Introduction to Data Analytics

A/Prof Richard Yi Da Xu
Yida.Xu@uts.edu.au
Wechat: aubedata
https://github.com/roboticcam/machine-learning-notes

University of Technology Sydney (UTS)

February 18, 2018

Machine Learning applications: The three-layers perspective

This is my *biased* view:

- **Layer 1: Application - Business analyst**
  - Define the problem. Obtain its business value and find out what to do.
  - Domain specific knowledge is essential
  - Each project is different, there are no identical projects!
  - knowledge on general overview of machine learning
- **Layer 2: Model Formulation - Data Scientist**
  - transform the business problem into a mathematical framework:
  - knowledge on how to apply machine learning
- **Layer 3: Solver - Machine learning practitioner/researcher**
  - Now we have the model, how we can solve these equations
  - need to consider program complexity when data is BIG
  - research knowledge on machine learning and mathematics

- **Layer 1: Application**
  From OPAL data, *the business* wants an estimate on the probability of passenger taps on at central station at various times (*e.g., what is the probability of someone taps on at central at 8:15am?)*
- **Layer 2: Model Formulation**
  Model *all* passenger tap on times using Bi-modal Gaussian Mixture Model (GMM)
- **Layer 3: Solver**
  Solve GMM using **expectation-maximization**;

see `one_d_opal_simulated.m`

The three layers perspective example (2)

- **Layer 1: Application**
  an online hotel *business* has a database containing every user's rating of their stayed hotels
  the *marketing team* wants to know which hotels to recommend to individual user in a promotional email (customized emails)
- **Layer 2: Model Formulation**
  data scientists decide to build a **recommendation system** using Non-Negative Matrix Factorization (NNMF) algorithm
- **Layer 3: Solver**
  there are many ways to solve NNMF, but the team decide to use Gradient Descend, because of the relative small size database.

- **Layer 1: Application**
  - hypothetically, UTS decides to build the world's best learning analytic system:
  - it takes into consideration of the student's histories of studies and their interest, then it produces a "future study plan" deems to be best fit for each and every student
  - *and of course, they asked Richard's team to conduct this work*
- **Layer 2: Model Formulation**
  Richard's team decides to base this model using a modified Recurrent Neural Network (RNN).
- **Layer 3: Solver**
  learn all the parameters of RNN using standard back-propagation.

Demos:

- Connected Ellipse fitting
- Automated PTZ Camera control
- Markov Random Field via Swendsen-Wang sampling
- ...

Exercise:

- In each of these settings, what are the three layers: **application**, **model**, **solver**

Real problems: (larger) projects

- Education to Employment alignment
- Data Hackerthorn
- these are the systems where machine learning plays a part.

Exercise:

- In each of these settings, what are the three layers: **application**, **model**, **solver**

For the rest of the course, we will:

- discuss a mixture of **application**, **model** and **Solver**
- very gentle introduction to some of the mathematics. For detailed coverage of topics, refer to my Machine Learning course:
  `http://www-staff.it.uts.edu.au/~ydxu/statistics.htm`
- stop me at any time if anything unclear. I will go over it again and I may even tell a **big-data joke**

Three most common Learning algorithm

- Classification (supervised)
- Regression (supervised)
- Clustering (unsupervised)

Supervised Learning: **Regression** or **Classification**:

A generic example:

	X			y	y
	attribute 1	**attribute 1**	**attribute 3**	**class label** y	**dependent variable** y
data 1	50	64	1.2	C1	1.5
data 2	23	23	15	C2	0.2
data 3	50	80	3.2	C1	1.0
...	...	...	...	...	...
data N	5	90	25	C3	1.3
new data	60	43	12	?	?

