best (black line): Train RMSE = 1443; Test RMSE = 1442

Lasso, ridge and elastic net

- Ridge regression: (L2 norm, without square root)

$$L(\theta) = \sum_{i=1}^{n} (y_i - (ax_i + b)))^2 + \lambda(a^2 + b^2)$$

Lasso, ridge and elastic net

- Ridge regression: (L2 norm, without square root)

$$L(\theta) = \sum_{i=1}^{n} (y_i - (ax_i + b)))^2 + \lambda(a^2 + b^2)$$

- Lasso regression: (L1 norm)

$$L(\theta) = \sum_{i=1}^{n} (y_i - (ax_i + b)))^2 + \lambda(|a| + |b|)$$

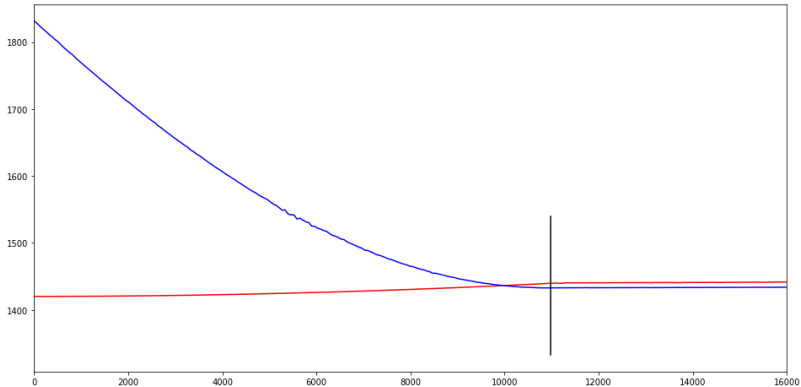
- Elastic net regression: (Blend of lasso and ridge)

$$L(\theta) = \sum_{i=1}^{n} (y_i - (ax_i + b)))^2 + \lambda \left(\gamma(|a| + |b|) + (1 - \gamma)(a^2 + b^2)\right)$$

(two hyperparameters: λ and γ)

Lasso regression

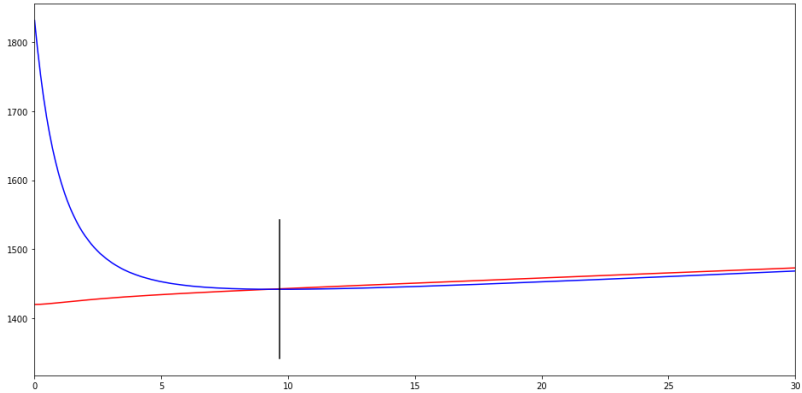
- Sweep: (x-axis = λ, y-axis = RMSE, blue = test, red = train)



- Best (black line): Train RMSE = 1440; Test RMSE = 1433

Elastic net regression

- $\gamma = 0.5$
- Sweep: (x-axis = λ, y-axis = RMSE, blue = test, red = train)



- Best (black line): Train RMSE = 1443; Test RMSE = 1442

Robustness

- Results (RMSE):
  - Linear: 1831
  - Ridge: 1442
  - Lasso: 1433
  - Elastic: 1442

- Other than linear all equivalent!

