Recitation 7

Feature Extraction

Artie Shen, David Rosenberg, Ben Jakubowski

CDS

March 9, 2020

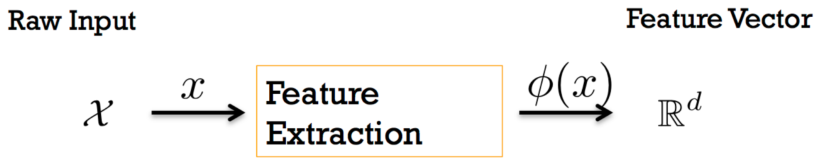
Agenda

- What is feature extraction?
- Why extract features?
- How to systematically extract non-linear features?
- Implementing feature extraction using Sklearn.
- Dealing with natural languages.

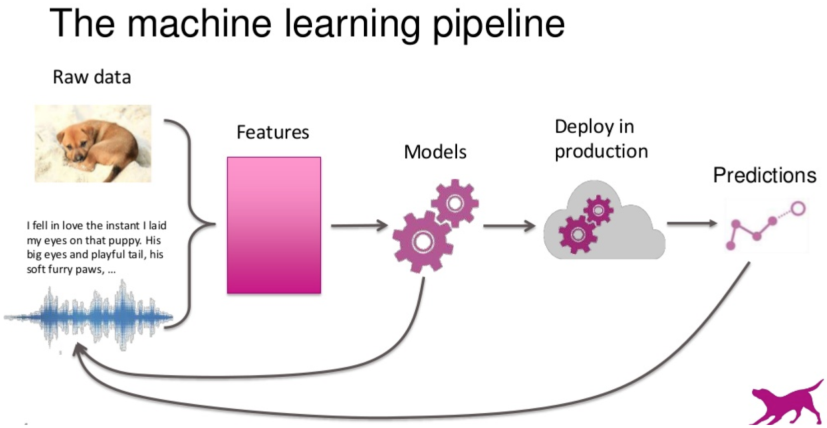
Feature Extraction

Definition

The mapping $\varphi(\cdot)$ that maps an input $x \in \mathcal{X}$ to a vector $\varphi(x) \in \mathbb{R}^d$ is called **feature extraction** or **featurization**.



The Big Picture


The machine learning pipeline
Raw data
I fell in love the instant I laid my eyes on that puppy. His big eyes and playful tail, his soft furry paws, ...
Features
Models
Deploy in production
Predictions

https://www.slideshare.net/AliceZheng3/understanding-feature-space-in-machine-learning

Motivation

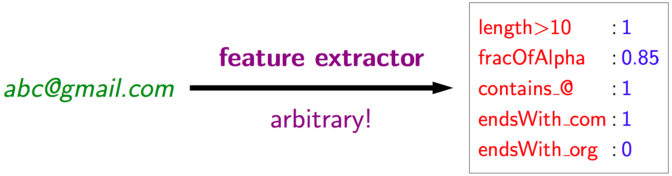
- Our general learning theory setup: no assumptions about $\mathcal{X}$.
- But $\mathcal{X} = \mathbb{R}^d$ for the specific methods we've developed:
    - Ridge regression
    - Lasso regression
    - Linear SVM
- What if inputs x are not naively in $\mathbb{R}^d$?
    - Text documents
    - Image files
    - Sound recordings
    - DNA sequences

Case 1: Detecting Email Addresses

Task

Given a string $s \in \mathcal{S}$, predict $y \in \{0, 1\}$ that indicates if s is an email address.

- What features could be useful?
- Can we systematically extract these features for all input $s \in \mathcal{S}$?



From Percy Liang's "Lecture 3" slides from Stanford's CS221, Autumn 2014.

Feature Templates

Definition (informal)

A **feature template** is a group of features all computed in a similar way.

- Input: *abc@gmail.com*
- Feature Templates:
  - Length greater than ___
  - Last three characters equal ___
  - Contains character ___

Based on Percy Liang's "Lecture 3" slides from Stanford's CS221, Autumn 2014.

