Introduction to machine learning

Maksim Kretov

Lecture 1: Basics of machine learning

5vision, 2017

Course information

Course

10 lectures + 2 seminars; February-May 2017.

Schedule and up-to-date syllabus

https://goo.gl/xExEuL

Contact information and discussion

Maksim Kretov (kretovmk@gmail.com)

Slack group: https://miptmlcourse.slack.com

OR https://goo.gl/Nz9S19

OR to get an invite, send e-mail to kretovmk@gmail.com.

Plan for the lecture

A. ML overview
    1. Types of tasks
    2. Courses and books
B. Present course
C. Introduction to ML
    1. Formulation of problem
    2. Deterministic view
    3. Probabilistic perspective
D. Homework

A.1 ML overview: Types of tasks

Classification (supervised)

Document classification, e-mail spam filtering

Face recognition / object detection

Image classification

Regression (supervised)

Temperature at any location inside a building

Tomorrow's stock market price

Knowledge discovery (unsupervised)

User segmentation (e-commerce)

Classification of stars based on clustering techniques

Separate signals into their different sources

A.1 ML overview: Types of tasks

ML = machine learning algorithms. Learning is:

A computer program is said to learn from experience **E** with respect to some class of tasks **T** and performance measure **P** , if its performance at tasks in **T** , as measured by **P**, improves with experience **E** [5].

No free lunch theorem

Averaged over all possible data generating distributions, every classification algorithm has the same error rate when classifying previously unobserved points [1, 4].

=> No machine learning algorithm is “universally” any better than any other.

A.2 ML overview: Courses and books

Machine learning

Courses

1*. cs229 (Stanford)
2*. ML specialization (MIPT/Yandex)
3. 9.520 (MIT)

Books

1. Kevin P. Murphy, Machine Learning: A probabilistic perspective.
2. David Barber, Bayesian Reasoning and Machine Learning.
3. Sergios Theodoridis, Machine Learning: A Bayesian and optimization perspective.

Neural networks and deep learning

Courses

1*. cs231n (Stanford)
2. 6.S191 (MIT)
3. cs20si (Stanford)
4. Hugo Larochelle's course on neural networks.

Books

1*. Ian Goodfellow, Yoshua Bengio and Aaron Courville, Deep Learning.
2. Same books as for ML.

B. Present course

<u>Math and basics of ML</u>	(1-2)	← **Today and next**	*Theoretical tasks*
Some of ML methods	(3)		
Seminar on ML basics	(4)		
Basics of neural networks	(5)	← **Start playing with NNs**	*Practical tasks*
Deep learning overview	(6)		
Training deep networks	(7)		
DL for Computer Vision	(8-9)	**Solving more complex ML tasks using NNs**	
DL for time series prediction	(10-11)		
Concluding seminar	(12)		

B. Present course

Focus areas:

ML foundations

Introduction to deep learning for computer vision

Tools:

Python; Frameworks: Tensorflow/Theano + Keras

Not covered or minimal focus:

Specific ML methods like decision trees, mixture models, SVM etc.

Goals:

Big picture of ML and ability to use DL methods for CV tasks.

C.1 Intro to ML: Formulation of problem

Supervised learning:

Training set: $\boldsymbol{D} = \{(\mathbf{x}_{\boldsymbol{n}}, y_n), n = 1, .. N\}$ *(inputs and labels!)*

Y are class ids => classification *(make partition of space)*

$Y \in \mathrm{R}$ => regression *(show how data are generated)*

$\mathbf{X}$ – $(N \times d)$ design matrix (features are d-dimensional)