Robustness

- Results (RMSE):
  - Linear: 1831
  - Ridge: 1442
  - Lasso: 1433
  - Elastic: 1442
- Other than linear all equivalent!
- In practise: L1 more often better than L2
- Elastic-net lets hyperparameter optimisation decide
- There are more complex metrics, e.g. robust statistics

UNIVERSITY OF BATH

Probabilistic: Maximum likelihood

- Find model parameters that maximise data probability
  (need model to be probabilistic)
- No regularisation!
- Requires enough data to work

Linear regression: ML I

For each exemplar:

$$y_i = ax_i + b + \epsilon_i, \qquad \epsilon_i \sim N(0, \sigma^2)$$

$N(\texttt{mean}, \texttt{standard deviation}^2)$ is the Normal distribution
(simplest modification of linear regression to be probabilistic)

Linear regression: ML I

For each exemplar:

$$y_i = ax_i + b + \epsilon_i, \qquad \epsilon_i \sim N(0, \sigma^2)$$

$N(\texttt{mean}, \texttt{standard deviation}^2)$ is the Normal distribution
(simplest modification of linear regression to be probabilistic)
Exemplar probability:

$$P(y_i|x_i, a, b, \sigma) \propto \frac{1}{\sigma} \exp\left(\frac{-(ax_i + b - y_i)^2}{2\sigma^2}\right)$$

Linear regression: ML I

For each exemplar:

$$y_i = ax_i + b + \epsilon_i, \qquad \epsilon_i \sim N(0, \sigma^2)$$

$N(\texttt{mean}, \texttt{standard deviation}^2)$ is the Normal distribution
(simplest modification of linear regression to be probabilistic)
Exemplar probability:

$$P(y_i|x_i, a, b, \sigma) \propto \frac{1}{\sigma} \exp\left(\frac{-(ax_i + b - y_i)^2}{2\sigma^2}\right)$$

Maximum likelihood solution:

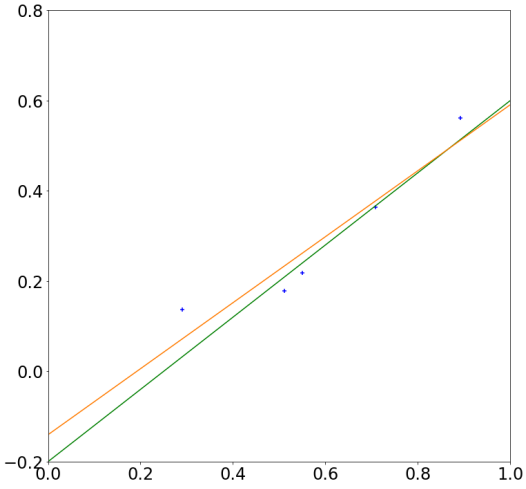
$$[a, b]^T = (X^T X)^{-1} X^T y$$

where

$$X = [[x_1, 1], [x_2, 1], \ldots, [x_n, 1]] \qquad y = [y_1, y_2, \ldots, y_n]^T$$

Given above know $\epsilon_i \therefore \sigma$ is mean of $|\epsilon_i|$

Linear regression: ML II



green = ground truth, orange = estimate

UNIVERSITY OF BATH

Probabilistic: Maximum a posteriori

- Introduce a **prior** over every model parameter
- prior = probability distribution

- Find maximum likelihood solution (again), including prior

- Model now complete – can generate answers **without** data!
- Works however much data you give it.

Linear regression: MAP I

For each exemplar:

$$y_i = ax_i + b + \epsilon_i, \qquad \epsilon_i \sim N(0, \sigma^2)$$

but add priors (one choice among many):

$$a, b \sim N(\mu_0, \Sigma_0), \qquad \sigma^2 \sim \texttt{Inv-Gamma}(\alpha_0, \beta_0)$$

where μ_0, Σ_0, α_0 and β_0 are hyper-parameters.

Linear regression: MAP I

For each exemplar:

$$y_i = ax_i + b + \epsilon_i, \qquad \epsilon_i \sim N(0, \sigma^2)$$

but add priors (one choice among many):