Feature Template: Last Three Characters Equal ___

- Don't think about which 3-letter suffixes are meaningful...
- Just **include them all.**

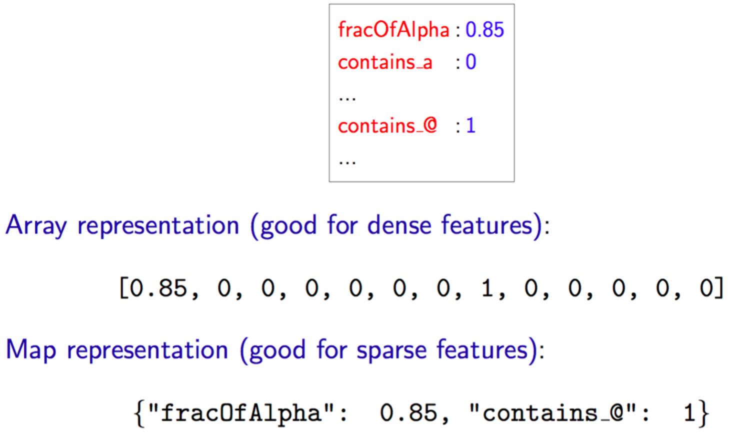
abc@gmail.com $\longrightarrow$

endsWith_aaa	: 0
endsWith_aab	: 0
endsWith_aac	: 0
...	
endsWith_com	: 1
...	
endsWith_zzz	: 0

- With regularization, our methods will not be overwhelmed.

From Percy Liang's "Lecture 3" slides from Stanford's CS221, Autumn 2014.

Feature Vector Representations



fracOfAlpha	: 0.85
contains_a	: 0
...	
contains_@	: 1
...	

Array representation (good for dense features):

```
[0.85, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0]
```

Map representation (good for sparse features):

```
{"fracOfAlpha": 0.85, "contains_@": 1}
```

From Percy Liang's "Lecture 3" slides from Stanford's CS221, Autumn 2014.

Feature Vector Representations

- Arrays
  - assumed fixed ordering of the features
  - appropriate when significant number of nonzero elements ("**dense feature vectors**")
  - very efficient in space and speed
- Map
  - best for **sparse feature vectors** (i.e. few nonzero features)
  - features not in the map have default value of zero
  - Has overhead compared to arrays, so much slower for dense features.

From Percy Liang's "Lecture 3" slides from Stanford's CS221, Autumn 2014.

Case 2: Document Classification

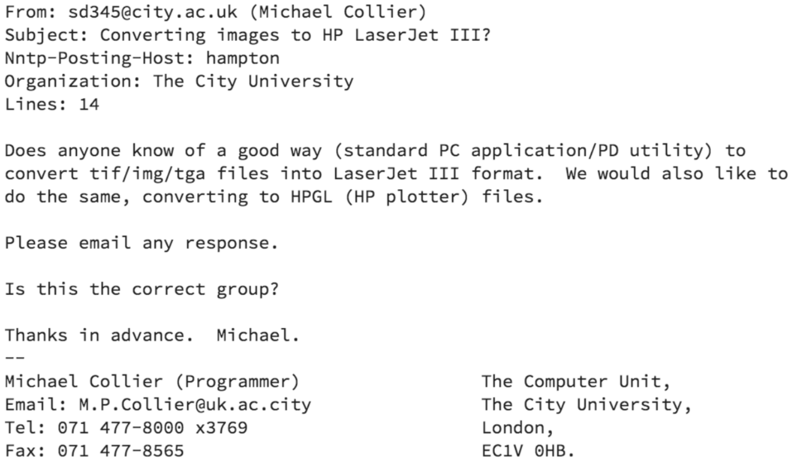
Task

Given a document x, predict the topic $y \in \{0, 1, 2, 3\}$ associated with x.

- The 20 Newsgroups Data Set comprises around 18000 newsgroups posts on 20 topics.
- Each post is associated with a single topic out of 20 pre-defined topics.
- We'll restrict ourselves to classifying posts within 4 topics:
  - 'alt.atheism'
  - 'soc.religion.christian'
  - 'comp.graphics'
  - 'sci.med'.

Example: Classifying documents from 20 newsgroups

Example Document:



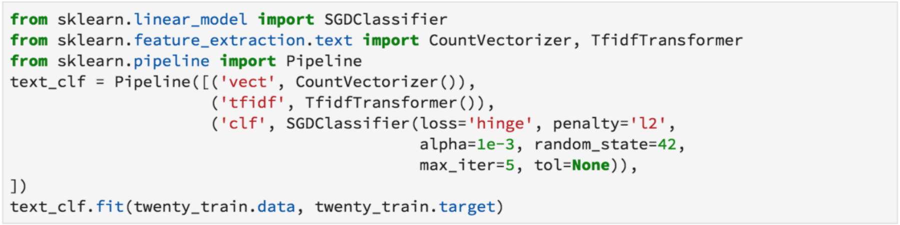
```
From: sd345@city.ac.uk (Michael Collier)
Subject: Converting images to HP LaserJet III?
Nntp-Posting-Host: hampton
Organization: The City University
Lines: 14

Does anyone know of a good way (standard PC application/PD utility) to
convert tif/img/tga files into LaserJet III format.  We would also like to
do the same, converting to HPGL (HP plotter) files.

Please email any response.

Is this the correct group?

Thanks in advance.  Michael.
--
Michael Collier (Programmer)                 The Computer Unit,
Email: M.P.Collier@uk.ac.city                The City University,
Tel: 071 477-8000 x3769                      London,
Fax: 071 477-8565                            EC1V 0HB.
```

