



Recap and Review – Exam 1

Applied Machine Learning
Derek Hoiem

Bias-Variance Trade-off

$$\underbrace{E_{\mathbf{x},y,D} \left[(h_D(\mathbf{x}) - y)^2 \right]}_{\text{Expected Test Error}} = \underbrace{E_{\mathbf{x},D} \left[(h_D(\mathbf{x}) - \bar{h}(\mathbf{x}))^2 \right]}_{\text{Variance}} + \underbrace{E_{\mathbf{x},y} \left[(\bar{y}(\mathbf{x}) - y)^2 \right]}_{\text{Noise}} + \underbrace{E_{\mathbf{x}} \left[(\bar{h}(\mathbf{x}) - \bar{y}(\mathbf{x}))^2 \right]}_{\text{Bias}^2}$$

Variance: due to limited data

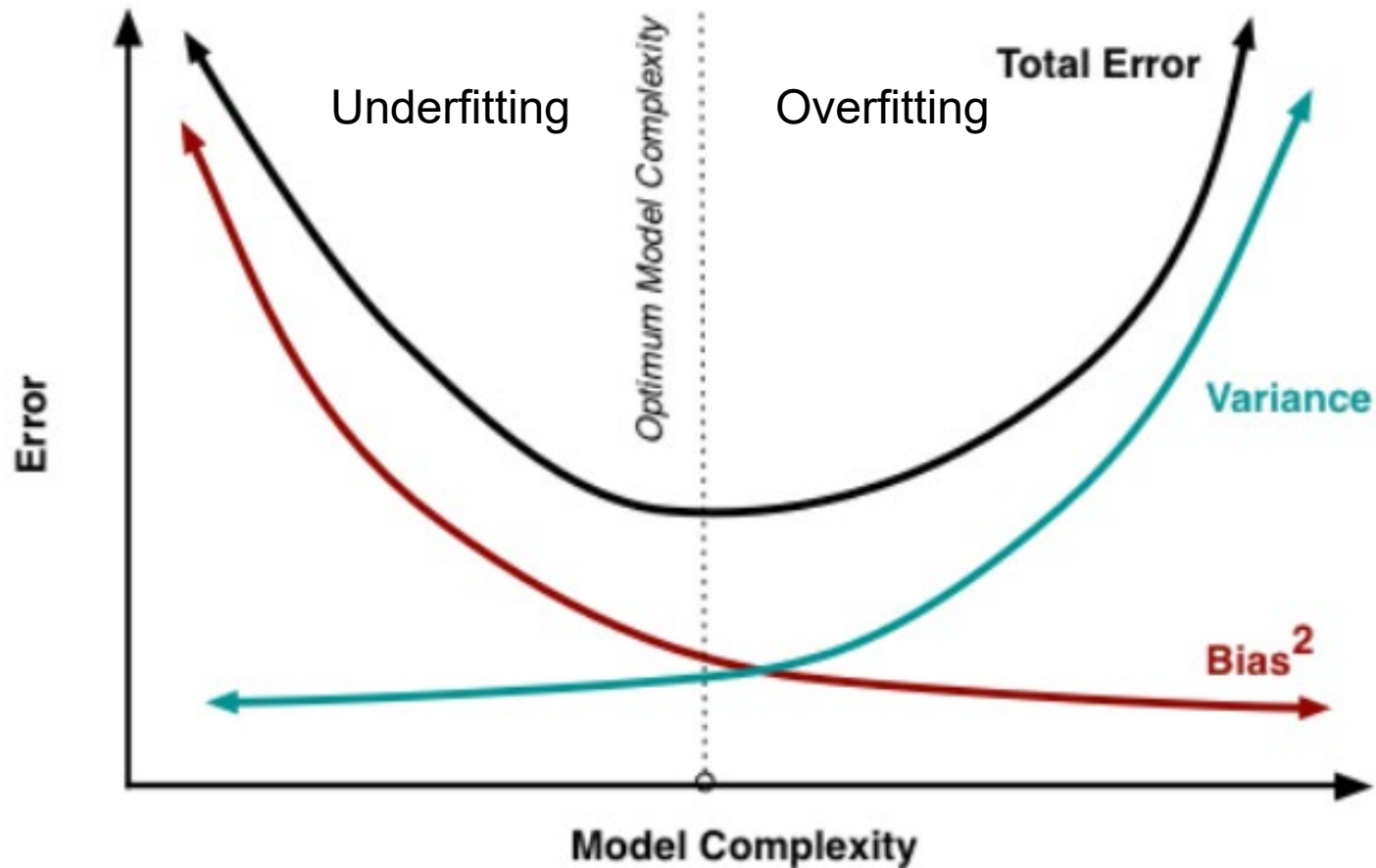
Different training samples will give different models that vary in predictions for the same test sample

“Noise”: irreducible error due to data/problem

Bias: error when optimal model is learned from infinite data

Above is for regression.