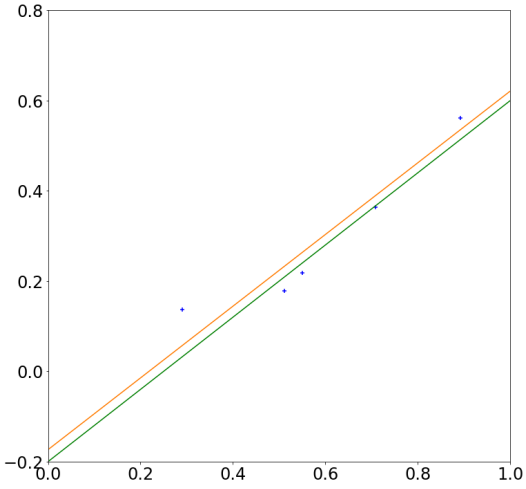
$$a, b \sim N(\mu_0, \Sigma_0), \qquad \sigma^2 \sim \texttt{Inv-Gamma}(\alpha_0, \beta_0)$$

where μ_0, Σ_0, α_0 and β_0 are hyper-parameters.
Answer:

$$[a, b]^T = (X^T X + \Sigma_0^{-1})^{-1}(\Sigma_0^{-1}\mu_0 + X^T y)$$

with same definitions of X and y as before.
Ignoring σ as complicated.

Linear regression: MAP II



green = ground truth, orange = estimate

Probabilistic: Bayesian

- Same as MAP, with priors.
- Instead of maximum likelihood solution find **posterior distribution**.

$$P(\texttt{model parameters}|\texttt{data}, \texttt{labels})$$

Probabilistic: Bayesian

- Same as MAP, with priors.
- Instead of maximum likelihood solution find **posterior distribution**.

$$P(\texttt{model parameters}|\texttt{data}, \texttt{labels})$$

- Has benefits of MAP.
- Plus a **distribution** over models – it knows how certain it is!
- Occam's razor is built in.

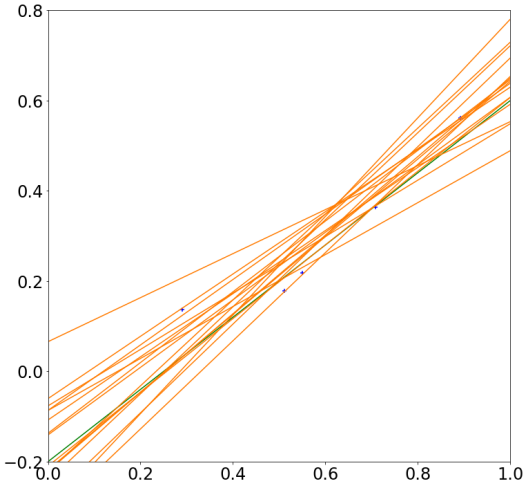
Linear regression: Bayesian I

Same formulation as MAP.
Answer:

$$[a, b]^T \sim N(\mu_n, \Sigma_n)$$
$$\mu_n = (X^T X + \Sigma_0^{-1})^{-1}(\Sigma_0^{-1}\mu_0 + X^T y)$$
$$\Sigma_n = \sigma^2(X^T X + \Sigma_0^{-1})^{-1}$$

Note: Dependent on σ, which has not been given.

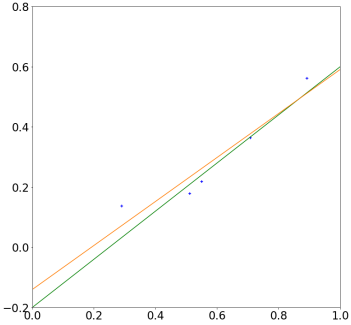
Linear regression: Bayesian II



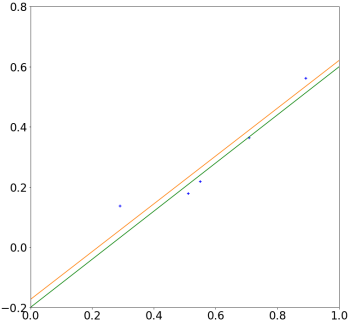
green = ground truth, orange = draws from estimate

Comparison

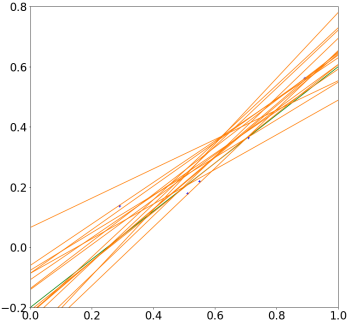
ML



MAP



Bayesian



- Given enough data they will all give the same answer
- Not enough:
  - Maximum likelihood fails
  - Maximum a posteriori gives a solution
  - Bayesian gives a solution and tells you how confident it is

UNIVERSITY OF BATH

Should all models be Bayesian?

