An Introduction to Regularisation

John D. Kelleher
The ADAPT Centre for Digital Content Technology
School of Computing, Dublin Institute of Technology

Machine Learning Dublin Meetup
Bank of Ireland
Grand Canal Square
30^{th} May 2016

Outline

What is Machine Learning?

What is Machine Learning?

(Supervised) Machine Learning techniques automatically learn a model of the relationship between a set of **descriptive features** and a **target feature** from a set of historical examples.

What is Machine Learning?

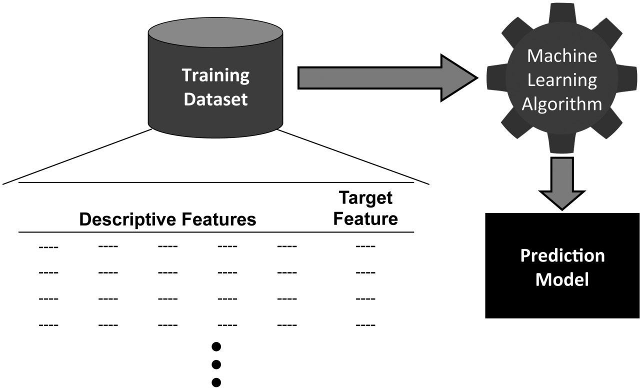


Figure: Using machine learning to induce a prediction model from a training dataset.

What is Machine Learning?

Figure: Using the model to make predictions for new query instances.

What is Machine Learning?

The **goal** of machine learning a model that **generalises** beyond the dataset and that isn't influenced by the noise in the dataset.

How Does Machine Learning Work?

How Does Machine Learning Work?

Machine learning algorithms work by **searching** through a set of possible prediction models for the model that best captures the relationship between the descriptive features and the target feature

How Does Machine Learning Work?

An obvious search criteria to drive this search is to look for models that are **consistent** with the training data.

How Does Machine Learning Work?

There are two problems with just searching for consistent models:

How Does Machine Learning Work?

There are two problems with just searching for consistent models:

1. Consistency $\approx$ memorizing the dataset

How Does Machine Learning Work?

There are two problems with just searching for consistent models:

1. Consistency $\approx$ memorizing the dataset
2. ML is **ill-posed**

How Does Machine Learning Work?

We need to bias our learning in order to select a single best model

How Does Machine Learning Work?

We need to bias our learning in order to select a single best model

Inductive bias:
the set of assumptions that define the model selection criteria of an ML algorithm

How Does Machine Learning Work?

There are two types of bias that we can use:

1. restriction bias
2. preference bias

What Can Go Wrong With ML?

What Can Go Wrong With ML?

There are two sources of information that guide our ML search for the best model:

1. the training data,
2. the inductive bias of the algorithm.

What Can Go Wrong With ML?

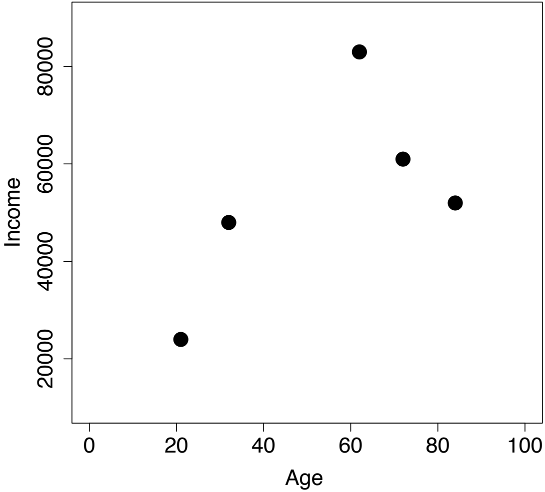
- What happens if we choose the wrong inductive bias:
  1. underfitting
  2. overfitting

What Can Go Wrong With ML?

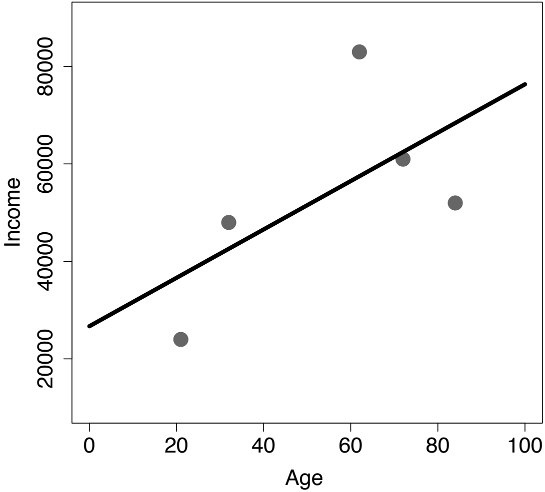
Table: The age-income dataset.

