- CV_M6_L01

Regression

CS/DSA 5970: Machine Learning Practice

Regression

High-level problem definition:

- Supervised learning problem
- In general, inputs can be numerical or categorical data
  - For now, our focus is on numerical inputs
- Outputs are numerical

Regression

Error metrics

- Generally: a function of the difference between ground truth and predicted values
- Common:
  - Sum squared error (or mean squared error)
  - Sum absolute error (or mean absolute error)

IPAD_M6_L01b

- Formulation of linear model
  - Scalar and vector formulations
  - With and without bias; incorporating bias term into input vector
- Graphical representation of the problem
- Error metrics
  - Mean squared error. rmse
  - Mean absolute error
- Solutions
  - Normal equation
  - Gradient descent

- CV_M6_L02

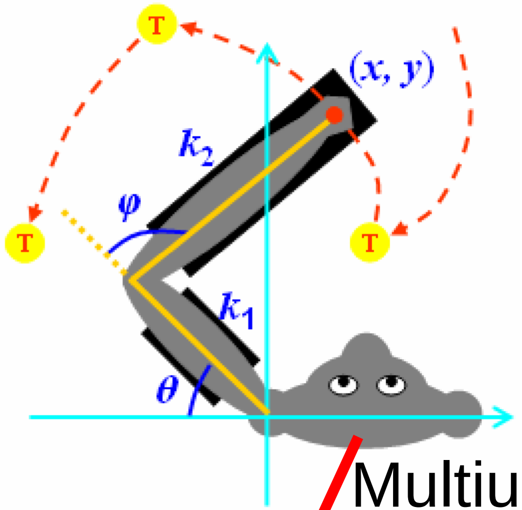
Brain-Machine Interface Problem

CS/DSA 5970: Machine Learning Practice

Brain-Machine Interfaces

Estimate of intended movement

Command prosthetic arm



Multiunit recording

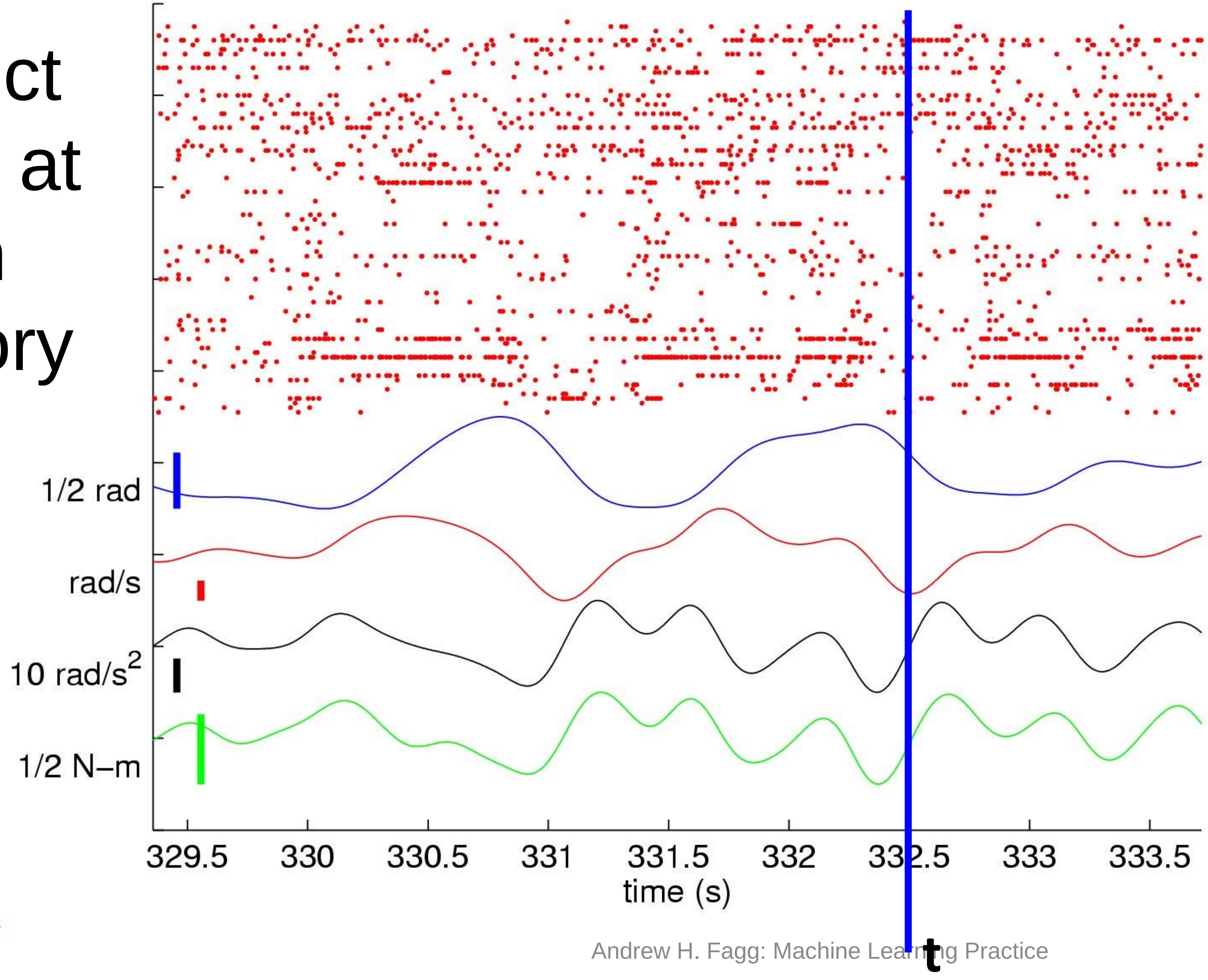
Predictive model

The UNIVERSITY of OKLAHOMA

In collaboration with Nicholas G. Hatsopoulos and Lee E. Miller

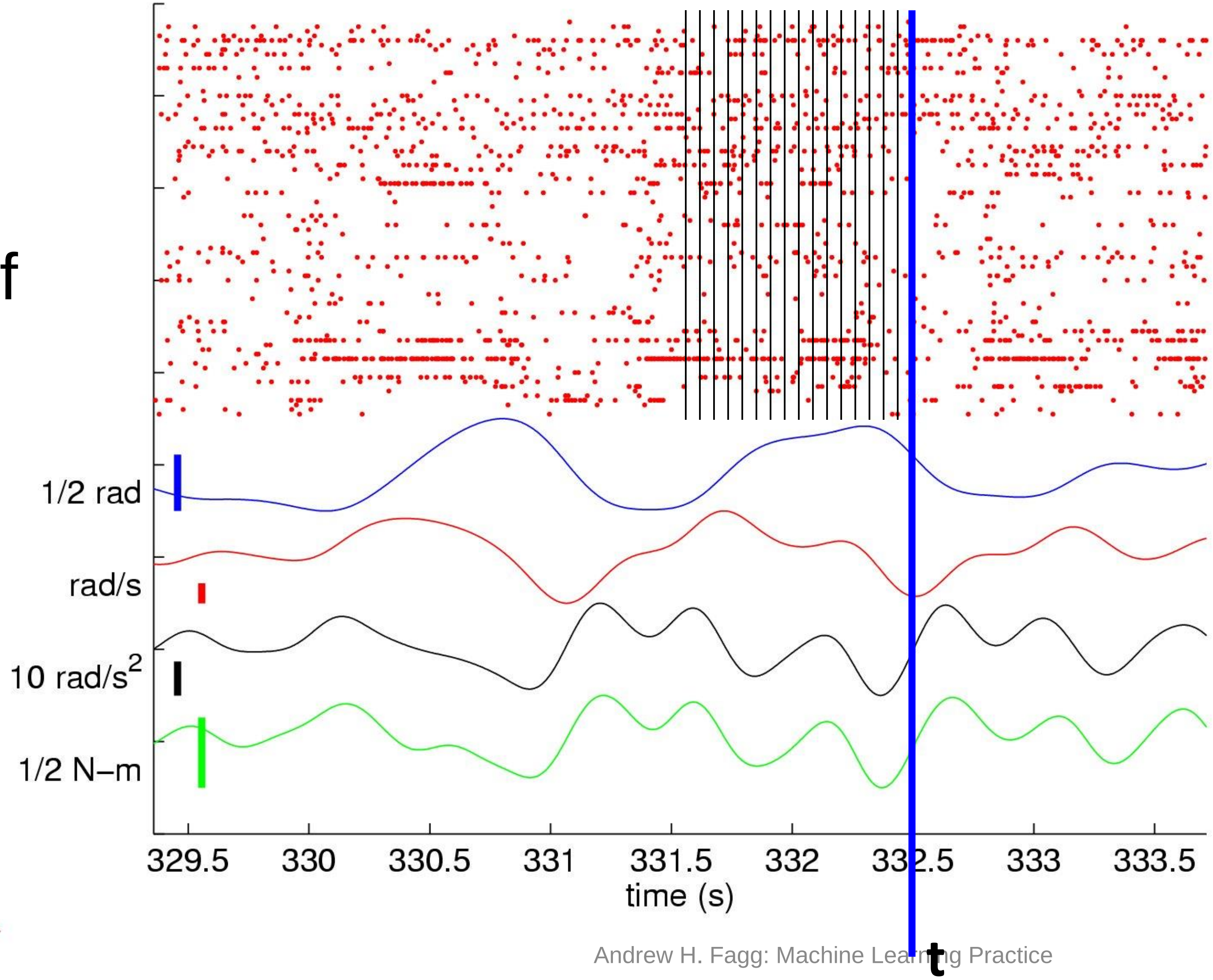
Decoding Arm State

Want to predict arm motion at time t given recent history of spiking behavior



Decoding Arm State

50ms bins: 20 descriptors of neural activation for each cell