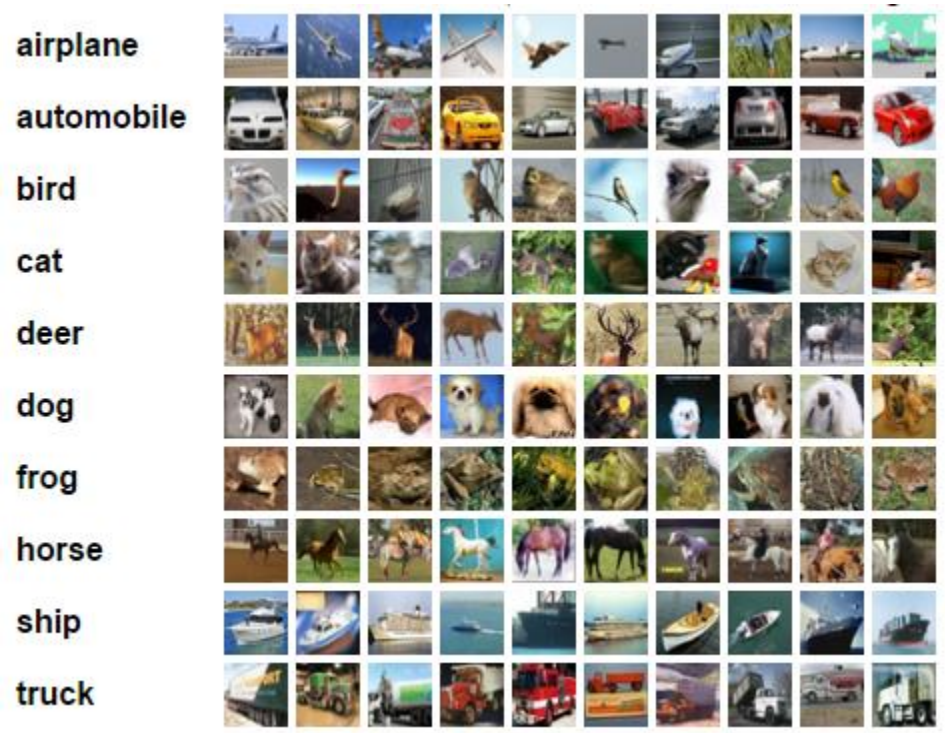
Task to solve:

Predict y^* for new input $\mathbf{x}^*$ => **focus on accurate prediction**

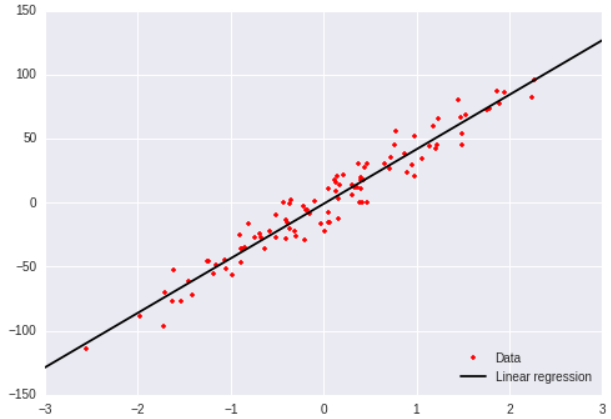
C.1 Intro to ML: Formulation of problem

CIFAR10 dataset* (60k images)

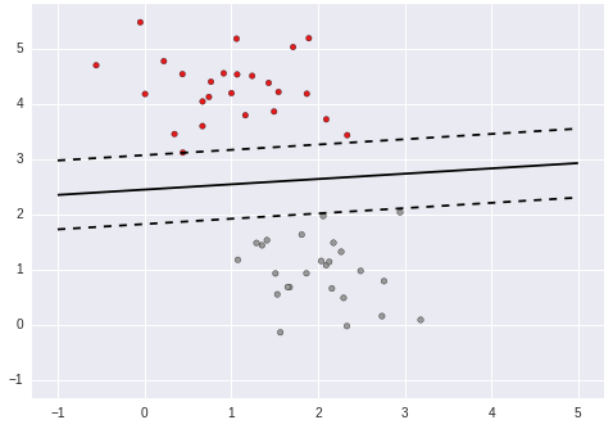
$\{\boldsymbol{y}_n\}$ – class ids $\{\mathbf{x}_n\}$ – 3-dimensional array (32x32x3)



<u>Regression</u>
example:
dim(x) = 1
dim(y) = 1



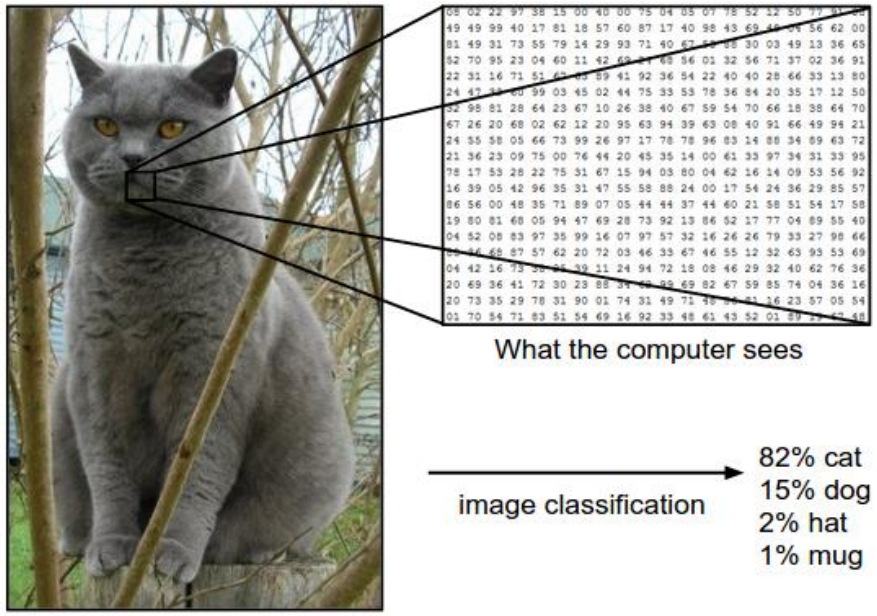
<u>Classification</u>
example:
dim(x) = 2
y = {0,1}



