Machine Learning (ML)
DA222
Suggested reading materials

Soumitra Samanta

March 1, 2024

Contents

Lecture 1

Motivation (ML practical examples), Syllabus, Prerequisites and Resources

1.1 Tentative syllabus

Here are is a tentative syllabus:

- Motivation: What is Machine Learning (ML) and why we need to study ML?
- Data: Representation/Featurization, Normalization (after some idea in classification/regression), Data partition (train, val and test)
- Regression: Linear, Ridge, LASSO
- Classification: kNN, Bayes classifier, Linear discriminant analysis, Logistic regression, SVM, Decision tree, Random forest, Boosting, Ensemble methods
- Clustering: K-means, Hierarchical and agglomerative clustering/linkage clustering, Spectral graph clustering
- Dimensionality reduction and data visualization: PCA, Multidimensional scaling, Random projection, Issomap, t-SNE, UMAP etc.

- Kernel methods: Definition, Reproducing Kernel Hilbert space, kernel-SVM, kernel-PCA, kernel-Least square regression

- Low rank matrix completion and compressive sensing

- ML and Society: Fairness, Explainability and Environment effect

- Learning theory: Approximation and estimation error, Empirical risk minimisation, Convergence and consistency, Capacity measure of function classes, Shattering coefficient, VC dimension, Rademacher complexity, Occam's razor

1.2 Prerequisites

- Mathematics: No worries, we will touch some background when we need

  - Linear Algebra: *Vector space, Basis, Dimension, Matrix algebra (Addition, Multiplication, Trace, Inverse etc.), Eigen value and Eigen vectors, Positive definite matrices, Singular value decomposition etc.*
  - Multivariate Calculus: *Derivative, Partial derivative, Taylor series expansion, Chain rules etc.*
  - Basic Optimisation: *Convex set, Convex hull, Convex function, Gradient of a function, Hessian, Constrained and Unconstrained optimisation problem, Optimality condition*
  - Probability: *Definition, Random variables, Distribution function and their different variants, Conditional probability, Independence, Expectation, Variance, Moments, Entropy, Low of large numbers, Central limit theorem*

- Computer programming: *Any one from C/C++/Python (recommended for the class project and assignments)/MATLAB/Octave*

- Basic concept in Algorithms and Data Structure

1.3 ML and related books

We will follow multiple books for different topics. Here are some suggested books will follow in our course :

[1] Kevin Patrick Murphy, *Machine Learning: a Probabilistic Perspective*, MIT Press, 2012 [online]

[2] Kevin Patrick Murphy, *Probabilistic Machine Learning: An Introduction*, MIT Press, 2022 [online]

[3] Christopher M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006 [online]

[4] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar, *Foundations of Machine Learning*, MIT Press, Second Edition, 2018 [online]

[5] Shai Shalev-Shwartz and Shai Ben-David, *Understanding Machine Learning: From Theory to Algorithms*, Cambridge University Press, 2014 [online]

[6] Trevor Hastie, Robert Tibshirani and Martin Wainwright, *Statistical Learning with Sparsity: The Lasso and Generalizations*, CRC Press, 2015 [online]

[7] Solon Barocas, Moritz Hardt and Arvind Narayanan, *Fairness and Machine Learning: Limitations and Opportunities*, fairmlbook.org, 2019 [online]

[8] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning.* MIT Press, 1st edition, 2016. [online]

[9] R. O. Duda, P. E. Hart and D. G. Stork, *Pattern Classification and Scene Analysis*, 2nd ed., Wiley, New York, 2000

[10] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, Academic Press, San Diego, 1999

[11] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, 2nd ed., Academic Press, New York, 1990

[12] Luc Devroye, Laszlo Gyorfi, and Gabor Lugosi, *A Probabilistic Theory of Pattern Recognition*, 1st edition, Springer, 1996

1.4 ML and related tools

Here are some popular ML tools:

- Machine Learning in Python - `https://scikit-learn.org/stable/`
- ML in GPU - `https://rapids.ai/`
- PyTorch - https://pytorch.org/
- ...

