# UVA CS 4501: Machine Learning

# Lecture 8: Review of Regression

Dr. Yanjun Qi

University of Virginia

Department of
Computer Science

# Where are we ? ➔ Five major sections of this course

- ❑ Regression (supervised)
- ❑ Classification (supervised)
- ❑ Unsupervised models
- ❑ Learning theory
- ❑ Graphical models

# Lecture 3

- ❑ Linear regression (aka **least squares**)
- ❑ Learn to derive the least squares estimate by normal equation
- ❑ Evaluation with Cross-validation

# Lecture-4

- ❑ More ways to train / perform optimization for linear regression models
  - ✓ Review: Gradient Descent
  - ✓ Gradient Descent (GD) for LR
  - ✓ Stochastic GD (SGD) for LR

# Lecture-5

- ❑ Regression Models Beyond Linear
  - ✓LR with non-linear basis functions
  - ✓Instance-based Regression: K-Nearest Neighbors
  - ✓Locally weighted linear regression
  - ✓Regression trees and Multilinear Interpolation (later)

# Lecture-6

## ❑ Linear Regression Model with Regularizations

- ✓ Review: (Ordinary) Least squares: squared loss (Normal Equation)
- ✓ Ridge regression: squared loss with L2 regularization
- ✓ Lasso regression: squared loss with L1 regularization
- ✓ Elastic regression: squared loss with L1 AND L2 regularization
- ✓ WHY and Influence of Regularization Parameter

# Lecture-7

- ❑ Feature Selection

✓ General Introduction

✓ Filtering

✓ Wrapper

✓ Embedded Method

# Machine Learning in a Nutshell

Task

↓

Representation

↓

Score Function

↓

Search/Optimization

↓

Models, Parameters

# Multivariate Linear Regression

| | |
|---|---|
| Task | Regression |
| ↓ | ↓ |
| Representation | Y = Weighted linear sum of X's |
| ↓ | ↓ |
| Score Function | Least-squares / GD / SGD |
| ↓ | ↓ |
| Search/Optimization | Linear algebra |
| ↓ | ↓ |
| Models, Parameters | Regression coefficients |

$$\hat{y} = f(x) = \theta^T x$$

# Multivariate Linear Regression with basis Expansion

Task → Representation → Score Function → Search/Optimization → Models, Parameters

Regression → Y = Weighted linear sum of (X basis expansion) → SSE → Linear algebra /GD /SGD → Regression coefficients

$$\hat{y} = \theta_0 + \sum_{j=1}^{m} \theta_j \varphi_j(x) = \varphi(x)^T \theta$$

# K-Nearest Neighbor

| | |
|---|---|
| Task | Regression/ classification |
| ↓ | ↓ |
| Representation | Local Smoothness |
| ↓ | ↓ |
| Score Function | NA |
| ↓ | ↓ |
| Search/Optimization | NA |
| ↓ | ↓ |
| Models, Parameters | Training Samples |

# Locally Weighted / Kernel Linear Regression

| | |
|---|---|
| Task | Regression |
| Representation | Y = Weighted linear sum of X' s |
| Score Function | Weighted SSE |
| Search/Optimization | Linear algebra |
| Models, Parameters | Local Regression coefficients (conditioned on each test point) |

$$\boxed{\theta^*(\mathbf{x}_0) = (B^T W(x_0) B)^{-1} B^T W(x_0) y}$$

$$\min_{\alpha(x_0),\beta(x_0)} \sum_{i=1}^{N} K_\lambda(x_0, x_i)[y_i - \alpha(x_0) - \beta(x_0)x_i]^2$$