But same error = variance + noise + bias² holds for classification error and logistic regression.



How to detect high variance:

- Test error is much higher than training error

How to detect high bias or noise:

- The training error is high

As you increase model complexity:

- Training error will decrease
- Test error may decrease (if you are currently “underfitting”) or increase (if you are “overfitting”)

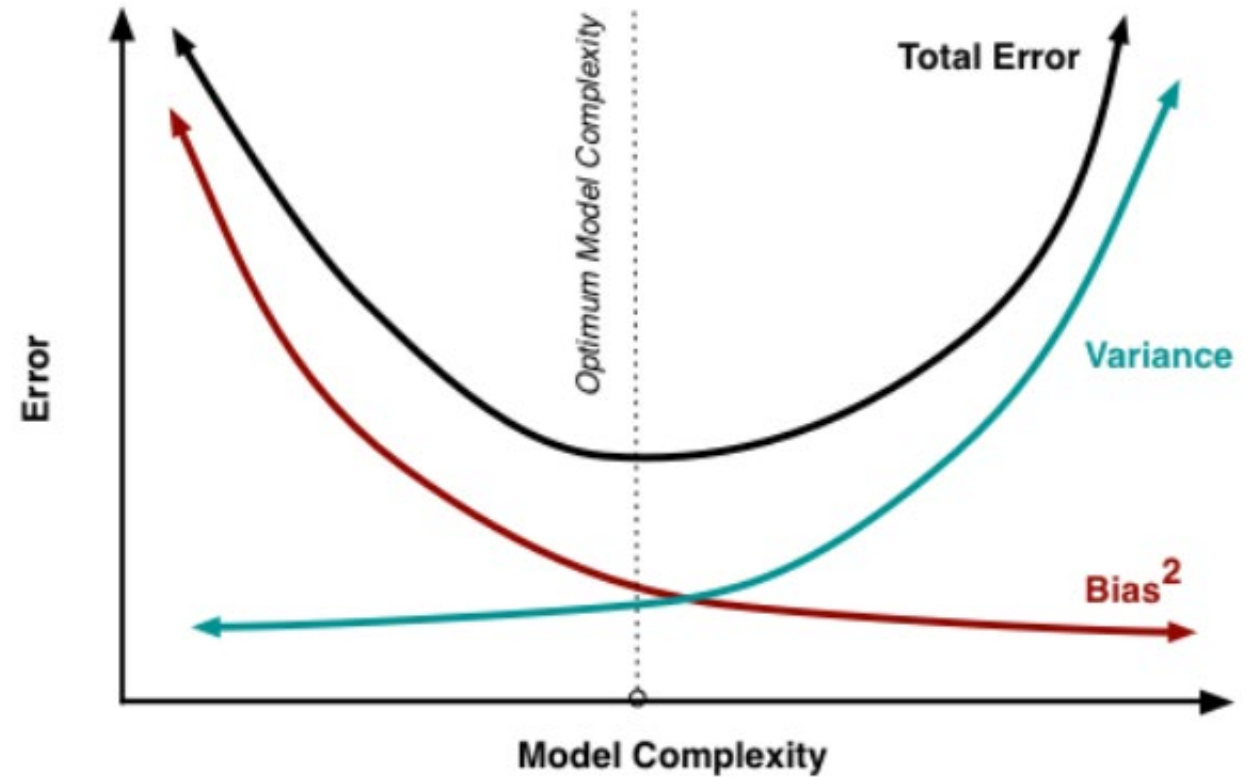
Bias-Variance / Generalization

Decreases variance

- Increase regularization weight
- Decrease model complexity
- Increase number of training examples