- In here, each data example $X_i = (\text{attribute1}, \text{attribute2}, \text{attribute3})$
- Two type of labels:
  - y_i = category indicator (classification), or,
  - y_i = a real number (regression)

Regression or **Classification** example: Building design

	building area	**window U-value**	**wall U-value**	**is energy efficient**	**energy consumption**
Building 1	128	2.8	1.2	Yes	1500
Building 2	23	1.5	15	No	240
Building 3	45	3.4	3.2	No	1000
...	...	...	...	...	...
Building N	65	4.5	3.2	Yes	1301
Building N+1	160	3.2	3.2	?	?

- In here, each data example $X_i =$ (building area, window U-value, wall U-value)
- Two type of labels:
  - $y_i =$ is energy efficient (classification), or,
  - $y_i =$ energy consumption (regression)

Regression or **Classification** example: Hypothetical UTS student analytic

	math mark	**program mark**	**research mark**	**will study honors?**	**salary level?**
student 1	98	74	76	Yes	50K
student 2	67	100	50	No	75K
student 3	60	89	80	No	102K
...	...	...	...	...	...
student N	65	54	98	Yes	60K
student N+1	78	79	68	?	?

- In here, each data example $X_i =$ (math mark, program mark, research mark)
- Two type of labels:
  - $y_i =$ will study honors (classification), or,
  - $y_i =$ salary level? (regression)

Many models can be applied:

- In terms of **regression**: Linear, Polynomial, Ridge, Lasso Regression, ElasticNet, Gaussian Process , Decision Tree . . . many more
- In terms of **classification**: Neural networks, Support Vector Machine, Multinomial Logistic Regression (Softmax), Decision Tree, Random Forest . . . again, many more

Learning Category Exercise

Determine from the following, which are **supervised** and which is **unsupervised** learning? If they are supervised learning, which is **classification** and which is a **regression** task?

- Looking at Microsoft `https://how-old.net/`, it was trained using images of people with known ages.
- Obtain the segments of passenger traveling behaviors from Transport Survey Data.
- Given a video sequence, separate any arbitrary (moving) foreground from its background, assuming background pixels varies less than the foreground
- Having a historical relationship between material consumption and labor hours, develop model to predict the labor hours for the material consumption in the future.
- develop world's best object recognition system
- Looking at Experian Mozaic `http://www.experian.com.au`, which train and categorize customers (with some data feature extraction) into a set of predefined categories.

Questions What is the difference between these two categorical variables?

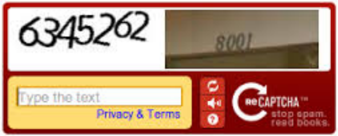
- **user ratings** of online products 1, 2, 3, 4, 5
- **email domain**: 1 = `@gmail.com`, 2 = `@mit.edu`, 3 = `@hotmail.com`, 4 = `@uts.edu.au`, 5 = `@amazon.com`,

Approaches to classification

- Generative approach
- Discriminative approach

Anything in-between?

- Now we know which are supervised learning VS unsupervised learning, is there anything in between?
- Human labeling is laborious and sometimes unpractical, for example, in the previous example, there are simply too many buildings, it takes too much time to **label** if the building is energy efficient.
- Crowd-sourcing sometimes helps:



- **Question** what if majority of people input random answer to it? Will it still work?
- without crowd source, is there a way we can learn without giving a label y_i to each of the x_i?
- The answer is **semi-supervised learning**

Semi-supervised learning

The problem definition (more formally)

- Given set of l independently identically distributed examples $x_1, \ldots, x_l \in X$ with corresponding labels $y_1, \ldots, y_l \in Y$
- Additionally, we are given u unlabeled examples $x_{l+1}, \ldots, x_{l+u} \in X$
- Semi-supervised learning attempts to make use of this combined information to surpass:
  - classification performance that could be obtained either by discarding the unlabeled data and
  - doing supervised learning or by discarding the labels and doing unsupervised learning.

The problem definition (less formally)

- Human labels a partial data
- Computer performs classification on the partially labelled data
- Computer performs classification on the unlabelled data, using assumptions and clever mathematics

Semi-supervised learning: an example approach

- We are given a set of independently identically distributed examples $x_1, \ldots, x_l \in X$ with corresponding labels $y_1, \ldots, y_l \in Y$.
- Additionally, we are given u unlabeled examples $x_{l+1}, \ldots, x_{l+u} \in X$.