Bag of Words

- In this case, it's hard to design an exhausted set of features.
- Instead, we can let the data decide what feature to use.
- Bag of Words (BOW):
  - Construct a vocabulary base $\mathcal{W}$, which could be the most frequent k words / bi-grams.
  - For each document x, count the number of occurrences of each word $w \in \mathcal{W}$.
  - The resulting feature map $\varphi(x) = [N_{w_1}, \cdots, N_{w_k}] \in \mathbb{R}^k$.
- With BOW, we transform each document x into a feature vector $\varphi(x) \in \mathbb{R}^k$. We can then fit a linear model on the feature vectors.

Sklearn Implementation

- Sklearn has great support for ML pipelines that connects feature extractions and modeling.



```
from sklearn.linear_model import SGDClassifier
from sklearn.feature_extraction.text import CountVectorizer, TfidfTransformer
from sklearn.pipeline import Pipeline
text_clf = Pipeline([('vect', CountVectorizer()),
                     ('tfidf', TfidfTransformer()),
                     ('clf', SGDClassifier(loss='hinge', penalty='l2',
                                           alpha=1e-3, random_state=42,
                                           max_iter=5, tol=None)),
])
text_clf.fit(twenty_train.data, twenty_train.target)
```

Natural Language Processing

- What would be an issue about BOW?
- "The rabbit sees the fox." v.s. "The fox sees the rabbit."
- Word embedding
    - GloVe
    - Word2Vec
- Sentence / document embedding:
    - SentEval
    - BERT

Motivation

- Suppose that our input space is $\mathbb{R}^d$, do we still need features?
- What if the input data and labels are correlated in a non-linear way?
- How to categorize non-linearity?
- Three types of non-linearity:
    - Non-monotonicity
    - Saturation
    - Interactions between features

Non-monotonicity: The Issue

- Scenario: suppose we would like to use a patient's body temperature x to predict this patient's health condition.
- Input: $x \in \mathbb{R}$
- Action: predict health score $y \in \mathbb{R}$ (positive is good).
- Hypothesis Space $\mathcal{F} = \{\text{affine functions of temperature}\}$
- Issue: health is not an affine function of temperature.
- Affine function can either say
  - Very high is bad and very low is good, or
  - Very low is bad and very high is good,
  - But here, both extremes are bad.

From Percy Liang's "Lecture 3" slides from Stanford's CS221, Autumn 2014.

Non-monotonicity: Solution 1

- Transform the input:

$$\varphi(x) = \left[1, \{\text{x-37}\}^2\right],$$

where 37 is “normal” temperature in Celsius.
- This requires domain knowledge.
- Do we really need that?

From Percy Liang's "Lecture 3" slides from Stanford's CS221, Autumn 2014.

Non-monotonicity: Solution 2

- Think less, put in more:

$$\varphi(x) = \left[1, x, x^2\right].$$

- **More expressive** than Solution 1.

General Rule

Features should be simple building blocks that can be pieced together.

From Percy Liang's "Lecture 3" slides from Stanford's CS221, Autumn 2014.

Saturation: The Issue

- Scenario: given the number of people who has bought the product $x \in \mathbb{R}$, predict how likely this product will win the "best seller" award.
- Input: buy count of a product x.
- Action: predict the probability $y \in [0, 1]$ of this product winning the award.
- We expect a monotonic relationship between x and relevance, but is relevance linear in x?
- Relevance score reflects confidence in relevance prediction.
- Are we 10 times more confident if $x = 1000$ vs $x = 100$?
- Bigger is better... but not that much better.

From Percy Liang's "Lecture 3" slides from Stanford's CS221, Autumn 2014.

Saturation: Solve with nonlinear transform

- Smooth nonlinear transformation:

$$\varphi(x) = [1, \log\{1 + N(x)\}]$$

- $\log(\cdot)$ good for values with large dynamic ranges
- *Does it matter what base we use in the log?*
- Discretization (a discontinuous transformation):

$$\varphi(x) = (\mathbf{1}5 \le N(x) < 10, \mathbf{1}10 \le N(x) < 100, \mathbf{1}100 \le N(x))$$

- Small buckets allow quite flexible relationship.