$$\hat{f}(x_0) = \hat{\alpha}(x_0) + \hat{\beta}(x_0)x_0$$

# Regularized multivariate linear regression

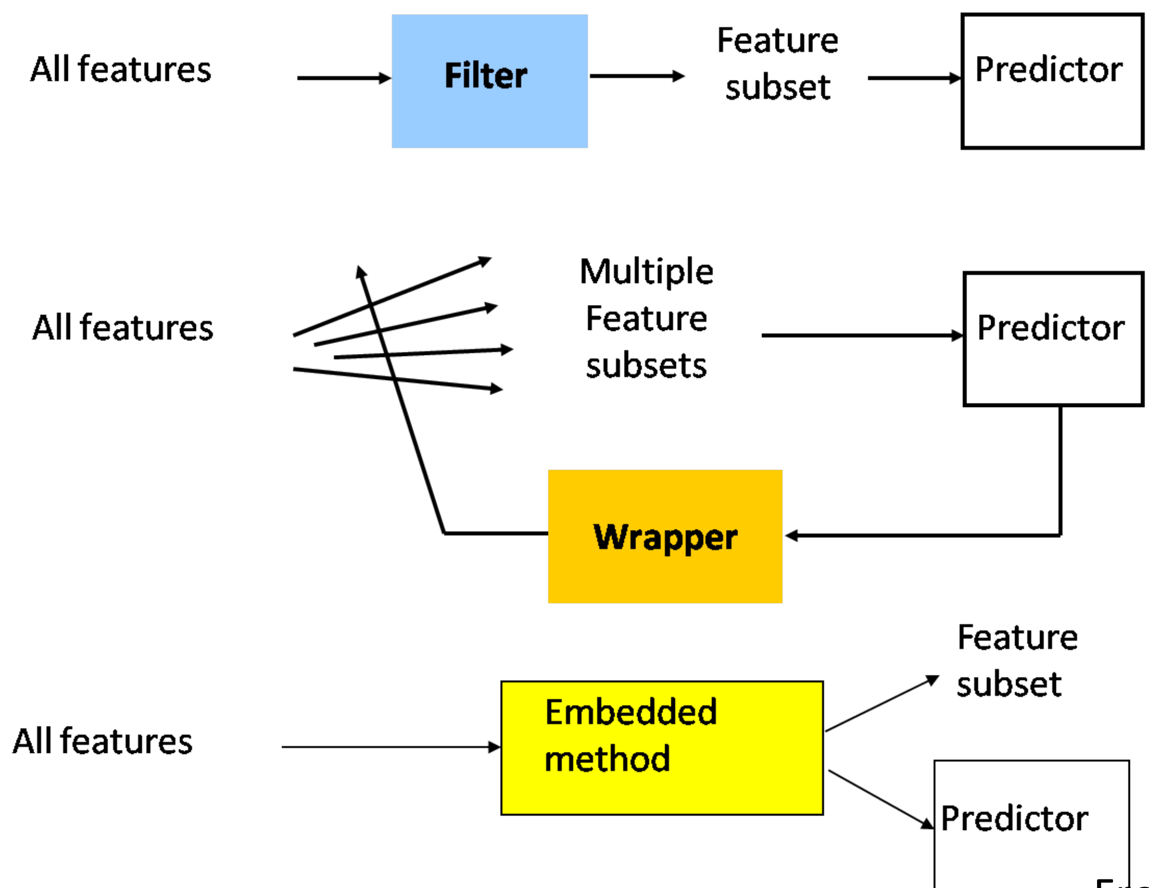
Task → Representation → Score Function → Search/Optimization → Models, Parameters

Regression → Y = Weighted linear sum of X’ s → Least-squares + Regularization → Linear algebra for Ridge / sub-GD for Lasso & Elastic → Regression coefficients (regularized weights)

$$\min J(\beta) = \sum_{i=1}^{n}\left(Y - \hat{Y}\right)^2 + \lambda\left(\sum_{j=1}^{p}\beta_j^q\right)^{1/q}$$