Assumptions:

- Data points which are close to each other are more likely to share a label.
- Data tend to form discrete clusters, and points in the same cluster are more likely to share a label

  Parameter is then chosen based on fitting to both the labeled and unlabeled data, weighted by λ:

$$\underset{\Theta}{\operatorname{argmax}} \left(\underbrace{\log p(\{x_i, y_i\}_{i=1}^{l} | \theta)}_{\text{supervised}} + \lambda \underbrace{\log p(\{x_i\}_{i=l+1}^{l+u} | \theta)}_{\text{unsupervised}} \right)$$

- Data lie approximately on a manifold of much lower dimension than the input space.

OK, you have built your classification algorithm (or classifier). Then, how good is it?

- **classification accuracy**: Percentage of records correctly identified
- Most used: simple
- What happens when the dataset is skewed, or class imbalance?
- for example, if we were to predict the **fire-alarm true positives**, where 99% of fire-alarm is false positive:

Classification Accuracy: accuracy under class imbalance

- **TP (true positive)**: classifier shows it is *positive*, it is really *positive*
- **FP (false positive)**: classifier shows it is *positive*, it is really *negative*
- **TN (true negative)**: classifier shows it is *negative*, it is really *negative*
- **FN (false negative)**: classifier shows it is *negative*, it is really *positive*

which one is worse?

- **Precision** or **positive predictive value**: $\frac{\text{TP}}{\text{TP}+\text{FP}}$
- **Recall** or **sensitivity**: $\frac{\text{TP}}{\text{TP}+\text{FN}}$
- **specificity**: $\frac{\text{TN}}{\text{TN}+\text{FP}}$
- $\mathbf{F}_1$: $2 \times \frac{\text{precision} \times \text{recall}}{\text{precision}+\text{recall}}$

Classification Accuracy

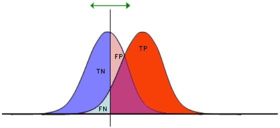
- Let X be the "score" generated from a binary classifier; (e.g., probability of being classified as 1 in logistic regression).
- Threshold T does **not** need to be 0.5, assume it's a variable
- $X \sim f_1(x)$ if the instance actually belongs to class "positive"
- $X \sim f_0(x)$ otherwise.
- True positive rate is given by:

$$\mathrm{TP}_R(T) = \int_T^{\infty} f_1(x)\, dx$$

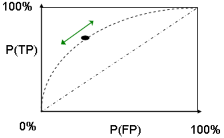
- False positive rate is given by:

$$\mathrm{FP}_R(T) = \int_T^{\infty} f_0(x)\, dx.$$

- The ROC curve plots parametrically $TP_R(T)$ versus $FP_R(T)$ with T as the varying parameter.



TP	FP
FN	TN
1	1



- purple part is "counted" both by TP and FP
- both TP and FP area stay on righthand side of the plot.
- **Question** which threshold value T leads to the point at **bottom left** of the figure?
- **Question** which threshold value T leads to the point at **top right** of the figure?
- **Question** what is an ideal plot?

In machine learning, there are **three** area of mathematics

- Linear Algebra
- Calculus
- Probablity and Statistics

Data “Structures”

Data “Structures” can be defined over:

- Scalar
- Vector
- Matrix
- Tensor

- In high school, arithmetic, probabilities and calculus are often defined just on scalar,
- but in machine learning, they usually defined over **vector** space, and sometimes can be defined over **matrix** and **tensor** .

- The idea of a function $f(x)$
- First and second derivatives
- Finding maximum and minimum of a function
- Important thing is that MOST problems we have in machine learning, we can't find an analytical solution, i.e., you can't solve $f'(x) = 0$ analytically.

Multivariate Calculus:

- x is usually defined over vector space
- positive definiteness
- Jacobian and Hessian Matrix etc