* https://www.cs.toronto.edu/~kriz/cifar.html

C.1 Intro to ML: Formulation of problem

Another classification task*



Input features $\mathbf{x}$

Output labels y

Once again. Task: Predict y for new input $\mathbf{x}$. But how to do it?

* http://cs231n.github.io/

C.2 Intro to ML: Deterministic view

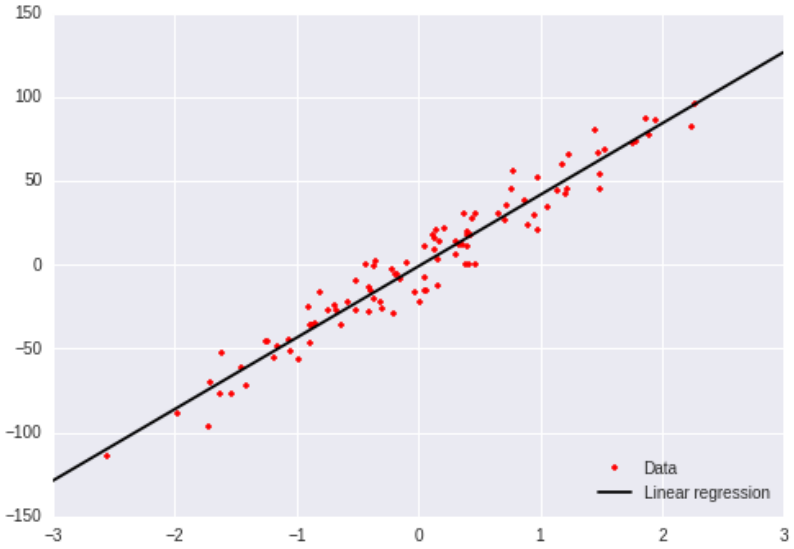
Naïve approach: deterministic point of view

1. Select parametric model $f_\theta(\mathbf{x})$
2. Select loss function:

$$L(\theta) = \sum_{n=1}^{N} L\big(y_n, f_\theta(\mathbf{x_n})\big) \xrightarrow{\text{example}} \sum_{n=1}^{N} \big(y_n - f_\theta(\mathbf{x_n})\big)^2$$

3. We can now solve optimization task!

(minimize $L(\theta)$ w.r.t. θ)



And we know very well how to solve optimization tasks: if not analytically, then by means of gradient methods, simulated annealing, genetic algorithms etc.

C.2 Intro to ML: Deterministic view

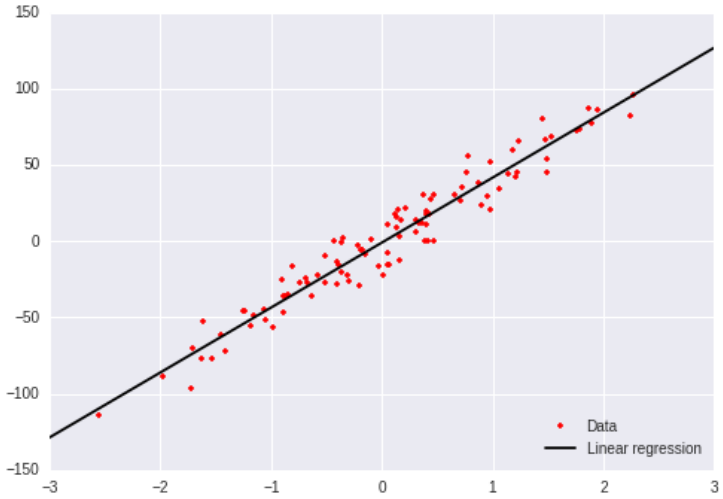
Naïve approach: deterministic point of view - regression

1. Select parametric model $f_\theta(\mathbf{x})$
2. Select loss function:

$$L(\theta) = \sum_{n=1}^{N} L\left(y_n, f_\theta(\mathbf{x_n})\right) \xrightarrow{\text{example}} \sum_{n=1}^{N} \left(y_n - f_\theta(\mathbf{x_n})\right)^{\mathbf{2}}$$

3. We can now solve optimization task!

(minimize $L(\theta)$ w.r.t. θ; convex surface)



What is wrong with naïve approach?