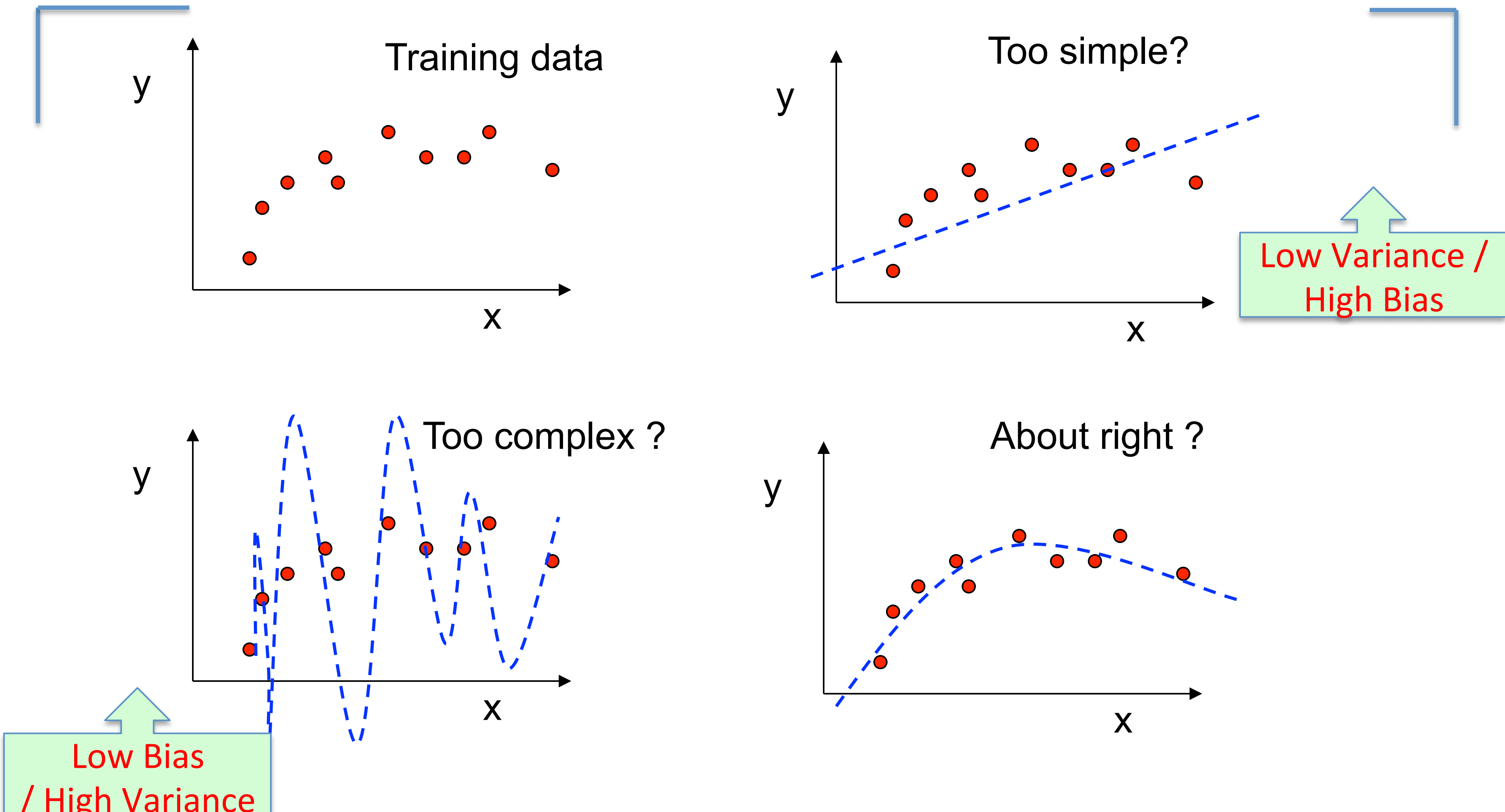
# Feature Selection: filters vs. wrappers vs. embedding

- **Main goal**: rank subsets of useful features




From Dr. **Isabelle Guyon**

# Complexity versus Goodness of Fit: Model Selection




What ultimately matters: ***GENERALIZATION***

Make sure that the train/test/validation folds are indeed independent samples.

