Machine Learning - MT 2016
16. Course Summary

Varun Kanade

University of Oxford

November 30, 2016

Machine Learning - What we covered

SVM

Naïve Bayes Convnets

k-Means Clustering

Kernels Logistic Regression Deep Learning

Least Squares PCA Discriminant Analysis Ridge Lasso

1800 2016

Gauss

Legendre

Hinton

Machine Learning Models and Methods

k-Nearest Neighbours
Linear Regression
Logistic Regression
Ridge Regression
Hidden Markov Models
Mixtures of Gaussian
Principle Component Analysis
Independent Component Analysis
Kernel Methods
Decision Trees
Boosting and Bagging
Belief Propagation
Variational Inference
EM Algorithm
Monte Carlo Methods
Spectral Clustering
Hierarchical Clustering
Recurrent Neural Networks

Linear Discriminant Analysis
Quadratic Discriminant Analysis
The Perceptron Algorithm
Naïve Bayes Classifier
Hierarchical Bayes
k-means Clustering
Support Vector Machines
Gaussian Processes
Deep Neural Networks
Convolutional Neural Networks
Markov Random Fields
Structural SVMs
Conditional Random Fields
Structure Learning
Restricted Boltzmann Machines
Multi-dimensional Scaling
Reinforcement Learning
$\cdots$

Machine Learning Models and Methods

k-Nearest Neighbours

Linear Regression

Logistic Regression

Ridge Regression

Hidden Markov Models

Mixtures of Gaussian

Principle Component Analysis

Independent Component Analysis

Kernel Methods

Decision Trees

Boosting and Bagging

Belief Propagation

Variational Inference

EM Algorithm

Monte Carlo Methods

Spectral Clustering

Hierarchical Clustering

Recurrent Neural Networks

Linear Discriminant Analysis

Quadratic Discriminant Analysis

The Perceptron Algorithm

Naïve Bayes Classifier

Hierarchical Bayes

k-means Clustering

Support Vector Machines

Gaussian Processes

Deep Neural Networks

Convolutional Neural Networks

Markov Random Fields

Structural SVMs

Conditional Random Fields

Structure Learning

Restricted Boltzmann Machines

Multi-dimensional Scaling

Reinforcement Learning

$\cdots$

Machine Learning Models and Methods

k-Nearest Neighbours

Linear Regression

Logistic Regression

Ridge Regression

Hidden Markov Models

~~Mixtures of Gaussian~~

Principle Component Analysis

Independent Component Analysis

Kernel Methods

~~Decision Trees~~

Boosting and Bagging

Belief Propagation

Variational Inference

~~EM Algorithm~~

Monte Carlo Methods

Spectral Clustering

Hierarchical Clustering

Recurrent Neural Networks

Linear Discriminant Analysis

Quadratic Discriminant Analysis

The Perceptron Algorithm

Naïve Bayes Classifier

Hierarchical Bayes

k-means Clustering

Support Vector Machines

Gaussian Processes

Deep Neural Networks

Convolutional Neural Networks

Markov Random Fields

Structural SVMs

Conditional Random Fields

Structure Learning

Restricted Boltzmann Machines

Multi-dimensional Scaling

Reinforcement Learning

$\cdots$

Learning Outcomes

On completion of the course students should be able to

- Describe and distinguish between various different paradigms of machine learning, particularly supervised and unsupervised learning
- Distinguish between task, model and algorithm and explain advantages and shortcomings of machine learning approaches
- Explain the underlying mathematical principles behind machine learning algorithms and paradigms
- Design and implement machine learning algorithms in a wide range of real-world applications (not to scale)

Model and Loss Function Choice

‘‘Optimisation’’ View of Machine Learning

- Pick model that you expect may fit the data well enough
- Pick a measure of performance that makes ‘‘sense’’ and can be optimised
- Run optimisation algorithm to obtain model parameters
- Supervised models such as Linear Regression (Least Squares), SVM, Neural Networks, *etc.*
- Unsupervised models PCA, k-means clustering, *etc.*