ID	Age	Income
1	21	24,000
2	32	48,000
3	62	83,000
4	72	61,000
5	84	52,000

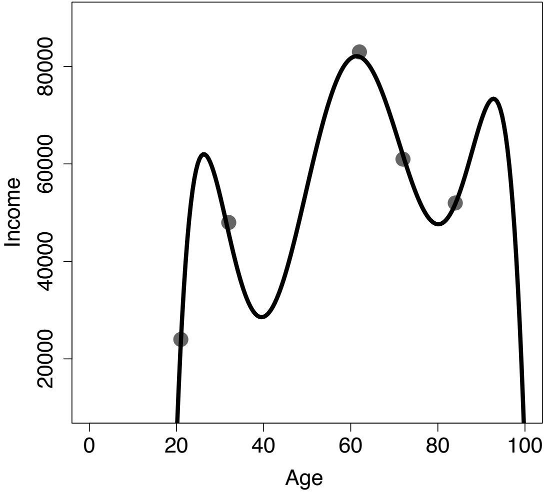
What Can Go Wrong With ML?



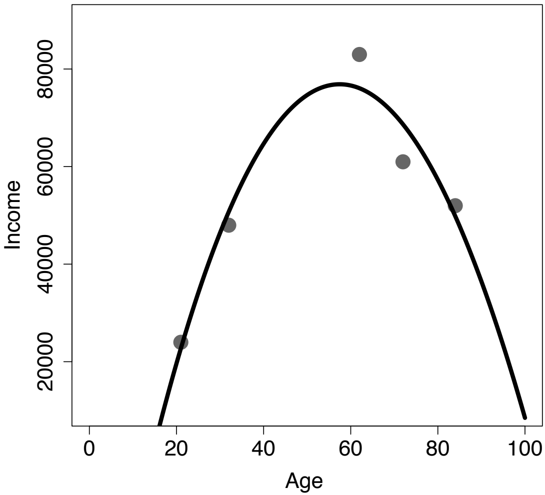
What Can Go Wrong With ML?



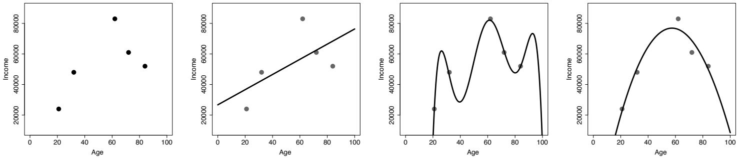
What Can Go Wrong With ML?



What Can Go Wrong With ML?



What Can Go Wrong With ML?



(a) Dataset (b) Underfitting (c) Overfitting (d) Just right

Figure: Striking a balance between overfitting and underfitting when trying to predict age from income.

Regularisation

Regularisation

- Regularisation is form of **preference bias** that is designed to help reduce **overfitting**

Weight Decay Regularisation

Weight Decay Regularisation

$$\mathbb{M}_{\mathbf{w}}(\mathbf{d}) = \mathbf{w}[0] + \mathbf{w}[1] \times \mathbf{d}[1] + \cdots + \mathbf{w}[n] \times \mathbf{d}[n]$$

Weight Decay Regularisation

$$\mathbb{M}_{\mathbf{w}}(\mathbf{d}) = \mathbf{w}[0] + \mathbf{w}[1] \times \mathbf{d}[1] + \cdots + \mathbf{w}[m] \times \mathbf{d}[m]$$

- Works by augmenting the loss function used during training so as to preference models that have small (close to zero) **weights** (coefficients) of the inputs

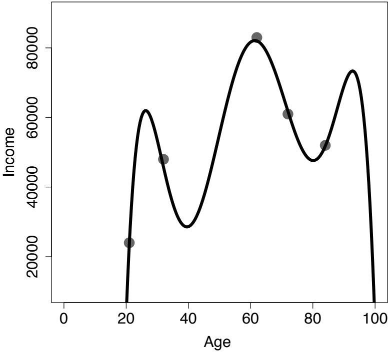
Weight Decay Regularisation

$$\mathbb{M}_{\mathbf{w}}(\mathbf{d}) = \mathbf{w}[0] + \mathbf{w}[1] \times \mathbf{d}[1] + \cdots + \mathbf{w}[m] \times \mathbf{d}[m]$$