Decreases bias

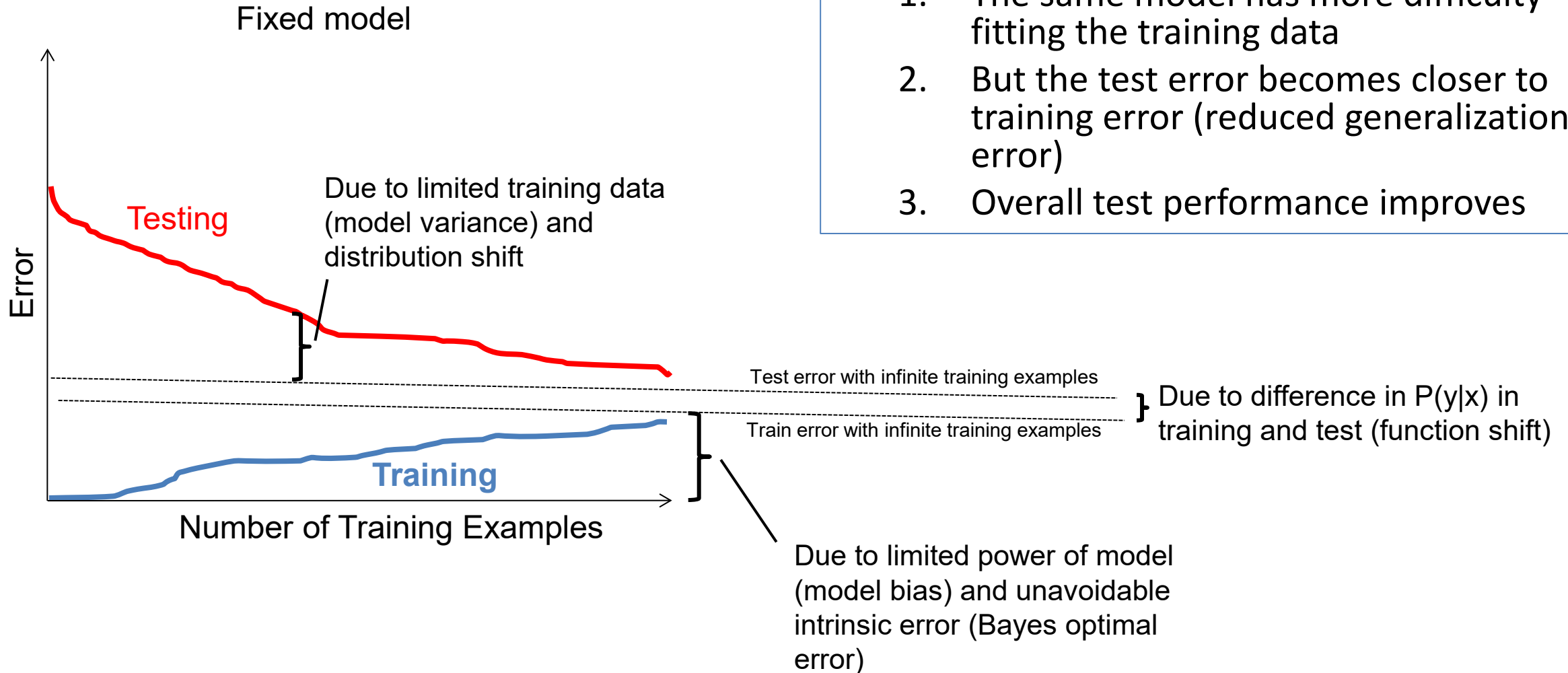
- Decrease regularization weight
- Increase model complexity, or select model that can better fit data



Performance vs training size

As we get more training data:

1. The same model has more difficulty fitting the training data
2. But the test error becomes closer to training error (reduced generalization error)
3. Overall test performance improves



How to think about a ML model

1. What are your inputs and outputs?
 - Are your features tabular or continuous or mixed?
 - Are you trying to predict a categorical label (classification) or a continuous value (regression)?
2. Do you expect a simple distance in features to be predictive, or that the features are linearly correlated with the target?
 - NN relies on example distance (can normalize or use representation learning ahead of time if needed)
 - Linear models learn feature weights, but rely on a direct (linear) relationship between features and target
3. How much training data do you have?
 - With more data, it's possible to use more complex models
4. How important is training time? Prediction time?
 - NN is fast in training, relatively slow in prediction compared to linear models

Classification methods

	Nearest Neighbor	Naïve Bayes	Linear Logistic Regression	Linear SVM
Objective	Instance-Based	Probabilistic	Probabilistic	Margin-based
Decision Boundary	Partition by example distance	Usually linear, assumes features conditionally independent	Linear	Linear
Model / Prediction	$i^* = \operatorname{argmin}_i \operatorname{dist}(X_{trn}[i], x)$ $y^* = y_{trn}[i^*]$	$y^* = \operatorname{argmax}_y \prod_i P(x_i y)P(y)$	$\omega^T x + b \approx \log \frac{P(y=1 x)}{P(y=0 x)}$ $y^* = \operatorname{argmax}_y P(y x)$	$\operatorname{score}(y=1) = \mathbf{w}^T \mathbf{x} + b$
Strengths	* Low bias * No training time * Widely applicable * Simple	* Estimate from limited data * Simple * Fast training/prediction	* Powerful in high dimensions * Widely applicable * Good confidence estimates * Fast prediction	* Powerful in high dimensions * Widely applicable * Good generalization theory * Fast prediction
Limitations	* Relies on good and normalized input features * Relatively slow prediction	* Limited modeling power	* Relies on good input features	* Relies on individually useful input features