Model and Loss Function Choice

Probabilistic View of Machine Learning

- Pick a model for data and explicitly formulate the deviation (or uncertainty) from the model using the language of probability
- Use notions from probability to define suitability of various models
- Frequentist Statistics: Maximum Likelihood Estimation
- Bayesian Statistics: Maximum-a-posteriori, Full Bayesian (Not Examinable)
- Discriminative Supervised Models: Linear Regression (Gaussian, Laplace, and other noise models), Logistic Regression, *etc.*
- Generative Supervised Models: Naïve Bayes Classification, Gaussian Discriminant Analysis (LDA/QDA)
- (Not Covered) Probabilistic Generative Models for Unsupervised Learning

Optimisation Methods

After defining the model, except in the simplest of cases where we may get a closed form solution, we used optimisation methods

Gradient Based Methods: GD, SGD, Minibatch-GD, Newton's Method

Many, many extensions exist: Adagrad, Momentum, BGFS, L-BGFS, Adam

Convex Optimization

- Convex Optimization is 'efficient' (*i.e.,* polynomial time)
- Linear Programs, Quadratic Programs, General Convex Programs
- Gradient-based methods converge to global optimum

Non-Convex Optimization

- Encountered frequently in deep learning (but also other areas of ML)
- Gradient-based methods give local minimum
- Initialisation, Gradient Clipping, Randomness, *etc.* is important

Supervised Learning: Regression & Classification

In regression problems, the target/output is real-valued

In classification problems, the target/output y is a category

$$y \in \{1, 2, \ldots, C\}$$

The input $\mathbf{x} = (x_1, \ldots, x_D)$, where

- Categorical: $x_i \in \{1, \ldots, K\}$
- Real-Valued: $x_i \in \mathbb{R}$

Discriminative Model: Only model the conditional distribution

$$p(y \mid \mathbf{x}, \boldsymbol{\theta})$$

Linear Regression, Logistic Regression, *etc.*

Generative Model: Model the full joint distribution

$$p(\mathbf{x}, y \mid \boldsymbol{\theta})$$

Naïve Bayes Classification, LDA, QDA

Models that have less natural probabilistic interpretations, such as SVM

Unsupervised Learning

Training data is of the form $\mathbf{x}_1, \ldots, \mathbf{x}_N$

Infer properties about the data

- Clustering: Group similar points together (k-Means, *etc.*)
- Dimensionality Reduction (PCA)
- Search: Identify patterns in data
- Density Estimation: Learn the underlying distribution generating data

Implementing Machine Learning Algorithms

Goal/Task

- Figure out what task you actually want to solve
- Think about whether you are solving a harder problem than necessary and whether this is desirable, *e.g.,* locating an object in an image vs simply labelling the image

Model and Choice of Loss Function

- Based on the task at hand, choose a model and a suitable objective
- See whether you can tweak the model, without compromising significantly on the objective, to make the optimisation problem convex

Algorithm to Fit Model

- Use library implementations for models if possible, *e.g.,* logistic regression, SVM, *etc.*
- If your model is significantly different or complex, you may have use to optimisation algorithms, such as gradient descent, directly
- Be aware of computational resources required, RAM, GPU memory, *etc.*

Implementing Machine Learning Algorithms

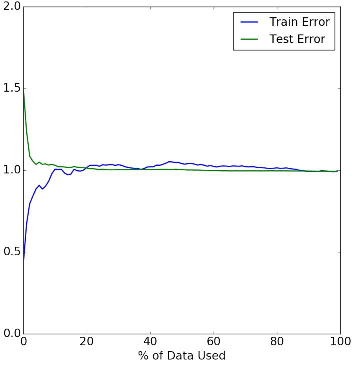
When faced with a new problem you want to solve using machine learning