# e.g. By k=10 fold Cross Validation

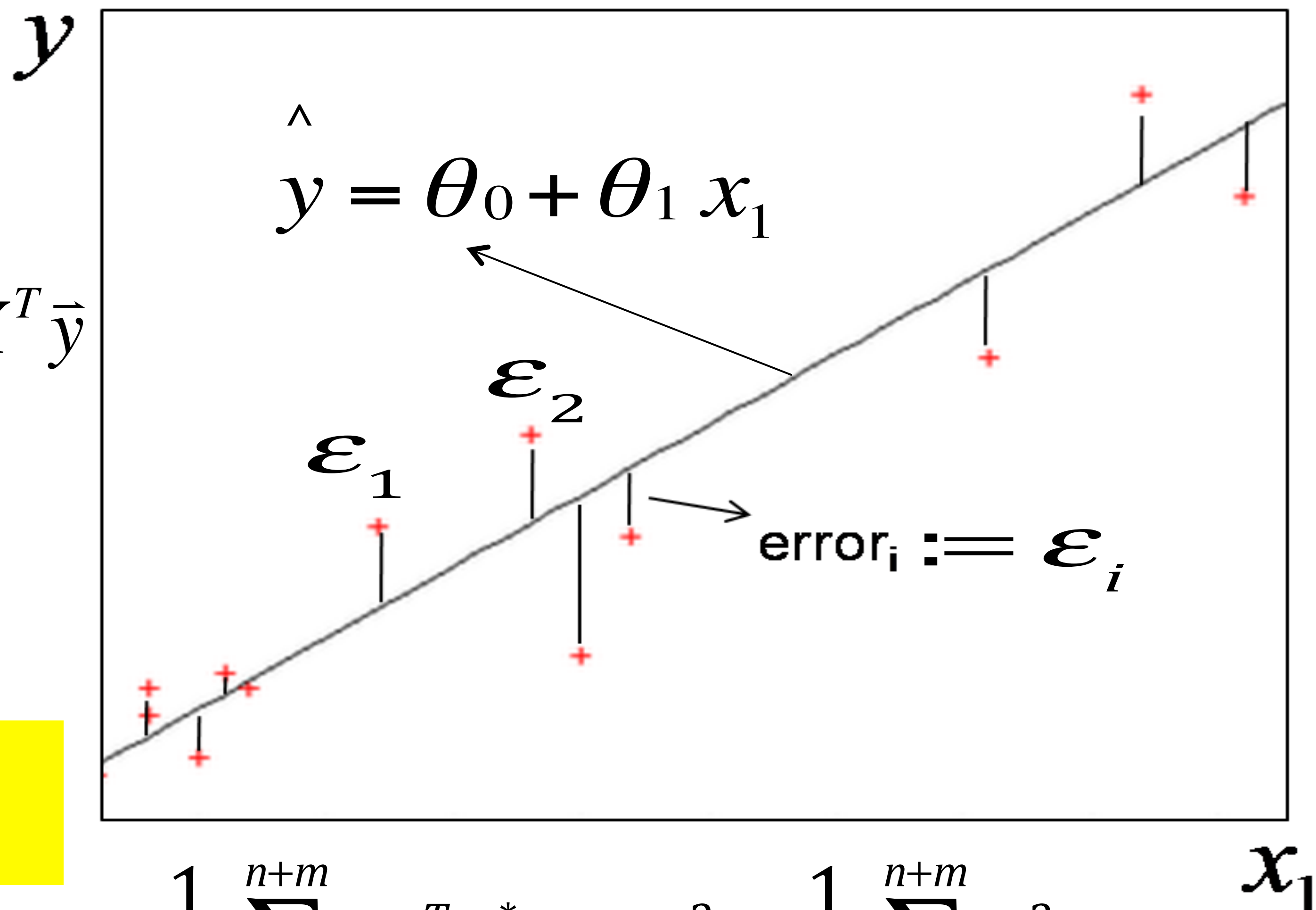
- Divide data into 10 equal pieces
- 9 pieces as training set, the rest 1 as test set
- Collect the scores from the diagonal
- We normally use the mean of the scores

| model | P1 | P2 | P3 | P4 | P5 | P6 | P7 | P8 | P9 | P10 |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | train | train | train | train | train | train | train | train | train | test |
| 2 | train | train | train | train | train | train | train | train | test | train |
| 3 | train | train | train | train | train | train | train | test | train | train |
| 4 | train | train | train | train | train | train | test | train | train | train |
| 5 | train | train | train | train | train | test | train | train | train | train |
| 6 | train | train | train | train | test | train | train | train | train | train |
| 7 | train | train | train | test | train | train | train | train | train | train |
| 8 | train | train | test | train | train | train | train | train | train | train |
| 9 | train | test | train | train | train | train | train | train | train | train |
| 10 | test | train | train | train | train | train | train | train | train | train |