No confidence intervals, implicit assumptions etc.

Exercise: *Derive LS formula for linear regression.*

Linear regression

$y = f_\theta(\mathbf{x}) = \boldsymbol{\theta}^{\boldsymbol{T}}\mathbf{x} + \varepsilon$

ε – noise; $\boldsymbol{\theta}$ includes intercept (θ_0)

With least squares (LS) loss function: $\theta_{opt} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\boldsymbol{y}$

C.2 Intro to ML: Deterministic view

Naïve approach: deterministic point of view - regression

Let's further improve our approach (Ridge regression):

Introduce new constraint:

$$L(\theta) = \sum_{n=1}^{N} \left(y_n - f_\theta(\mathbf{x_n})\right)^2 + \boldsymbol{\lambda}\left\|\tilde{\theta}\right\|^2$$

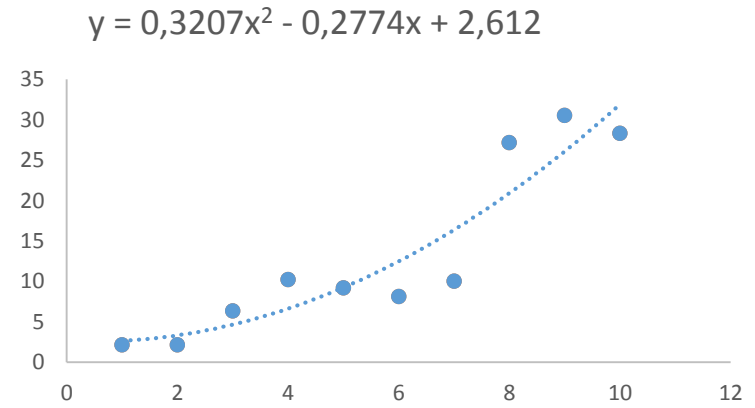
$$\tilde{\theta} = (0, \theta_1, \theta_1, \ldots \theta_d)$$

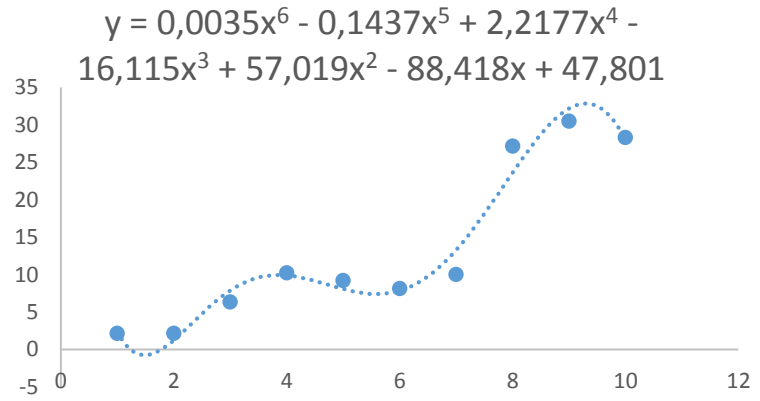
What for?

to "simplify" the model (prevent overfitting)

to improve numerical stability (inverse matrices)

Exercise: *Derive LS formula with regularization.*





C.2 Intro to ML: Deterministic view

Naïve approach: deterministic point of view - classification

Same steps, but different loss function.