Regression methods

	Nearest Neighbor	Naïve Bayes	Linear Regression
Objective	Instance-Based	Probabilistic	Data fit
Decision Boundary	Partition by example distance	Usually linear, assumes features conditionally independent	Linear
Model / Prediction	$i^* = \operatorname{argmin}_i \operatorname{dist}(X_{trn}[i], x)$ $y^* = y_{trn}[i^*]$	$y^* = \operatorname{argmax}_y \prod_i P(x_i y)P(y)$	$y^* = w^T x$
Strengths	* Low bias * No training time * Widely applicable * Simple	* Estimate from limited data * Simple * Fast training/prediction	* Powerful in high dimensions * Widely applicable * Fast prediction * Coefficients may be interpretable
Limitations	* Relies on good and normalized input features * Relatively slow prediction	* Limited modeling power	* Relies on good input features

Training a model

$$\theta^* = \operatorname{argmin}_{\theta} \operatorname{Loss}(f(\mathbf{X}; \theta), \mathbf{y})$$

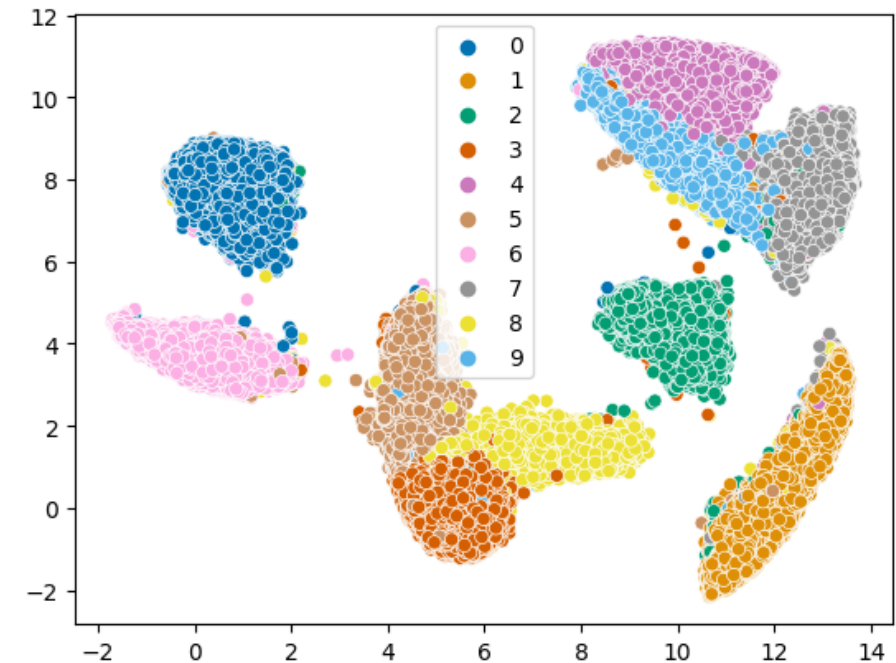
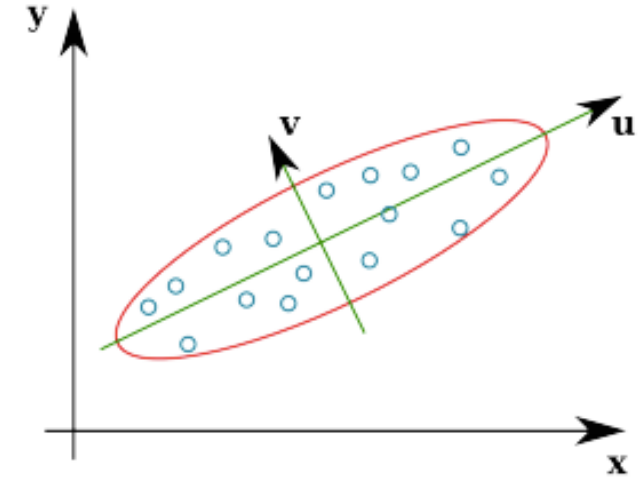
- $f(\mathbf{X}; \theta)$: the model, e.g. $y = \mathbf{w}^T \mathbf{x}$
- θ : parameters of the model (e.g. $\mathbf{w}$)
- $(\mathbf{X}, \mathbf{y})$: pairs of training samples
- $\operatorname{Loss}()$: defines what makes a good model
 - Good predictions, e.g. minimize $-\sum_n \log P(y_n | \mathbf{x}_n)$
 - Likely parameters, e.g. minimize $\mathbf{w}^T \mathbf{w}$
 - Regularization and priors indicate preference for particular solutions, which tends to improve generalization (for well chosen parameters) and can be necessary to obtain a unique solution

