ORIE 4741: Learning with Big Messy Data

Feature Engineering

Professor Udell
Operations Research and Information Engineering
Cornell

August 29, 2019
Agenda

▶ announcements
  ▶ Vote today! + registration deadline October 12 2017 (for Nov 2017 election, or to change parties for 2018 election)
  ▶ Homework 1 due this morning
  ▶ Homework 2 out tonight, due in two weeks
  ▶ Always use Julia v0.6 (when you use Julia)
  ▶ Section this week: Github + projects
  ▶ No section next Tuesday
  ▶ Lecture videos: http://cornell.videonote.com/channels/1036/videos
  ▶ Midterm exam in class on October 5, covers through decision trees
  ▶ The role of confusion in learning

▶ feature engineering

▶ project discussion
To fit a linear model (≡ linear in parameters \( w \))

▶ pick a transformation \( \phi : \mathcal{X} \rightarrow \mathbb{R}^d \)
▶ predict \( y \) using a linear function of \( \phi(x) \)

\[
h(x) = w^T \phi(x) = \sum_{i=1}^{d} w_i \phi(x)_i
\]

▶ we want \( h(x_i) \approx y_i \) for every \( i = 1, \ldots, n \)
Linear models

To fit a linear model (= linear in parameters $w$)

- pick a transformation $\phi : \mathcal{X} \rightarrow \mathbb{R}^d$
- predict $y$ using a linear function of $\phi(x)$

$$h(x) = w^T \phi(x) = \sum_{i=1}^{d} w_i(\phi(x))_i$$

- we want $h(x_i) \approx y_i$ for every $i = 1, \ldots, n$

Q: why do we want a model linear in the parameters $w$?
Linear models

To fit a linear model (≡ linear in parameters $w$)

- pick a transformation $\phi : \mathcal{X} \rightarrow \mathbb{R}^d$
- predict $y$ using a linear function of $\phi(x)$

$$h(x) = w^T \phi(x) = \sum_{i=1}^{d} w_i \phi(x)_i$$

- we want $h(x_i) \approx y_i$ for every $i = 1, \ldots, n$

**Q:** why do we want a model linear in the parameters $w$?

**A:** because the optimization problems are easy to solve! e.g., just use least squares.
How to pick $\phi : \mathcal{X} \rightarrow \mathbb{R}^d$?

- so response $y$ will depend linearly on $\phi(x)$
- so $d$ is not too big
Feature engineering

How to pick $\phi : \mathcal{X} \to \mathbb{R}^d$?

- so response $y$ will depend linearly on $\phi(x)$
- so $d$ is not too big

if you think this looks like a hack: you’re right
Feature engineering

examples:

▶ adding offset
▶ standardizing features
▶ polynomial fits
▶ products of features
▶ autoregressive models
▶ local linear regression
▶ transforming images
▶ transforming Booleans
▶ transforming ordinals
▶ transforming nominals
▶ transforming sentences
▶ concatenating data
▶ all of the above

https://xkcd.com/2048/
Adding offset

- $\mathcal{X} = \mathbb{R}^{d-1}$
- let $\phi(x) = (x, 1)$
- now $h(x) = w^T \phi(x) = w_{1:d-1}^T x + w_d$
Fitting a polynomial

\[ X = \mathbb{R} \]

let

\[ \phi(x) = (1, x, x^2, x^3, \ldots, x^{d-1}) \]

be the vector of all monomials in \( x \) of degree \(<\ d \)

now \( h(x) = w^T \phi(x) = w_1 + w_2 x + w_3 x^2 + \cdots + w_d x^{d-1} \)
Demo: Linear models

https://github.com/ORIE4741/demos
Fitting a multivariate polynomial

- $\mathcal{X} = \mathbb{R}^2$
- pick a maximum degree $k$
- let
  
  $$\phi(x) = (1, x_1, x_2, x_1^2, x_1 x_2, x_2^2, x_1^3, x_1^2 x_2, x_1 x_2^2, x_2^3 \ldots, x_2^k)$$

  be the vector of all monomials in $x_1$ and $x_2$ of degree $< d$
- now $h(x) = w^T \phi(x)$ can fit any polynomial of degree $\leq k$ in $\mathcal{X}$
Fitting a multivariate polynomial