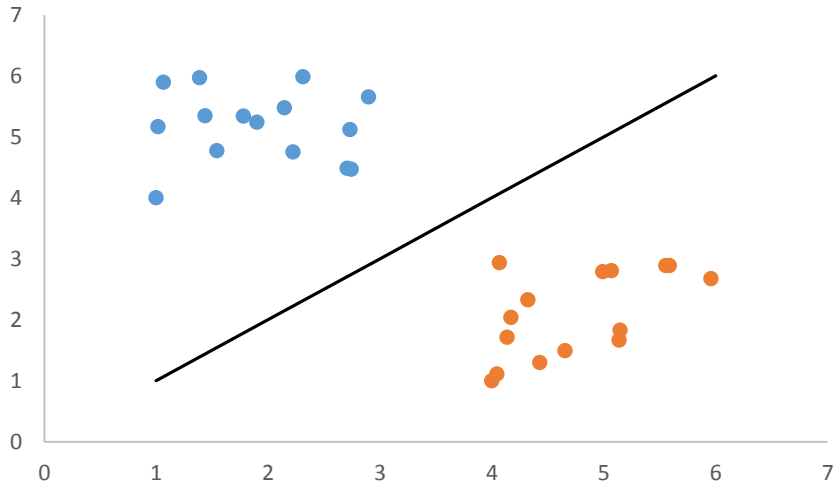
Why? – want to minimize probability of error.

For 2 classes:

$$L(\theta) = -\sum_{n=1}^{N} \left(y_n \ln f_\theta(\mathbf{x_n}) + (1 - y_n) \ln\left(1 - f_\theta(\mathbf{x_n})\right) \right)$$

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

Exercise: *Derive formula for* $\nabla_\theta L(\theta)$.



Logistic regression (!)

$f_\theta(\mathbf{x_n}) = \sigma(\theta^T \mathbf{x_n})$

$\theta^i = \theta^{i-1} + \alpha \nabla_\theta L(\theta)$

Or second order methods

10 minute break..

C.3 Intro to ML: Probabilistic perspective

Moving towards probabilistic perspective..

Still parameter, not RV!

Generative classifier

Discriminative classifier

Key ingredients:

Joint probability $p(\mathbf{x}, y|\theta) = p(\mathbf{x}|y, \theta)p(y|\theta) = p(y|\mathbf{x}, \theta)p(\mathbf{x}|\theta)$

=> We deal with one of parametric models:

$p(\mathbf{x}|y, \theta)$ - for generative classifiers (QDA)

$p(y|\mathbf{x}, \theta)$ - for discriminative classifiers (Logistic regression)

C.3 Intro to ML: Probabilistic perspective

Framework "**Maximum likelihood estimation**" (MLE)

Key idea: select θs which give the highest probability for the data under study to be generated.

Given:

$$\boldsymbol{D} = \{(\mathbf{x}_{\boldsymbol{n}}, y_n), n = 1, .. N\}$$

Parametric distribution $p(\mathbf{x}, y|\theta)$

Select optimal parameters:

$$\theta_{opt} = \text{argmax}_{\theta} p(\boldsymbol{D}|\theta)$$

C.3 Intro to ML: Probabilistic perspective

Framework "**Maximum likelihood estimation**" (MLE)

Example: Linear regression – discriminative classifier!

Given:

$$\boldsymbol{D} = \{(\mathbf{x}_{\boldsymbol{n}}, y_n), n = 1, .. N\}$$

Total parametric distribution $p(\mathbf{x}, y|\theta^{tot}) = p(y|\mathbf{x}, \theta^y)p(\mathbf{x}|\theta^x)$

Only care about y : $p(y|\mathbf{x}, \theta^y) = \mathrm{Norm}(y|\theta^T\mathbf{x}, \sigma^2)$

Equation for optimal parameters:

$$\theta^y_{opt} = \mathrm{argmax}_\theta p(Y|\mathbf{X}, \theta^y) = \mathrm{argmin}_\theta \sum_{n=1}^{N}(y_n - \theta^T\mathbf{x}_{\boldsymbol{n}})^{\mathbf{2}}$$

Exercise: *Prove the last transition*

C.3 Intro to ML: Probabilistic perspective

Rigorous approach: "**Bayesian inference**"

Key idea: Treat θ as random variables and introduce prior distribution on θ.

$\boldsymbol{D} = \{(\mathbf{x}_{\boldsymbol{n}}, y_n), n = 1, .. N\}$ and new input $\mathbf{x}^*$

Likelihood $p(\boldsymbol{D}|\theta)$ and prior distribution $p(\theta)$.

Key difference: we are going to exploit full posterior distribution of θ!

Key steps:

1. Infer posterior distribution $p(\theta|\boldsymbol{D})$.
2. Calculate distribution for new input: $p(\mathbf{x}, y|\boldsymbol{D}) = \int p(\mathbf{x}, y|\theta)p(\theta|\boldsymbol{D})d\theta$

C.3 Intro to ML: Probabilistic perspective

Framework "**Maximum a posteriori**" (MAP)

Key idea: Let's simplify Bayesian inference and make point estimate.