# Evaluation

## e.g. Regression (1D example)

$$\theta^* = \left(X^T X\right)^{-1} X^T \vec{y}$$




Testing MSE Error to report:

$$J_{test} = \frac{1}{m} \sum_{i=n+1}^{n+m} (\mathbf{x}_i^T \theta^* - y_i)^2 = \frac{1}{m} \sum_{i=n+1}^{n+m} \varepsilon_i^2$$

# e.g. A Practical Application of Regression Model

## Movie Reviews and Revenues: An Experiment in Text Regression*

**Mahesh Joshi Dipanjan Das Kevin Gimpel Noah A. Smith**
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213, USA
{maheshj,dipanjan,kgimpel,nasmith}@cs.cmu.edu

### Abstract

We consider the problem of predicting a movie's opening weekend revenue. Previous work on this problem has used metadata about a movie—e.g., its genre, MPAA rating, and cast—with very limited work making use of text *about* the movie. In this paper, we use the text of film critics' reviews from several sources to predict opening weekend revenue. We describe a new dataset pairing movie reviews with metadata and revenue data, and show that review text can substitute for metadata, and even improve over it, for prediction.

Proceedings of HLT '2010 Human Language Technologies:

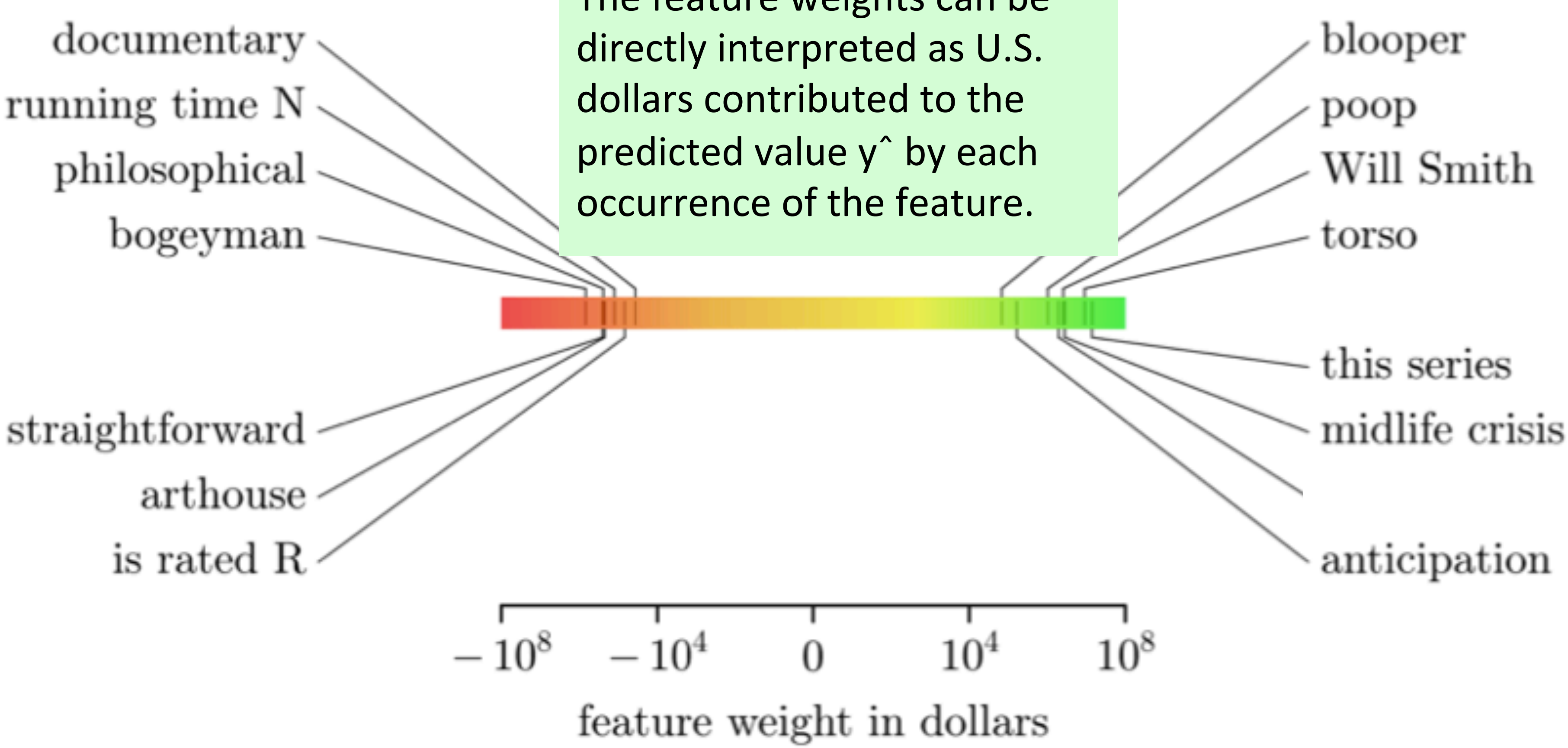
**A REAL APPLICATION:** Movie Reviews and Revenues: An Experiment in Text Regression, Proceedings of HLT '10 Human Language Technologies:

## VIII. Get the Data!

www.ark.cs.cmu.edu/movie$-data

## V. What May Have Brought You to movies

The feature weights can be directly interpreted as U.S. dollars contributed to the predicted value y^ by each occurrence of the feature.



| | Features | Site | Total MAE ($M) | Total $r$ | Per Screen MAE ($K) | Per Screen $r$ |
|---|---|---|---|---|---|---|
| | Predict mean | | 11.672 | – | 6.862 | – |
| | Predict median | | 10.521 | – | 6.642 | – |
| meta | Best | | 5.983 | 0.722 | 6.540 | 0.272 |
| text | I | – | 8.013 | 0.743 | 6.509 | 0.222 |
| text | I | + | 7.722 | 0.781 | 6.071 | 0.466 |
| text | *see Tab. 3* | B | 7.627 | 0.793 | 6.060 | 0.411 |
| text | I ∪ II | – | 8.060 | 0.743 | 6.542 | 0.233 |
| text | I ∪ II | + | **7.420** | 0.761 | 6.240 | 0.398 |
| text | I ∪ II | B | 7.447 | 0.778 | 6.299 | 0.363 |
| text | I ∪ III | – | 8.005 | 0.744 | 6.505 | 0.223 |
| text | I ∪ III | + | 7.721 | 0.785 | 6.013 | **0.473** |
| text | I ∪ III | B | 7.595 | **0.796** | †**6.010** | 0.421 |
| meta ∪ text | I | – | 5.921 | **0.819** | 6.509 | 0.222 |
| meta ∪ text | I | + | 5.757 | 0.810 | 6.063 | 0.470 |
| meta ∪ text | I | B | 5.750 | **0.819** | 6.052 | 0.414 |
| meta ∪ text | I ∪ II | – | 5.952 | 0.818 | 6.542 | 0.233 |
| meta ∪ text | I ∪ II | + | 5.752 | 0.800 | 6.230 | 0.400 |
| meta ∪ text | I ∪ II | B | 5.740 | **0.819** | 6.276 | 0.358 |
| meta ∪ text | I ∪ III | – | 5.921 | **0.819** | 6.505 | 0.223 |
| meta ∪ text | I ∪ III | + | **5.738** | 0.812 | 6.003 | **0.477** |
| meta ∪ text | I ∪ III | B | 5.750 | **0.819** | †**5.998** | 0.423 |

Table 2: Test-set performance for various models, measured using mean absolute error (MAE) and Pearson's correlation ($r$), for two prediction tasks.

I. $n$-grams. We considered unigrams, bigrams, and trigrams. A 25-word stoplist was used; bigrams and trigrams were only filtered if all words were stopwords.

II. Part-of-speech $n$-grams. As with words, we added unigrams, bigrams, and trigrams. Tags were obtained from the Stanford part-of-speech tagger (Toutanova and Manning, 2000).