- Generally we don't apply regularisation to the **intercept** (which is simply a measure of the mean value of the target when all the descriptive features equal 0)

Weight Decay Regularisation

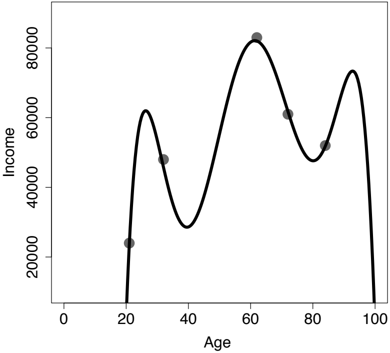
Why do we want small weights? (answer 1)



- Small changes in the input $\rightarrow$ big changes in the output

Weight Decay Regularisation

Why do we want small weights? (answer 1)



- Small changes in the input $\rightarrow$ big changes in the output
- Keeping weights small helps stop this (smooths the line)

Weight Decay Regularisation

Why do we want small weights? (answer 2)

- Making the algorithm preference models with small weights also reduces the variance between models that are trained on different versions of the dataset

Weight Decay Regularisation

Why do we want small weights? (answer 2)

- Making the algorithm preference models with small weights also reduces the variance between models that are trained on different versions of the dataset
- This means that the algorithms selection of models is less dependent on variation in the data $\rightarrow$ reduces the probability of overfitting to the noise in a specific version of the dataset

Weight Decay Regularisation

There are different ways to augment the loss function so as to preference small weights

1. Ridge Regression
2. Lasso

Ridge Regression

Ridge Regression

$$L_2(\mathbb{M}_{\mathbf{w}}, \mathcal{D}) = \sum_{i=1}^{n} (t_i - \mathbb{M}_{\mathbf{w}}(\mathbf{d}_i))^2$$

Ridge Regression

$$L_2(\mathbb{M}_{\mathbf{w}}, \mathcal{D}) = \sum_{i=1}^{n}(t_i - \mathbb{M}_{\mathbf{w}}(\mathbf{d}_i))^2 \quad \underbrace{+\lambda \sum_{j=1}^{n} w[j]^2}_{\textit{penalty on large weights}}$$

Ridge Regression

$$L_2(\mathbb{M}_{\mathbf{w}}, \mathcal{D}) = \sum_{i=1}^{n}(t_i - \mathbb{M}_{\mathbf{w}}(\mathbf{d}_i))^2 + \lambda \sum_{j=1}^{n} w[j]^2$$

λ is a tuning parameter (hyper-parameter) often set by cross-validation

Ridge Regression

$$L_2(\mathbb{M}_{\mathbf{w}}, \mathcal{D}) = \sum_{i=1}^{n} (t_i - \mathbb{M}_{\mathbf{w}}(\mathbf{d}_i))^2 + \lambda \sum_{j=1}^{n} w[j]^2$$

- $\lambda = 0$ penalty term has no effect and we end up with standard least squares estimates
- $\lambda \to \infty$ impact of penalty term grows pushing weights closer to zero

Ridge Regression

$$L_2(\mathbb{M}_{\mathbf{w}}, \mathcal{D}) = \sum_{i=1}^{n} (t_i - \mathbb{M}_{\mathbf{w}}(\mathbf{d}_i))^2 + \lambda \sum_{j=1}^{n} w[j]^2$$

- A potential drawback of ridge regression is that although it pushes weights to zero it doesn't set any of them to exactly zero

Ridge Regression

$$L_2(\mathbb{M}_{\mathbf{w}}, \mathcal{D}) = \sum_{i=1}^{n} (t_i - \mathbb{M}_{\mathbf{w}}(\mathbf{d}_i))^2 + \lambda \sum_{j=1}^{n} w[j]^2$$

- A potential drawback of ridge regression is that although it pushes weights to zero it doesn't set any of them to exactly zero
- Sometimes we would like to do **feature selection** so as to help with model interpretability

Lasso

Lasso

- Least absolute shrinkage and selection operator (Lasso)
- Implements both regularisation and feature selection

Lasso

$$L_2(\mathbb{M}_{\mathbf{w}}, \mathcal{D}) = \sum_{i=1}^{n} (t_i - \mathbb{M}_{\mathbf{w}}(\mathbf{d}_i))^2 \quad \underbrace{+\lambda \sum_{j=1}^{n} |w[j]|}_{\mathbf{L_1}\ \textit{shrinkage penalty}}$$

Lasso

$$L_2(\mathbb{M}_{\mathbf{w}}, \mathcal{D}) = \sum_{i=1}^{n} (t_i - \mathbb{M}_{\mathbf{w}}(\mathbf{d}_i))^2 \quad \underbrace{+\lambda \sum_{j=1}^{n} |w[j]|}_{\mathbf{L_1} \textit{ shrinkage penalty}}$$

The $\mathbf{L_1}$ penalty forces some of the weights to be exactly zero when λ is sufficiently large

Why does Lasso push weights to exactly zero?