From Percy Liang's "Lecture 3" slides from Stanford's CS221, Autumn 2014.

Interactions: The Issue

- Input: Patient information x
- Action: Health score $y \in \mathbb{R}$ (higher is better)
- Feature Map

$$\varphi(x) = [\text{height}(x), \text{weight}(x)]$$

- Issue: It's the weight **relative** to the height that's important.
- Impossible to get with these features and a linear classifier.
- Need some **interaction** between height and weight.

From Percy Liang's "Lecture 3" slides from Stanford's CS221, Autumn 2014.

Interactions: Approach 1

- Google "ideal weight from height"
- J. D. Robinson's "ideal weight" formula (for a male):

$$\text{weight(kg)} = 52 + 1.9\,[\text{height(in)} - 60]$$

- Make score square deviation between height(h) and ideal weight(w)

$$f(x) = (52 + 1.9\,[h(x) - 60] - w(x))^2$$

- WolframAlpha for complicated Mathematics:

$$f(x) = 3.61h(x)^2 - 3.8h(x)w(x) - 235.6h(x) + w(x)^2 + 124w(x) + 3844$$

From Percy Liang's "Lecture 3" slides from Stanford's CS221, Autumn 2014.

Interactions: Approach 2

- Just include all second order features:

$$\varphi(x) = \left[1, h(x), w(x), h(x)^2, w(x)^2, \underbrace{h(x)w(x)}_{\text{cross term}}\right]$$

- More flexible, no Google, no WolframAlpha.

General Principle

Simpler building blocks replace a single "smart" feature.

From Percy Liang's "Lecture 3" slides from Stanford's CS221, Autumn 2014.

Predicate Features and Interaction Terms

Definition

A **predicate** on the input space $\mathcal{X}$ is a function $P : \mathcal{X} \to \{\text{True}, \text{False}\}$.

- Many features take this form:
  - $x \mapsto s(x) = \mathbf{1}$subject is sleeping
  - $x \mapsto d(x) = \mathbf{1}$subject is driving
- For predicates, interaction terms correspond to **AND** conjunctions:
  - $x \mapsto s(x)d(x) = \mathbf{1}$subject is sleeping AND subject is driving

So What's Linear?

- Non-linear feature map $\varphi : \mathcal{X} \to \mathbb{R}^d$
- Hypothesis space:

$$\mathcal{F} = \left\{ f(x) = w^T \varphi(x) \mid w \in \mathbb{R}^d \right\}.$$

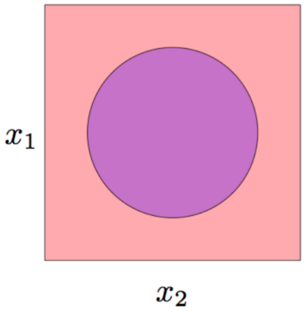
- Linear in w? Yes.
- Linear in $\varphi(x)$? Yes.
- Linear in x? No.
  - Linearity not even defined unless $\mathcal{X}$ is a vector space

Key Idea: Non-Linearity

- Nonlinear $f(x)$ is important for **expressivity**.
- $f(x)$ linear in w and $\varphi(x)$: makes finding $f^*(x)$ much easier

From Percy Liang's "Lecture 3" slides from Stanford's CS221, Autumn 2014.

Geometric Example: Two class problem, nonlinear boundary



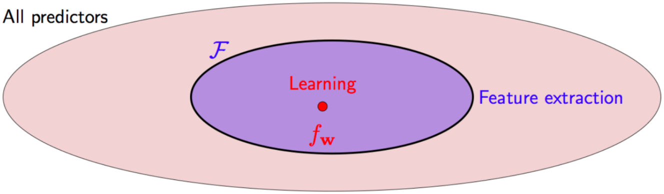
- With linear feature map $\varphi(x) = (x_1, x_2)$ and linear models, no hope
- With appropriate nonlinearity $\varphi(x) = \left(x_1, x_2, x_1^2 + x_2^2\right)$, piece of cake.

From Percy Liang's "Lecture 3" slides from Stanford's CS221, Autumn 2014.

Expressivity of Hypothesis Space

Consider a linear hypothesis space with a feature map $\varphi : \mathcal{X} \to \mathbb{R}^d$:

$$\mathcal{F} = \left\{ f(x) = w^T \varphi(x) \right\}$$



We can grow the linear hypothesis space $\mathcal{F}$ by adding more features.

From Percy Liang's "Lecture 3" slides from Stanford's CS221, Autumn 2014.