Model evaluation process

1. Collect/define training, validation, and test sets
2. Decide on some candidate models and hyperparameters
3. For each candidate:
 - a. Learn parameters with **training set**
 - b. Evaluate trained model on the **validation set**
4. Select best model
5. Evaluate best model's performance on the **test set**
 - Cross-validation can be used as an alternative
 - Common measures include error or accuracy, root mean squared error, precision-recall

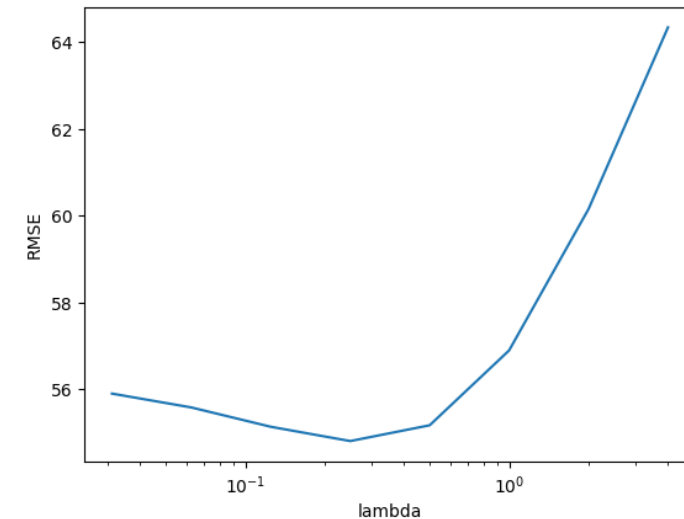
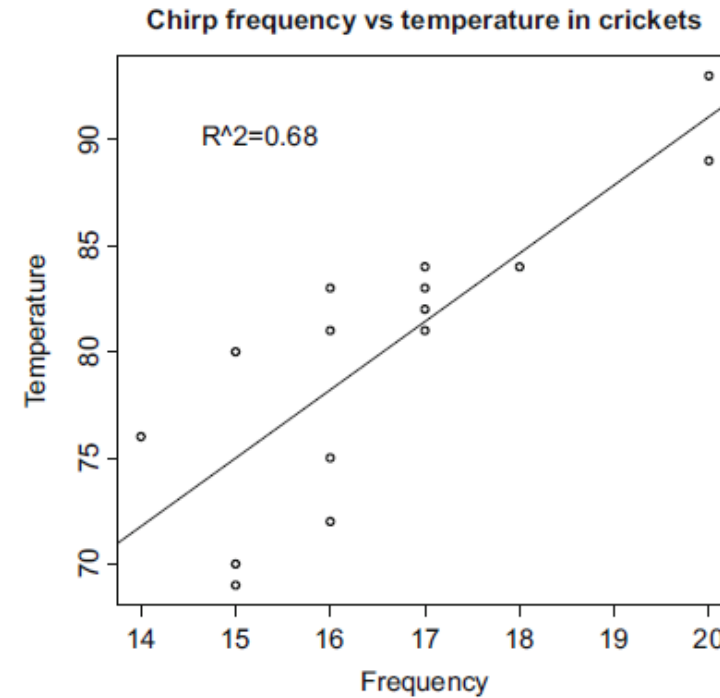
PCA/Embedding (L5)

- PCA reduces dimensions by linear projection
 - Preserves variance to reproduce data as well as possible, according to mean squared error
 - May not preserve local connectivity structure or discriminative information
- Other methods try to preserve relationships between points
 - MDS: preserve pairwise distances
 - IsoMap: MDS but using a graph-based distance
 - t-SNE: preserve a probabilistic distribution of neighbors for each point (also focusing on closest points)
 - UMAP: incorporates k-nn structure, spectral embedding, and more to achieve good embeddings relatively quickly



Linear Regression (L6)

- Linear regression fits a linear model to a set of feature points to predict a continuous value
 - Explain relationships
 - Predict values
 - Extrapolate observations
- Regularization prevents overfitting by restricting the magnitude of feature weights
 - L1: prefers to assign a lot of weight to the most useful features
 - L2: prefers to assign smaller weight to everything



Practice questions

<https://tinyurl.com/AML441-Exam1Review>



All in-class practice questions

<https://tinyurl.com/AML441-AllQs>