Given:

$$\boldsymbol{D} = \{(\mathbf{x}_{\boldsymbol{n}}, y_n), n = 1, .. N\}$$

Distribution $p(\theta|\boldsymbol{D}) = \frac{p(\boldsymbol{D}|\theta)p(\theta)}{p(\boldsymbol{D})}$

Encapsulates our prior knowledge about θ

Select optimal parameters:

$$\theta_{opt} = \text{argmax}_\theta p(\theta|\boldsymbol{D}) = \text{argmax}_\theta p(\boldsymbol{D}|\theta)p(\theta)$$

Exercise: *Show that establishing Gaussian prior on θ is equivalent to L2 regularization.*

C.3 Intro to ML: Probabilistic perspective

Notion of **loss function** (utility function)

Key idea: let's quantify, how harmful deviations from target are!

Loss function $L(y, \hat{y})$, where $\hat{y} = f(\mathbf{x}, \theta)$- prediction label (value) for $\mathbf{x}$

So, ideal loss function for optimization:

$$f_{opt} = \text{argmin}_f \int L(y, f(\mathbf{x}))p(y, \mathbf{x})dyd\mathbf{x}$$

Examples of loss functions:

1. Error frequency: $L(y, \hat{y}) = \text{I}(y \neq \hat{y})$
2. Mean absolute error (MAE): $L(y, \hat{y}) = |y - \hat{y}|$
3. Mean squared error (MSE): $L(y, \hat{y}) = (y - \hat{y})^2$

C.3 Intro to ML: Probabilistic perspective

Empirical risk minimization approach

Deterministic view + probabilistic perspective

$$\theta^{opt} = \mathbf{argmin}_\theta \mathbf{E}_{p(\mathbf{x},y)}\left[L\left(y, f_\theta(\mathbf{x})\right)\right] = \mathbf{argmin}_\theta \sum_{\mathbf{x}} \sum_{y} L\left(y, f_\theta(\mathbf{x})\right) p(\mathbf{x}, y) \qquad (1)$$

Problem: we don't know correct "nature's" distribution (otherwise task is solved)

Solution: replace correct $p(\mathbf{x}, y)$ with empirical distribution $p_{emp}(\mathbf{x}, y)$:

$$p(\mathbf{x}, y) \approx p_{emp}(\mathbf{x}, y) = \frac{1}{N} \sum_{n=1}^{N} \delta(\mathbf{x}, \mathbf{x}_n)\delta(y, y_n) \qquad (2)$$

=> Formula for fitting the model within ERM framework:

$$\theta^{opt} = \mathbf{argmin}_\theta \frac{1}{N} \sum_{n=1}^{N} L\left(y_n, f(\mathbf{x_n}, \theta)\right) + \lambda P(\theta) \qquad (3)$$

Penalised empirical risk

Classification of new input: $\hat{y} = f(\mathbf{x}^*, \theta^{opt})$

C.3 Supervised: Probabilistic perspective

Empirical risk minimization approach

Benefits

In the limit of large amount of data empirical distribution is correct.

Procedure is straightforward.

Drawbacks

No measure of uncertainty for predictions.

For small datasets underlying assumptions seem to be extreme.

C.3 Supervised: Probabilistic perspective

General algorithm for SL

1. Engineer/Select good features
2. Learn model (class of models)
3. Inference: get y^* for unseen $\mathbf{x}^*$

All steps are difficult!

Or we can use original features (for now..)

What model to select?
How to fit its parameters?
Should we use just one model or average over them?

Depends on how we performed on step 2.

Next lecture

1. Introduction to machine learning (continued).
    a) Bias-variance tradeoff.
    b) Generalization ability of classifiers.

2. Decision theory. How to connect ERM, MLE, MAP etc.
    a) Bayesian approach.
    b) Frequentist approach.

3. Bayesian model selection.

D. Homework

For all:

1. ML: [1] ch. 1, 2, 3 or [2] ch.8, 12, 13 or [3] ch.2, 3.
2. Exercises from presentation.
3. Python (Jupyter, Numpy).

Optional for enthusiasts:

1. Start ML course.

Refs

1. Kevin P. Murphy, Machine Learning: A probabilistic perspective.
2. David Barber, Bayesian Reasoning and Machine Learning.
3. Sergios Theodoridis, Machine Learning: A Bayesian and optimization perspective.
4. Ian Goodfellow, Yoshua Bengio and Aaron Courville, Deep Learning.
5. T. M Mitchell, Machine Learning.