The UNIVERSITY of OKLAHOMA

BMI Data Configuration

- Data already cut into 20 independent folds
- Time is continuous, but with gaps
  - We kept only valid time periods
- Each row in the CSV file contains 20 spike counts for each neuron
  - Each count corresponds to 50ms of time
  - A single row is a contiguous set of samples (no gaps!)

- CV_M6_L03

Example: BMI Data

CS/DSA 5970: Machine Learning Practice

The UNIVERSITY of OKLAHOMA

Live demo

- Loading and organizing the BMI data

- CV_M6_L04

Example: Predicting Arm Motion

CS/DSA 5970: Machine Learning Practice

Live demo

- LinearRegression example
  - Training set performance: plot and aggregate statistics
  - Test set performance

- CV_M6_L05

Gradient Descent Methods

CS/DSA 5970: Machine Learning Practice

The UNIVERSITY of OKLAHOMA

Limits of the Normal Equation

- The "Normal Equation" requires the inversion of an N+1 x N+1 matrix, where N is the number of features
- This can be really expensive as N becomes large
  - And unnecessary if the features are rather sparse

Gradient Descent Methods

Gradient Descent Approach:

- Guess at an initial set of parameters
- Update the parameters in a direction so that the error metric is lowered
- Repeat until error is low enough or stops improving

Gradient Descent Challenges

- It is hard to tell *a priori* how many steps will be necessary
- Unclear what the "learning rate" should be
- Computing the gradient of the error with respect to the parameters:
  - Computation of the gradient is done for each training sample
  - These gradients are then summed together to estimate the global gradient
  - This is ***Batch Gradient Descent***
  - If the training set is large, then this is a computationally expensive process

IPAD_M6_L05b

- Error surface
- Learning rate: too low to too high
- Different parts of the surface have different shapes
- Each sample "tugs on" the weights in some direction
  - Direction of movement in gd is the sum of these tugs
  - Want a way to estimate this sum without computing all of the tugs

Estimating the Gradient

- Stochastic Gradient Descent
  - Randomly select a single training example, compute the gradient and update the parameters
- Mini-Batch Gradient Descent
  - Cut the training set into batches
  - Use one batch at a time to compute gradient and update parameters
  - Cycle through these batches
- Stochastic Mini-Batch
  - Each training step: sample M training examples & use these to compute the gradient and update parameters

- CV_M6_L06

Example: Gradient Descent Methods

CS/DSA 5970: Machine Learning Practice

Live demo

- Stochastic
- Batch
- Stochastic mini-batch

- CV_M6_L07

Example: Training Sensitivity

CS/DSA 5970: Machine Learning Practice

Number of Training Steps

How many training steps do we need for a given problem?