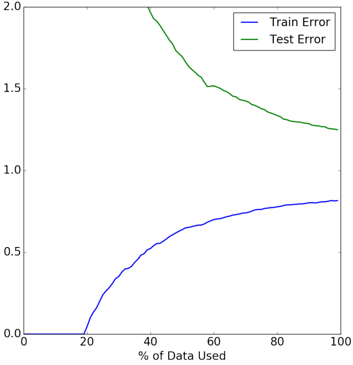
- Try to visualise the data, the ranges and types of inputs and outputs, whether scaling, centering, standardisation is necessary
- Determine what task you want to solve, what model and method you want to use
- As a first exploratory attempt, implement an easy out-of-the-box model, *e.g.,* linear regression, logistic regression, that achieves something non-trivial
- For example, when classifying digits make sure you can beat the 10% random guessing baseline
- Then try to build more complex models, using kernels, neural networks
- When performing exploration, be aware that unless done carefully, this can lead to overfitting. Keep aside data for validation and testing.

Learning Curves

- Learning curves can be used to determine whether we have high bias (underfitting) or high variance (overfitting) or neither. Then we can answer questions such as whether to perform basis expansion (when underfitting) or regularise (when overfitting).

- Plot the training error and test error as a function of training data size



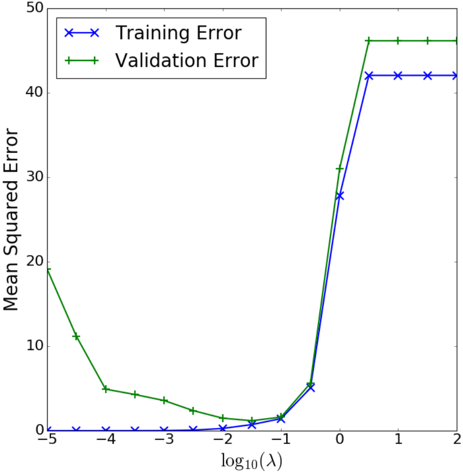


More data is not useful

More data would be useful

Training and Validation Curves

- Training and Validation Curves are useful to choose hyperparameters (such as λ for Lasso)
- Validation error curve is U-shaped



What do you need to know for the exam?

- The focus will be on testing your understanding of machine learning ideas, not prowess in calculus (though there will be some calculations)
- You do not need to remember all formulas. You will need to remember basic models such as linear regression, logistic regression, *etc.* However, the goal is to test your skills, not memory. You do not need to remember the forms of any probability distributions except Bernoulli and Gaussian.
- M.Sc. Students see the Hilary Term 2016 paper for reference (your paper will be simpler - that was a take-home final)
- Undergrads - See the M.Sc. paper for this course (will be posted early in Hilary term 2017). Your exam will be shorter. (Part C students need to attempt all 3 questions, Third years can do 2 out of 3)

A Holistic View of ML Methods

- Ultimately the goal is to have a more holistic view of machine learning
- Many ideas and tools can be applied in several settings: max-margin, (sparsity-inducing) regularization, kernels
- Understand the assumptions that different models and methods are making. For example, throughout the course we assume that all our data was i.i.d.
- Think of questions such as: Is there a lot of noise in your data? Are there outliers?
- Determine if you are overfitting or underfitting. And think of what approach you would use in either case

What next?

- This course has been a whirlwind tour of supervised and unsupervised machine learning methods
- Basic ideas and methods covered in the course will persist
- Other things such as what models to use, which flavours of gradient descent to use, etc. will change as research progresses
- To use machine learning in your work, you will need to keep applying the methods and follow the latest advances
- Try Kaggle competitions, your own projects, Biomod Project (link on website)

What next?

Two Courses in Hilary Term 2017

- Deep Learning for NLP (Phil Blunsom, *et al*. from DeepMind)
- Advanced Machine Learning (Computational Learning Theory)

Thanks!

- Please complete the official feedback forms
- Feel free to email me directly. Slip a note under my door or use piazza if you prefer being anonymous!