Machine Learning (ML)
DA222
Suggested reading materials

Soumitra Samanta

May 26, 2023

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/
- $\cdots$

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

1.9 Suggested reading

Please go through the class slides.

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 [11] book *Chapter 4, Section 4.5: The nearest-neighbour Rule*. You can see the original paper [8] 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 [10] book *Chapter 5*. You may find some helpful results in *Chapter 19* of Shalev-Shwartz and Ben-David's [27] book as well.

3.2 Assignment

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

- Download the dataset 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 ($p = 1,\ 2,\ \infty$) 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: 21-02-2023 (11:59 PM)

Lecture 4

Bayesian decision rule

4.1 Suggested reading

For Bayesian decision theory, you can go through Duda et al.'s [11] book *Chapter 2*.

4.2 Assignment

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

- Download the four satellite images 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: 03-03-2023 (11:59 PM)

Lecture 5

Regression: Linear least squares (LLS) regression

5.1 Suggested reading

You can find Linear least squares regression in any Machine Learning book.

For ordinary linear least square regression, you can go through *Chapter 9, Section 9.2: Linear Regression* of Shalev-Shwartz and Ben-David's book [27] or book by Mohri et al. [18] *Chapter 11, Section 11.3.1: Linear regression*.

Lecture 6

Regression: Ridge

6.1 Suggested reading

Ridge regression was first invented by Tikhonov [32] in 1943 in the context of integral equations. Later Horel and Kennard [14] introduced in Statistics in 1970. As in this case we are taking L_2 norm of the parameters as a penalty/constraint, some people called it least squares with L_2 regularization. For book reference, you can go through Bishop's book [4] *Chapter 3, Section 3.1.4: Regularized least squares* and Murphy's book [20] *Chapter 7, Section 7.5: Ridge regression*. You can find some numerical tricks in *Sub-section 7,5.2*.

Lecture 7

Regression: LASSO

7.1 Suggested reading

The *Least absolute shrinkage and selection operator*(LASSO) was invented by Tibshirani [30] in 1996 and later some more modifications in the literature [31]. For reference book by Hastie et al. [13] *Chapter 1 & 2*.

To correlate the different least square regression with statistical likelihoods and priors, please have a look at Murphy's book [20] *Chapter 7 (Table 7.1 give you the summary)*.

7.2 Assignment

Implement *Regression (LLS, Ridge, and LASSO)* for the power consumption problem we have discussed in the class with the following settings:

- Download the dataset from here: `https://github.com/Ratulchakra/Effect-Of-Climate-On-Energy-Consumption/blob/main/Consumption_Climate_Average_West_Bengal_2018_2019_2020.csv#L2`
- Consider *avg_max, avg_min, avg_rain* as independent variables and *Consumption* as dependent variable.
- Consider the data partitions as: Training - 50%, Validation - 20% and Testing - 30% (Please remember this is a time series type of data and we have discussed how to partition the data based on the date).

- Apply the LLS, Ridge, and LASSO on the data and plot the original and prediction for all the data's.
- Your implementation should be in general for three regressions: One function for each regression, which will take the inputs: data (X_i, Y_i), $i = 1 : n$; parameters like degree of the polynomials and regularisation parameter (λ) and return the output as W.
- Evaluate your model with the *root-mean-square error* (RMSE) and *Coefficient of determination* (R^2).

Submission deadline: 17-03-2023 (11:59 PM)

Lecture 8

Linear classifier (perceptron learning algorithm)

8.1 Suggested reading

The perceptron learning algorithms were proposed by Frank Rosenblatt in 1958 [24]. An online version of the original paper can be found here. Also, he wrote a technical report [25] 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 [27] *Chapter 9, Section 9.1.2: Perceptron for Half-spaces* and for the convergence proof please go through the *Theorem 9.1* in [27].

Mohri et al. book [18] *Chapter 8, section 8.3.1: Perceptron algorithm* and for convergence proof *Theorem 8.8* [27].

For a brief history and original perceptron setup picture [4]

8.2 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/2133/mod_resource/content/1/gui_inputs.py`

- 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: 29-03-2023 (11:59 PM)

Lecture 9

Support Vector Machine (SVM)

9.1 Suggested reading

Support Vector Machine is one of the most popular classification algorithms in Machine Learning. It was developed by Boser et al. [5] in 1992. The original paper can be found here. For rigorous theoretical results, You can go through Vapnik's (father of SVM) book [33]. Here are some book references we will follow for our course:

Mohri et al. book [18] *Chapter 5* and Shalev-Shwartz et al. book [27] *Chapter 15*.