- \( \mathcal{X} = \mathbb{R}^2 \)
- pick a maximum degree \( k \)
- let

\[
\phi(x) = (1, x_1, x_2, x_1^2, x_1x_2, x_2^2, x_1^3, x_1^2x_2, x_1x_2^2, x_2^3 \ldots, x_2^k)
\]

be the vector of all monomials in \( x_1 \) and \( x_2 \) of degree \( < d \)

- now \( h(x) = w^T \phi(x) \) can fit any polynomial of degree \( \leq k \) in \( \mathcal{X} \)

and similarly for \( \mathcal{X} = \mathbb{R}^d \ldots \)
Example: fitting a multivariate polynomial

- $\mathcal{X} = \mathbb{R}^2$, $\mathcal{Y} = \{0, 1\}$
- let
  \[ \phi(x) = (1, x_1, x_2, x_1^2, x_1x_2, x_2^2) \]
  be the vector of all monomials of degree $\leq 2$
- now let $h(x) = \text{sign}(w^T \phi(x))$

**Q:** if $h(x) = \text{sign}(5 - 3x_1 + 2x_2 + x_1^2 - x_1x_2 + 2x_2^2)$, what is $\{x : h(x) = 1\}$?
Example: fitting a multivariate polynomial

- $\mathcal{X} = \mathbb{R}^2$, $\mathcal{Y} = \{0, 1\}$
- let
  \[
  \phi(x) = (1, x_1, x_2, x_1^2, x_1x_2, x_2^2)
  \]
  be the vector of all monomials of degree $\leq 2$
- now let $h(x) = \text{sign}(w^T \phi(x))$

**Q:** if $h(x) = \text{sign}(5 - 3x_1 + 2x_2 + x_1^2 - x_1x_2 + 2x_2^2)$, what is $\{x : h(x) = 1\}$?

**A:** An ellipse!
Example: fitting a multivariate polynomial

- $\mathcal{X} = \mathbb{R}^2$, $\mathcal{Y} = \{-1, 1\}$
- let
  \[ \phi(x) = (1, x_1, x_2, x_1^2, x_1x_2, x_2^2) \]
  be the vector of all monomials of degree $\leq 2$
- now let \( h(x) = \text{sign}(w^T \phi(x)) \)

Q: if \( h(x) = \text{sign}(5 - 3x_1 + 2x_2 + x_1^2 - x_1x_2 - 2x_2^2) \), what is
\[ \{x : h(x) = -1\} \]
Example: fitting a multivariate polynomial

- \( \mathcal{X} = \mathbb{R}^2, \mathcal{Y} = \{-1, 1\} \)
- let
  \[
  \phi(x) = (1, x_1, x_2, x_1^2, x_1 x_2, x_2^2)
  \]
  be the vector of all monomials of degree \( \leq 2 \)
- now let \( h(x) = \text{sign}(w^T \phi(x)) \)

**Q:** if \( h(x) = \text{sign}(5 - 3x_1 + 2x_2 + x_1^2 - x_1 x_2 - 2x_2^2) \), what is

\[
\{x : h(x) = -1\}?
\]

**A:** A hyperbola!
Fitting a time series

- given a time series $x_t \in \mathbb{R}$, $t = 1, \ldots, T$
- want to predict the value at the next time $T + 1$
- input: time index $t$ and time series $x_{1:t}$ up to time $t$
- let
  \[ \phi(t, x) = (x_{t-1}, x_{t-2}, \ldots, x_{t-d}) \]
  (called the “lagged outcomes”)
- now $h(x) = w^T \phi(x) = w_1 x_{t-1} + w_2 x_{t-2} + \cdots + w_d x^{t-d}$
  also called an **auto-regressive (AR) model**
New notation: boolean indicator function

define

\[ \mathbb{1}(\text{statement}) = \begin{cases} 1 & \text{statement is true} \\ 0 & \text{statement is false} \end{cases} \]

examples:

- \[ \mathbb{1}(1 < 0) = 0 \]
- \[ \mathbb{1}(17 = 17) = 1 \]
Fitting a local model: local neighbors

- \( \mathcal{X} = \mathbb{R}^m \)
- pick a set of points \( \mu_i \in \mathbb{R}^m, i = 1, \ldots, d \), radius \( \delta \in \mathbb{R} \)
- let \( \phi(x) = [\mathbb{1}(\|x - \mu_1\|^2 \leq \delta), \ldots, \mathbb{1}(\|x - \mu_d\|^2 \leq \delta)] \)

often, \( d = n \) and \( \mu_i = x_i \)
Fitting a local model: 1 nearest neighbor

- $\mathcal{X} = \mathbb{R}^m$
- pick a set of points $\mu_i \in \mathbb{R}^m$, $i = 1, \ldots, d$
- let $\delta = \min(\|x - \mu_1\|^2, \ldots, \|x - \mu_d\|^2)$
- let $\phi(x) = [\mathbb{1}(\|x - \mu_1\|^2 \leq \delta), \ldots, \mathbb{1}(\|x - \mu_d\|^2 \leq \delta)]$

often, $d = n$ and $\mu_i = x_i$
Fitting a local model: smoothing

- $\mathcal{X} = \mathbb{R}^m$
- pick a set of points $\mu_i \in \mathbb{R}^m$, $i = 1, \ldots, d$, parameter $\alpha \in \mathbb{R}$
- let

$$\phi(x) = (\exp(-\alpha\|x - \mu_1\|^2), \ldots, \exp(-\alpha\|x - \mu_d\|^2))$$

often, $d = n$ and $\mu_i = x_i$
Demo: Crime

preprocessing: https://juliabox.com/notebooks/demos/Crime.ipynb

predicting: https://juliabox.com/notebooks/demos/Predicting%20crime.ipynb
Boolean variables

\[ \mathcal{X} = \{\text{true, false}\} \]

\[ \text{let } \phi(x) = \mathbb{1}(x) \]
Boolean expressions

\[ \mathcal{X} = \{\text{true}, \text{false}\}^2 = \{\text{true}, \text{false}\}^2 = \{(\text{true, true}), (\text{true, false}), (\text{false, true}), (\text{false, false})\}^2. \]

- let \( \phi(x) = [ \mathbb{1}(x_1), \mathbb{1}(x_2), \mathbb{1}(x_1 \text{ and } x_2), \mathbb{1}(x_1 \text{ or } x_2)] \)
- equivalent: polynomials in \([\mathbb{1}(x_1), \mathbb{1}(x_2)]\) span the same space
- encodes logical expressions!
Nominal values

- $\mathcal{X} = \{\text{apple, orange, banana}\}$
- let $\phi(x) = [\mathbb{1}(x = \text{apple}), \mathbb{1}(x = \text{orange}), \mathbb{1}(x = \text{banana})]$
- called **one-hot encoding**: only one element is non-zero
Language

- $\mathcal{X} = \text{sentences, documents, tweets, \ldots}$
- let $\{w_1, \ldots, w_d\}$ be a set of words (or hashtags, or emoji, or \ldots)
- let $\phi(x) = [\mathbb{1}(x \text{ contains } w_1), \ldots, \mathbb{1}(x \text{ contains } w_d)]$
- called **bag-of-words model**: ignores order of words in sentence
Review

- linear models are linear in the parameters $w$
- can fit many different models by picking feature mapping $\phi : \mathcal{X} \rightarrow \mathbb{R}^d$
What makes a good project?

- Clear outcome to predict
- Linear regression should do something interesting
- New, interesting model; not a Kaggle competition
- Avoid: images, time series, NLP