- In an ideal world, yes!

UNIVERSITY OF BATH

Should all models be Bayesian?

- In an ideal world, yes!
- But...
  - Often harder to code and optimise
  - Often slower
  - Good prior problem...

Priors

- Purpose is to regularise – bias towards simple solutions.

Priors

- Purpose is to regularise – bias towards simple solutions.
- Do so by indicating which model parameters are likely, which unlikely
- Assumption that the model is sensible – that you can reason about its parameters
- e.g. a chaotic simulation would be almost impossible to set a prior for

Prior: Types

- Uninformative
- Improper
- Minimum description length

- Extra knowledge
- Data driven (dodgy)
- Human belief

Prior: Conjugate

- A prior that allows an analytic solution
- Choice of Gaussian and inverse Gamma for linear regression made answer analytic

Prior: Conjugate

- A prior that allows an analytic solution
- Choice of Gaussian and inverse Gamma for linear regression made answer analytic

- Problem: Conjugate priors are simple, bad match to data
- Bayesian methods often under perform due to using simple priors

UNIVERSITY OF BATH

Model kinds II

x – Data y – Label

UNIVERSITY OF BATH

Model kinds II

x – Data y – Label

Discriminative

Generative

Model kinds II

x – Data y – Label

Discriminative

- Learns $P(y|x)$

Generative

- Learns $P(y, x)$

Model kinds II

x – Data $\quad$ y – Label

Discriminative

- Learns $P(y|x)$
- Used directly

Generative

- Learns $P(y, x)$
- Apply Bayes rules: $P(y|x) = \frac{P(y,x)}{P(x)}$
- Often actually $P(x|y)$ and $P(y)$

UNIVERSITY OF BATH

Model kinds II

x – Data y – Label

Discriminative

- Learns $P(y|x)$
- Used directly

- Learns boundary between data
  (no requirement to be probabilistic)

Generative

- Learns $P(y, x)$
- Apply Bayes rules: $P(y|x) = \frac{P(y,x)}{P(x)}$
- Often actually $P(x|y)$ and $P(y)$

- Learns distribution of data
  (must be probabilistic)

Model kinds II

x – Data y – Label

Discriminative

- Learns $P(y|x)$
- Used directly

- Learns boundary between data
  (no requirement to be probabilistic)
- Can only discriminate between classes

Generative

- Learns $P(y, x)$
- Apply Bayes rules: $P(y|x) = \frac{P(y,x)}{P(x)}$
- Often actually $P(x|y)$ and $P(y)$

- Learns distribution of data
  (must be probabilistic)
- Can also generate data

UNIVERSITY OF BATH

Model kinds II

x – Data y – Label

Discriminative

- Learns $P(y|x)$
- Used directly

- Learns boundary between data
  (no requirement to be probabilistic)
- Can only discriminate between classes

Generative

- Learns $P(y, x)$
- Apply Bayes rules: $P(y|x) = \frac{P(y,x)}{P(x)}$
- Often actually $P(x|y)$ and $P(y)$

- Learns distribution of data
  (must be probabilistic)
- Can also generate data
- Handle missing data
- Less vulnerable to overfitting
- Know when they are unreliable

UNIVERSITY OF BATH

Should all models be generative?

- In an ideal world, yes!

UNIVERSITY OF BATH

Should all models be generative?

- In an ideal world, yes!
- But...
  - Often harder to code and optimise
  - Often slower
  - Discriminative approaches often "win"...

Discriminative vs generative

- If winning means highest accuracy: They keep switching places

Discriminative vs generative

- If winning means highest accuracy: They keep switching places
- Currently, discriminative is winning...
  ...but can already see generative successors
  (GANs, Auto-encoders)

UNIVERSITY OF BATH

Summary

- Regularisation embodies common sense
- Use it!

- Models can be probabilistic or not
- Probabilistic models have three main approaches (others exist)
- Models can be discriminative or generative

- Generative Bayesian models are the gold standard

Further reading

- Chapter 28, of "Information Theory, Inference, and Learning Algorithms" by MacKay.
- Maths for linear regression variants:
  `https://en.wikipedia.org/wiki/Bayesian_linear_regression`