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`

Lecture 10

Decision tree

10.1 Suggested reading

Decision tree (DT) is one of the widely used algorithm in machine learning. For motivating example, please go through the class slides. There are many versions for the DT and here is a survey on *fifty years celebration on classification and regression tree* [16]. There is a book by Breiman et al. title with *Classification and Regression Trees* [7] [available in our library]. Here are some popular algorithms based on different impurity measurements ID3 [22], C4.5 [23], CART [7]. For a summary, you can look into Shalev-Shwartz et al. book [27] *Chapter 18*. For detailed theory and related results, you can go through *Chapter 20* of Devroye et al. book [10].

Lecture 11

Random forest

11.1 Suggested reading

The main paper of Random forest [6] and for more theoretical details you can look into *Chapter 15* of Hastie et al. book [12]. Here is some recent work on *When do random forests fail?* [28].

Lecture 12

Boosting

12.1 Suggested reading

There is a whole book on Boosting by Schapire and Freund [26]. For motivating example, please go through the class slides. You can find the AdaBoost algorithm in *Algorithm 1.1* in [26], and for more details, you can go through the whole book. You can find the AdaBoost algorithm in Shalev-Shwartz et al. book [27] *Chapter 10, Section 10.2*, and I am encouraging you to go through the whole chapter of this book, particularly the *Theorem 10.2*.

Lecture 13

Clustering

13.1 Suggested reading

You can find *k-means* clustering algorithm in any machine learning book. You can go through the Shalev-Shwartz and Ben-David's book *Chapter 22, Section 22.2* [27] and I am also encouraging you to go through the LEMMA 22.1. The origin of the *k-means* algorithm idea was first proposed by Stuart P Lloyd in 1957. But it was published later in 1982 [15]. The algorithm was first used by James MacQueen in 1967 [17]

For *Hierarchical clustering (bottom-up or agglomerative and top-down or divisive)*, you can go through the *Chapter 14, Section 14.3.12* of Hastie at al. book [12]. I am encouraging you to read the *Chapter 10* of Duda et al.'s book [11].

Here is a survey on various clustering algorithms [35].

13.2 Assignment

Implement *k-means* clustering algorithm and test your implementation with the following settings:

- Download the four satellite images from here: `https://www.isical.ac.in/~murthy/`

- Consider *four* bands as one image and each image co-ordinate as your data point. So, each data point will be a *four*-dimensional vector, where each dimension will be the pixel value of the corresponding band image.

- Run your *k-means* algorithm on the above data for $k = 2, 3, 4, 5, 6, 7$ and plot those as images with proper colours.

Submission deadline: 29-04-2023 (11:59 PM)

Lecture 14

Dimensionality reduction

14.1 Suggested reading

You can find different dimensionality reduction techniques in Murphy's new book [21] *Chapter 20*. The book is freely downloadable from here. For **PCA** you can follow *Chapter 20, Section 20.1*.

For **Multi-dimensional scaling (MDS)** you can follow Härdle and Simar book *Chapter 17* [34], particularly, *The Classical Solution* under the **Metric Multidimensional Scaling** in *Chapter 17, Section 17.2*. A hard copy of the book is available in our library here. We have discussed a similar type of algorithm in our class.

For **Isomap**, see the original paper by Tenenbaum et al. [29] [you can download it from here]. For algorithm, you can directly jump into Table 1 here. For book reference, you can find the algorithm in Mohri et al.'s book [18], *Chapter 15, Section 15.3.1*.

For **random projection**, you see this paper [3] for image and text data. The proof of **Johnson-Lindenstrauss** can be found in [9] and [1].

Some modern data visualization techniques are t-SNE[implementation and detailed information from author's site] and UMAP[not for the exam.].

Lecture 15

Logistic regression

15.1 Suggested reading

Please go through the Murphy's old book [20] *Chapter 8*. For our purpose *Sections 8.1, 8.2 and 8.3* are sufficient, but I am encouraging you to go through the whole chapter or his new book [21] *Chapter 10*.