1.5 ML datasets repository

You can find some datasets to evaluate yous ML models in *UCI Machine Learning Repository* (`https://archive.ics.uci.edu/ml/datasets.php`)

1.6 ML/AI top tier conference

- International Conference on Machine Learning (ICML) - `https://icml.cc/`
- Neural Information Processing Systems (NeurIPS) - `https://neurips.cc/`
- International Conference on Learning Representations (ICLR) - `https://iclr.cc/`
- Association for the Advancement of Artificial Intelligence (AAAI) - `https://www.aaai.org/`
- Computer Vision Foundation (CVF) - `https://openaccess.thecvf.com/menu`

1.7 ML top journals

- Journal of Machine Learning Research (JMLR) - `https://www.jmlr.org/`

1.8 For recent updates on ML you can follow the arXiv

You can go to Computer Science (CS) section in arXiv and under that you can find different branches of CS (like AI, ML, etc.).

- AI - `https://arxiv.org/list/cs.AI/recent`
- ML - `https://arxiv.org/list/cs.LG/recent`

Lecture 2

Data: Representation/Featurization, Normalization (after some idea in classification/regression), Data partition (train, val and test)

2.1 Suggested reading

Please go through the class slides.

2.2 Homework

[1] Consider *Minkowski distance* for $p = -2, -1,\ 0,\ 1,\ 1.5,\ 2,\ 3,\ 6,\ \infty$. Draw all the points within the the intervals has unit distance from the origin for all p's:

- 2D: $x \in [-1,\ 1]$ and $y \in [-1,\ 1]$
- 3D: $x \in [-1,\ 1]$, $y \in [-1,\ 1]$ and $z \in [-1,\ 1]$

[2] Create a random dataset in $\mathbf{R}^{100}$ of size 50000 with random class labels from $\{1, 2, 3, 4\}$. Now partition the data into the following subsets:

- Training: 50%
- Validation: 20%
- Testing: 30%

Plot (bar) the frequency of each class label for each subset.

Lecture 3

k- nearest neighbour (kNN) classifier

3.1 Suggested reading

For algorithm, please go through the class slides and for general discussion, you can go through Duda et al.'s [5] book *Chapter 4, Section 4.5: The nearest-neighbour Rule*. You can see the original paper [3] title with *Nearest neighbor pattern classification*, an online version can be found here.

For deep theoretical development, you can look at Devroye et al.'s [4] book *Chapter 5*. You may find some helpful results in *Chapter 19* of Shalev-Shwartz and Ben-David's [9] book as well.

3.2 Assignment

Implement kNN classifier and test on MNIST digit data with the following settings:

- You can use the supplied data in *.gz* form or can be downloaded from: `http://yann.lecun.com/exdb/mnist/`
- Strictly follow their data partition
- There is no validation set! Make your own validation set from the training set (20%)

- Use different similarity metrics ($p = 1,\ 2, and\ \infty$) and ($k = 1,\ 3, \cdots,\ 25$) calculate the classifier errors
- Plot ($3 - D$) the classification errors/accuracy for different p's and k's

Submission deadline: 19-02-2024 (11:59 PM)
Submission file format: `knn_mnist_data_your_name_version_no.ipynb`

Lecture 4

Bayesian decision rule

4.1 Suggested reading

For Bayesian decision theory, you can go through Duda et al.'s [5] book *Chapter 2, Section 2.1 - 2.6* and try to solve the corresponding exercises.