III. Dependency relations. We used the Stanford parser (Klein and Manning, 2003) to parse the critic reviews and extract syntactic dependencies. The dependency relation features consist of just the relation part of a dependency triple ⟨relation, head word, modifier word⟩.

A combination of the meta and text features **achieves the best performance both in terms of MAE and pearson r.**

We consider three ways to combine the collection of reviews for a given movie. The first ("–") simply concatenates all of a movie's reviews into a single document before extracting features. The second ("+") conjoins each feature with the source site (e.g., *New York Times*) from whose review it was extracted. A third version (denoted "B") combines both the site-agnostic and site-specific features.

| | Feature | Weight ($M) |
|---|---|---|
| rating | pg | +0.085 |
| | *New York Times*: adult | -0.236 |
| | *New York Times*: rate_r | -0.364 |
| sequels | this_series | +13.925 |
| | *LA Times*: the_franchise | +5.112 |
| | *Variety*: the_sequel | +4.224 |
| people | *Boston Globe*: will_smith | +2.560 |
| | *Variety*: brittany | +1.128 |
| | ^_producer_brian | +0.486 |
| genre | *Variety*: testosterone | +1.945 |
| | *Ent. Weekly*: comedy_for | +1.143 |
| | *Variety*: a_horror | +0.595 |
| | documentary | -0.037 |
| | independent | -0.127 |
| sentiment | *Boston Globe*: best_parts_of | +1.462 |
| | *Boston Globe*: smart_enough | +1.449 |
| | *LA Times*: a_good_thing | +1.117 |
| | shame_$ | -0.098 |
| | bogeyman | -0.689 |
| plot | *Variety*: torso | +9.054 |
| | vehicle_in | +5.827 |
| | superhero_$ | +2.020 |

Table 3: Highly weighted features categorized manually. ^ and $ denote sentence boundaries.

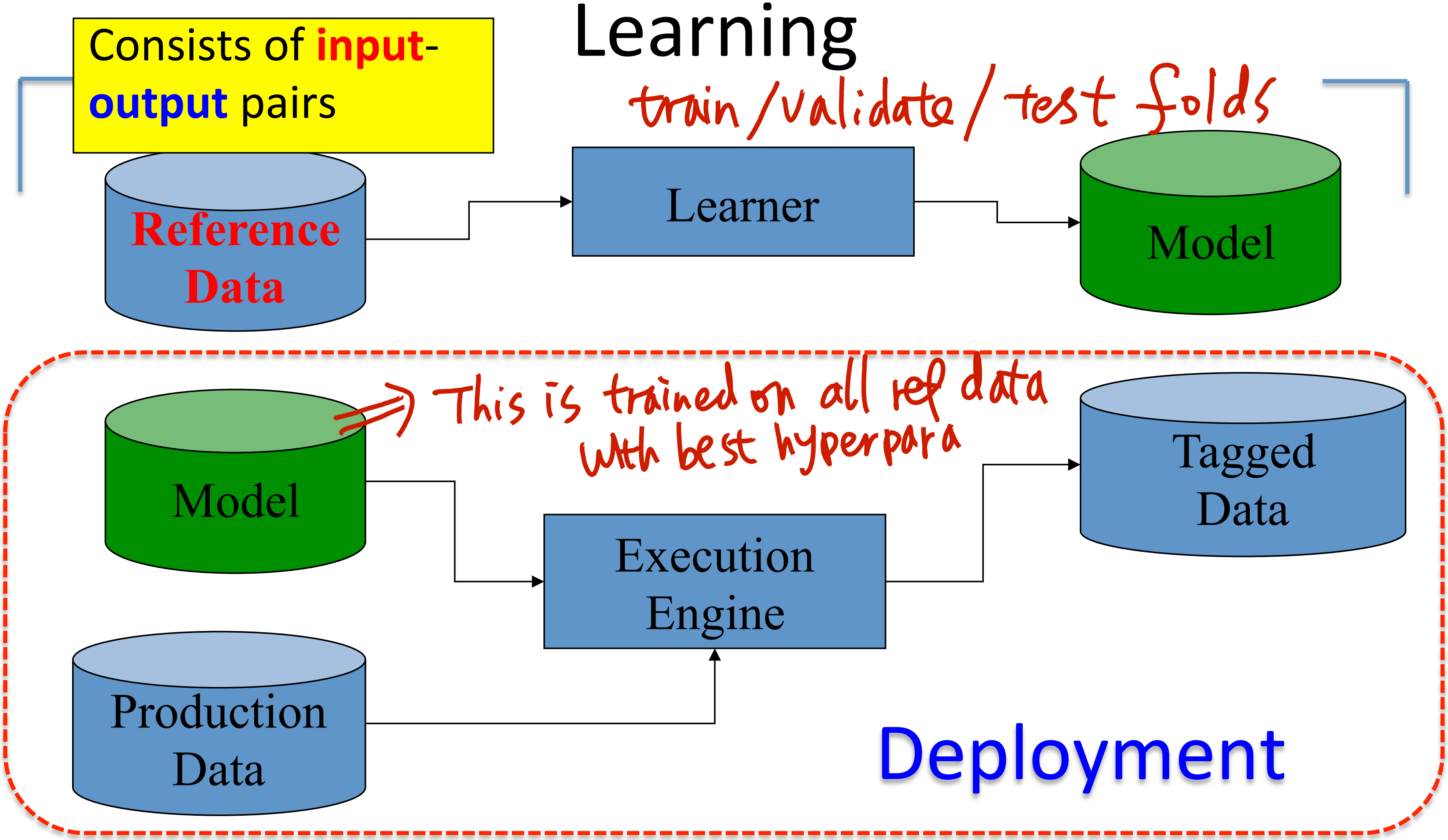
Movie Reviews and Revenues: An Experiment in Text Regression, Proceedings of HLT '10 Human Language Technologies:

The features are from the text-only model annotated in Table 2 (total, not per screen).

The feature weights can be directly interpreted as U.S. dollars contributed to the predicted value by each occurrence of the feature.

Sentiment-related text features are not as prominent as might be expected, and their overall proportion in the set of features with non-zero weights is quite small (estimated in preliminary trials at less than 15%). Phrases that refer to metadata are the more highly weighted and frequent ones.

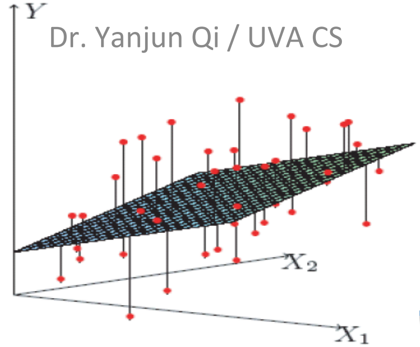
# An Operational Model of Machine Learning



# Goals in General

- 1. Generalize Well
  - Connecting to Asymptotic ERROR BOUND
- 2. Interpretable
  - Especially for some domains, this is about trust!
- 3. Computational Efficient

# Probabilistic Interpretation of Linear Regression (LATER)




- Let us assume that the target variable and the inputs are related by the equation:

$$y_i = \theta^T \mathbf{x}_i + \varepsilon_i$$

error data on each point

where $\varepsilon$ is an error term of unmodeled effects or random noise

- Now assume that $\varepsilon$ follows a Gaussian $N(0,\sigma)$, then we have:

$$p(y_i \mid x_i;\theta) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left( -\frac{(y_i - \theta^T \mathbf{x}_i)^2}{2\sigma^2} \right)$$

Many more variations of LinearR from this perspective, e.g. binomial / poisson (LATER)

- By iid (among samples) assumption:

$$L(\theta) = \prod_{i=1}^{n} p(y_i \mid x_i;\theta) = \left( \frac{1}{\sqrt{2\pi}\sigma} \right)^n \exp\left( -\frac{\sum_{i=1}^{n} (y_i - \theta^T \mathbf{x}_i)^2}{2\sigma^2} \right)$$

# References

- Big thanks to Prof. Eric Xing @ CMU for allowing me to reuse some of his slides
- Prof. Alexander Gray's slides