- This is an empirical question
- Can visualize using a learning curve
  - Take a small step
  - Record performance on a training set and a validation set
  - Repeat

Training Set Size

With our first regression-based models:

- Performance with the training set was high
- But, performance with an independent data set was generally quite poor
- In our problem, this is due to a dramatic over-fit of the training data
  - Note: 961 parameters and only 1193 samples

Training Set Size

Whenever we face a new problem, it is ***very important*** to ask the question of whether we have enough training data

- One approach: train a model with varying amounts of training data & ask how the model performs on an independent data set
- Sensitive to training set size: you are overfitting and need more data
- Insensitive: you have plenty of training data

Note that this is a model-specific (and hyper-parameter-specific) question

Live demo

- CV_M6_L08

Multi-Regression

CS/DSA 5970: Machine Learning Practice

Multi-Regression

- So far, our models have only predicted a single output value for a given input
- In practice, we would like to handle entire vectors

Multi-Regression

Multi-regression is a generalization of regression

- Multiple outputs
- For our linear models, the parameters are completely separate from one-another
- Error metric is the sum of errors across the individual outputs

IPAD_M6_L08b

- input / output pair notation
- Math behind the model
- Mean squared error:
  - Simple sum
  - Can be weighted sum

The UNIVERSITY of OKLAHOMA

- CV_M6_L09

Example: Multi-Regression

CS/DSA 5970: Machine Learning Practice

Live demo

- Predict two velocities

- CV_M6_L10

Utility and Limits of Linear Regression

CS/DSA 5970: Machine Learning Practice

Linear Regression

Utility:

- Inexpensive to evaluate models
- Can compute the solution to a problem directly ("Normal Equation")
- Gradient descent approach is straight-forward and relatively inexpensive computationally
- There is only one minimum in the error space

Linear Regression

Limits:

- The world is rarely linear
- Would like to capture non-linear effects
- Would also like to constrain the output to match our expectations of the valid range of outputs
  - For example, if we are trying to output a probability

Next Steps in Regression

- Non-linear preprocessing of input features
  - Otherwise, the model is linear
- Non-linear on the output of the model
  - Otherwise, the model is linear
  - Logistic regression
- Non-linearities built into the model throughout

IPAD_M6_L10b

Non-Linear Preprocessing

CS/DSA 5970: Machine Learning Practice

IPAD_M6_L10b

- General formulation: transformation of one vector to another
- Polynomial features
- Cosine features

- CV_M6_L11

Example: Non-Linear Preprocessing

CS/DSA 5970: Machine Learning Practice

Live demo

- Polynomial transformation of the neural data

- CV_M7_L01

The Overfitting Problem

CS/DSA 5970: Machine Learning Practice

Overfitting

- Any situation where a model performs well on a training set, but not on an independent data set drawn from the same distribution as the training set
- In this case, the learned model has captured the peculiarities of the training set, but not the general trend of the entire distribution
- Detecting this situation is done by comparing model performance on training and independent data

Sources of Overfitting (or Apparent Overfitting)

- Training set is too small relative to the complexity of the model that is being fit
  - One clue: # of samples ~ # of model parameters
- Training set samples are not drawn independently
- Training data not actually drawn from the same distribution as the rest of the data

IPAD_M7_L01b

- Overfitting polynomial example
- Example small random variation in an input dimension
  - Narrow Gaussian
  - Show a sampling that does not touch a tail
- LMS algorithm wants to fit a line with a very high slope
  - Show sample from tail
- Goal: want to limit the slopes
  - More generally: want functions that do not change rapidly

- CV_M7_L02

Regularization

CS/DSA 5970: Machine Learning Practice

Regularization

Approach: add terms to our cost function that punish models that have large coefficients

IPAD_M7_L02b

- Linear model
- MSE
- Matrix math computation
- Ridge Regression
- Lasso
- Elastic Net

Regularization

- LMS: happy with high coefficients
- Ridge: wants to make coefficients small, especially ones that are already large
  - But, is happy to have very small coefficients
- Lasso: wants to make coefficients small
  - Wants to make as many coefficients zero as possible
- Elastic Net: also wants to make coefficients small
  - Can walk smoothly between the Ridge and Lasso solutions

Example: Regularization

- CV_M7_L03

Example: Regularization

CS/DSA 5970: Machine Learning Practice

The UNIVERSITY of OKLAHOMA

Regularization

- Simple regression problem
- Compare Ridge, Lasso and Elastic Net Solutions

- Live demo

Example: Regularization in the BMI Problem

- CV_M7_L04

Example: Regularization in the BMI Problem

CS/DSA 5970: Machine Learning Practice

Regularization in the BMI Problem

- We have already shown that LMS does not perform well with small training data set sizes
- How does regularization help with small training sets?

Example: Regularization in the BMI Problem

- CV_M7_L04b

- Live demo