Why does Lasso push weights to exactly zero?

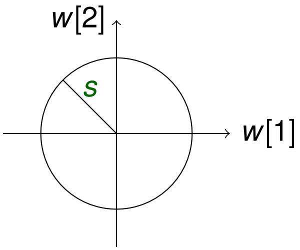
Ridge Regression

$$\arg\min_{\mathbf{w}} \sum_{i=1}^{n} (t_i - \mathbb{M}_{\mathbf{w}}(\mathbf{d}_i))^2 \text{ subject to } \sum_{j=1}^{m} \mathbf{w}[j]^2 \leq s$$

Lasso

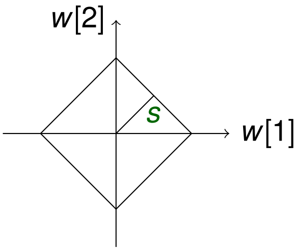
$$\arg\min_{\mathbf{w}} \sum_{i=1}^{n} (t_i - \mathbb{M}_{\mathbf{w}}(\mathbf{d}_i))^2 \text{ subject to } \sum_{j=1}^{m} |\mathbf{w}[j]| \leq s$$

Ridge Regression



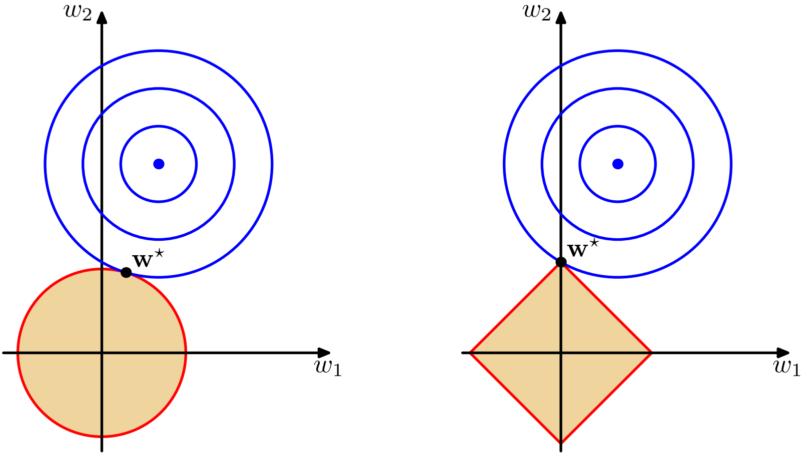
$$\sum_{j=1}^{m} \mathbf{w}[j]^2 \leq s$$

Lasso



$$\sum_{j=1}^{m} |\mathbf{w}[j]| \leq s$$

Why does Lasso push weights to exactly zero?



[1]

[1] These images are Figure 3.4 in *Pattern Recognition and Machine Learning* by Christopher Bishop (2006)

Summary

Summary

- Regularisation is a way to encode a preference bias into an ML algorithm that helps to avoid overfitting
- Weight decay (shrinkage methods) prefer regression models that have small weights (coefficients)
- Regularisation is most applicable in contexts where least squares estimates have high variance (e.g., small datasets)

Summary

- Ridge Regression and Lasso are just two methods of weight decay regularisation
- Lasso implicitly assumes that some of the weights should be zero (i.e., it implements feature selection)

Summary

- Ridge regression works best in contexts where you believe all the descriptive features are relevant and all are equally relevant

Summary

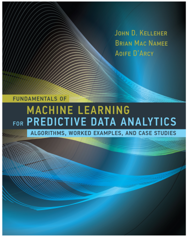
- Lasso implicitly assumes that some of the weights should be zero, so it works best in contexts where some of the descriptive features are irrelevant
- Lasso models are generally easier to interpret (some of the descriptive features are excluded)

Hiring

- We are currently looking to hire a Post-Doc to work on machine learning projects.
- We are also looking to recruit an MSc. candidate to work on a machine learning project on activity recognition.
- So if you are interested in either of these posts please email me: john.d.kelleher@dit.ie

Thank you for your attention!

john.d.kelleher@dit.ie
@johndkelleher
www.comp.dit.ie/jkelleher
www.machinelearningbook.com



Engaging Content
Engaging People