Case 3: Boston Housing

Task

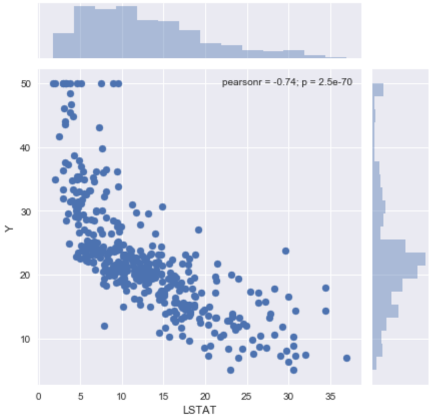
Given information of a community x, predict the median price of properties y in this community.

We have 13 features that include:

- CRIM: per capita crime rate by town
- DIS: weighted distances to five Boston employment centres
- AGE: proportion of owner-occupied units built prior to 1940
- LSTA: % lower status of the population
- RM: average number of rooms per dwelling
- TAX: full-value property-tax rate per $10,000
- ...

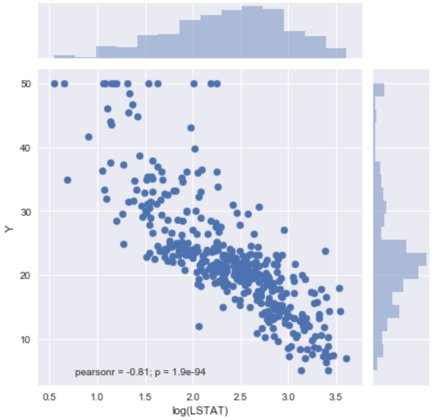
Boston Housing

Let's plot LSTA v.s. y. Is this relation linear? Can we make the trend "more linear"?



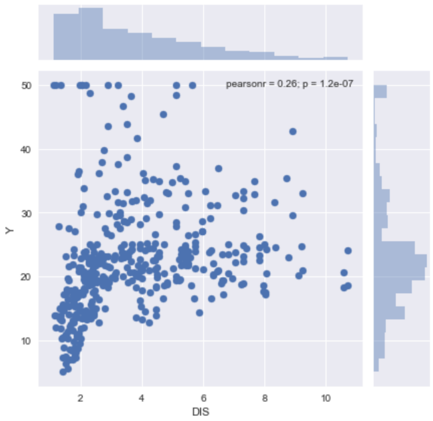
Boston Housing

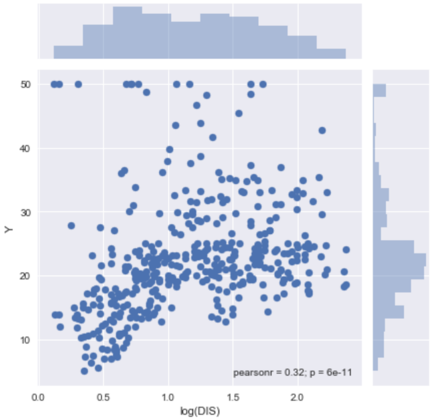
Yes, we can use log operator.



Boston Housing

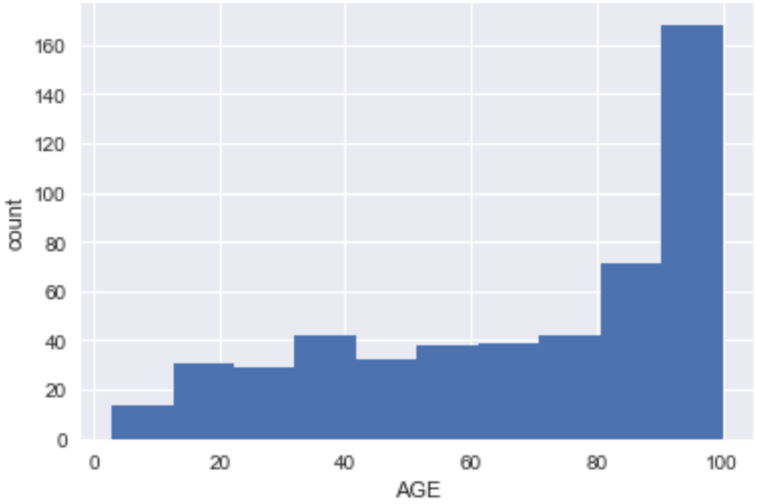
Similar technique can be applied to DIS feature.





Boston Housing

What can we do about the age feature?



Learning Objectives

- Understand where a *feature map* sits in a machine learning pipeline.
- Understand that featurization/featuring mapping is inherently required to allow predictors to ingest many types of data.
- Understand how feature extraction can be used to extend the power of linear methods.
- Build pipelines with expanded feature spaces using the sklearn ecosystem.