15.2 Assignment

Implement *Logistic regression* and test on MNIST digit data with the following settings:

- Download the dataset 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%)
- Report the accuracy and plot confusion matrix.

Submission deadline: 28-05-2023 (11:59 PM)

Lecture 16

Kernel methods

16.1 Suggested reading

You can find kernel methods in Murphy's new book [21] *Chapter 17* (The book is freely downloadable from here). For **SVM** - you can follow *Chapter 17, Section 17.3*, **Kernel ridge regression** - *Section 17.3.9* and **Kernel PCA** - the main paper and Murphy's new book [21] *Section 20.4.6*.

Here is a nice tutorial on kernel methods.

Lecture 17

ML and Society

17.1 Suggested reading

Whatever has been discussed in the class. If you are further interested, for *fairness in ML*, you can go through Barocas et al. book [2] (available here) and for *interpretability in ML*, you can check Christoph book [19] (available here).

Bibliography

[1] Dimitris Achlioptas. Database-friendly random projections: Johnson-lindenstrauss with binary coins. *Journal of Computer and System Sciences*, 66(4):671–687, 2003.

[2] Solon Barocas, Moritz Hardt, and Arvind Narayanan. *Fairness and Machine Learning*. fairmlbook.org, 1st edition, 2019.

[3] Ella Bingham and Heikki Mannila. Random projection in dimensionality reduction: Applications to image and text data. In *In Knowledge Discovery and Data Mining*, pages 245–250. ACM, 2001.

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

[5] 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.

[6] Leo Breiman. Random forests. *Machine Learning*, 45(1):5 – 32, 2001.

[7] Leo Breiman, Jerome H. Friedman, Richard A. Olshen, and Charles J. Stone. *Classification and Regression Trees*. Chapman and Hall/CRC, 1st edition, 1984.

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

[9] Sanjoy Dasgupta and Anupam Gupta. An elementary proof of a theorem of johnson and lindenstrauss. *Random Structures & Algorithms*, 22(1):60–65, 2003.

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

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

[12] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer, 2nd edition, 2009.

[13] Trevor Hastie, Robert Tibshirani, and Martin Wainwright. *Statistical Learning with Sparsity: The Lasso and Generalizations*. CRC Press, 1st edition, 2015.

[14] Arthur E. Hoerl and Robert W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

[15] Stuart P. Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.

[16] Wei-Yin Loh. Fifty years of classification and regression trees. *International Statistical Review*, 82(3):329 – 348, 2014.

[17] James MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of Fifth Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, 1967.

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

[19] Christoph Molnar. *Interpretable Machine Learning - A Guide for Making Black Box Models Explainable*. Lulu.com, 2022. `https://christophm.github.io/interpretable-ml-book/`.

[20] Kevin Patrick Murphy. *Machine Learning: a Probabilistic Perspective*. The MIT Press, 1st edition, 2012.

[21] Kevin Patrick Murphy. *Probabilistic Machine Learning: An introduction*. The MIT Press, 1st edition, 2022.

[22] John Ross Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986.

[23] John Ross Quinlan. *C4.5 : programs for machine learning*. Morgan Kaufmann, 1st edition, 1992.

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

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

[26] Robert E. Schapire and Yoav Freund. *Boosting: Foundations and Algorithms*. The MIT Press, 1st edition, 2012.

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

[28] Cheng Tang, Damien Garreau, and Ulrike von Luxburg. When do random forests fail? In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[29] Joshua B. Tenenbaum, Vin de Silva, and John C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Sciences*, 290(5500):2319–2323, 2000.

[30] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B*, 58(1):267–288, 1996.

[31] Robert Tibshirani. Regression shrinkage and selection via the lasso: a retrospective. *Journal of the Royal Statistical Society. Series B*, 73(3):273–282, 2011.

[32] Andrey Nikolayevich Tikhonov. On the stability of inverse problems. *Doklady Akademii Nauk SSSR*, 39(5):195–198, 1943.

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

[34] Léopold Simar Wolfgang Karl Härdle. *Applied Multivariate Statistical Analysis*. Springer, 4th edition, 2015.

[35] Dongkuan Xu and Yingjie Tian. A comprehensive survey of clustering algorithms. *Annals of Data Science*, 2(2):165–193, 2015.