4.2 Assignment

Implement *Bayesian decision* rule for two class (River vs Non-river) problem discussed in the class with the following settings:

- You can use the supplied four satellite images in *.gif* form or can be downloaded from here: `https://www.isical.ac.in/~murthy/`
- Manually select 50 points from the river and 150 points from non-river from band4 as training set
- Use the image annotation tool provided or you can use your own tools
- Apply the Bayesian decision rule on four images to classify River and Non-river areas (Consider densities as normal in higher dimensions (discusses in class))
- Create a matrix of River and Non-river with the class labels River as 0 and Non-river as 255 then save as a image form
- Plot the confusion matrix of your model performance

Submission deadline: 27-02-2024 (11:59 PM)
Submission file format: `bdr_band_image_your_name_version_no.ipynb`

Lecture 5

Linear classifier (perceptron learning algorithm)

5.1 Suggested reading

The perceptron learning algorithms were proposed by Frank Rosenblatt in 1958 [7]. An online version of the original paper can be found here. Also, he wrote a technical report [8] in detail about perceptron.

You can find perceptron learning algorithms in any Machine Learning or Pattern Recognition book. Here are some references:

You can find the algorithm in Shalev-Shwartz et al. book [9] *Chapter 9, Section 9.1.2: Perceptron for Half-spaces* and for the convergence proof please go through the *Theorem 9.1* in [9].

Mohri et al. book [6] *Chapter 8, section 8.3.1: Perceptron algorithm* and for convergence proof *Theorem 8.8* [9].

For a brief history and original perceptron setup picture in Bishop's book [1] *Chapter 4, Section 4.1.7*.

5.2 Assignment

Implement the *perceptron learning algorithm* for a two class synthetic data we have discussed in the class with the following settings:

- Consider a two class classification problem and generate the dataset (100 points uniformly from each class) using the script form here:

`https://xlms.rkmvu.ac.in/pluginfile.php/4106/mod_assign/introattachment/0/gui_inputs.py?forcedownload=1`

- Implement the *perceptron leaning algorithm* discussed in the class with following three initialisation:
  - Randomly
  - With the help from your dataset
  - With zeros
- Plot the results (your linear separators) with the data points for the above three cases.

Submission deadline: 01-03-2024 (11:59 PM)
Submission file format: `perceptron_2d_data_your_name_version_no.ipynb`

Lecture 6

Support Vector Machine (SVM)

6.1 Suggested reading

Support Vector Machine is one of the most popular classification algorithms in Machine Learning. It was developed by Boser et al. [2] in 1992. The original paper can be found here. For rigorous theoretical results, You can go through Vapnik's (father of SVM) book [10]. Here are some book references we will follow for our course:

Mohri et al. book [6] *Chapter 5, Sections 5.1 - 5.2.3, 5.3 - 5.3.3* and Shalev-Shwartz et al. book [9] *Chapter 15.* For further reference, a modified version of SMO algorithm to train the SVM.

You can find plenty of implementation for the SVM. Here are some reference:

- Python based: `https://scikit-learn.org/stable/modules/svm.html#`
- Multiple environment support: `https://www.csie.ntu.edu.tw/~cjlin/libsvm/index.html`
- Multiple environment GPU support: `https://github.com/Xtra-Computing/thundersvm`

Bibliography

[1] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 1st edition, 2006.

[2] Bernhard E Boser, Isabelle M Guyon, and Vladimir Naumovich Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, pages 144–152. ACM, 1992.

[3] T. M. Cover and P. E. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967.

[4] Luc Devroye, Laszlo Gyorfi, and Gabor Lugosi. *A Probabilistic Theory of Pattern Recognition*. Springer, 1st edition, 1996.

[5] Richard Duda, Peter Hart, and David Stork. *Pattern Classification*. Wiley, 2nd edition, 2007.

[6] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. The MIT Press, 2nd edition, 2018.

[7] Frank Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408, 1958.

[8] Frank Rosenblatt. Principles of neurodynamics: Perceptrons and the theory of brain mechanisms. Technical report, Cornell Aeronautical Laboratory, 1961.

[9] Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 1st edition, 2014.

[10] Vladimir Naumovich Vapnik. *Statistical Learning Theory*. Wiley, 1st edition, 1998.