10-606 Mathematical Foundations for Machine Learning

Machine Learning Department
School of Computer Science
Carnegie Mellon University

Sets, Data Types, and Functions

Matt Gormley
Lecture 2
August 29, 2018

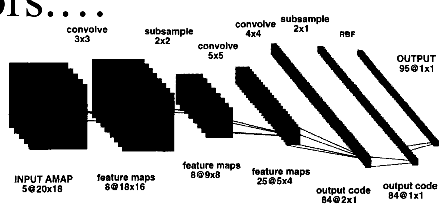
Computer Vision

Correction...

4. Learning to recognize images

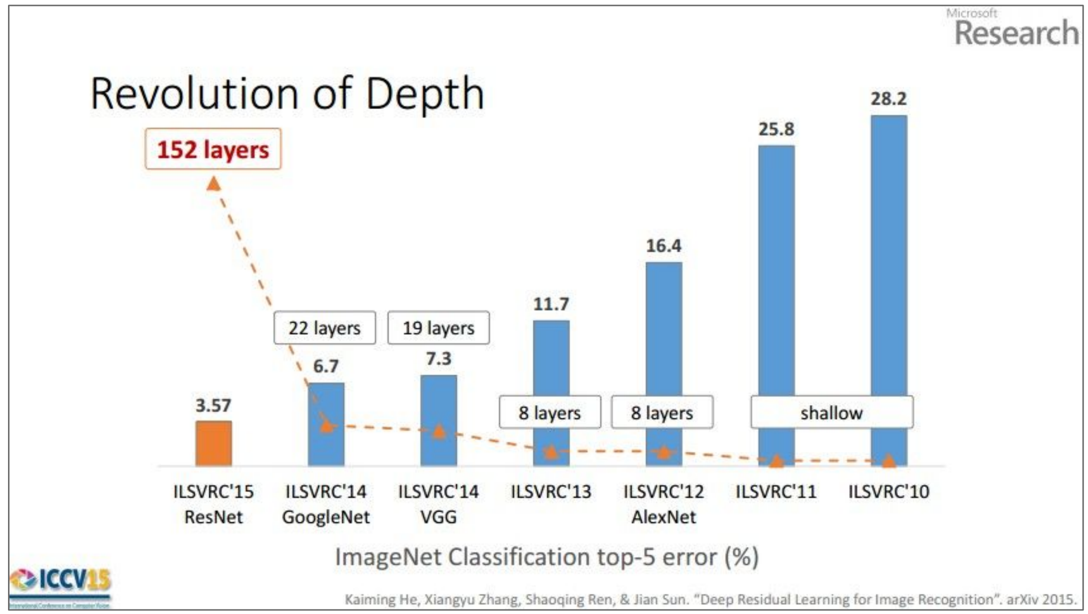
THEN

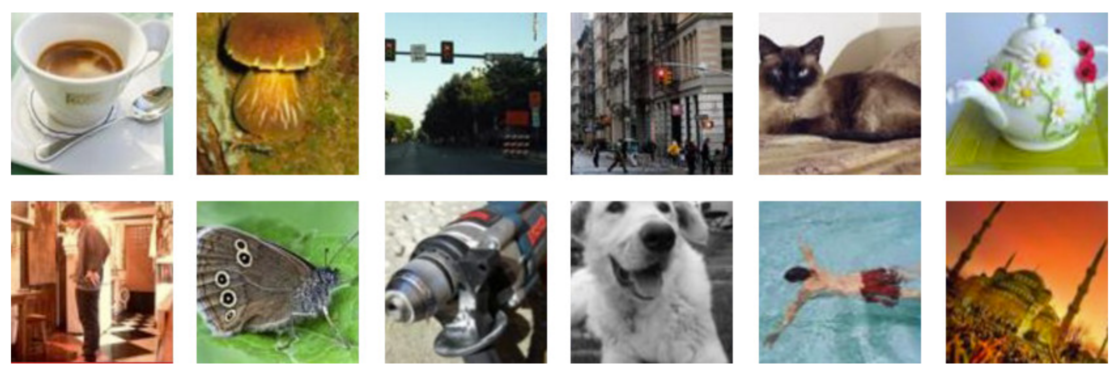
"…The recognizer is a convolution network that can be spatially replicated. From the network output, a hidden Markov model produces word scores. The entire system is globally trained to minimize word-level errors.…"



(LeCun et al., 1995)

NOW





Images from https://blog.openai.com/generative-models/

SYLLABUS HIGHLIGHTS

Syllabus Highlights

The syllabus is located on the course webpage:

http://www.cs.cmu.edu/~mgormley/courses/606-607-f18

The **course policies** are **required** reading.

606/607 Syllabus Highlights

- **Grading**: 55% homework, 10% in-class quizzes, 30% final exam, 5% participation
- **Final Exam**:
  - 606: Mini-I final exam week, date TBD
  - 607: Mini-II final exam week, date TBD
- **In-Class Quizzes**: always announced ahead of time
- **Homework**: 4 assignments with written / programming portions
  - 2 grace days for the unexpected
  - Late submissions: 80% day 1, 60% day 2, 40% day 3, 20% day 4
  - No submissions accepted after 4 days w/o extension
  - Extension requests: see syllabus
- **Recitations**: Fridays, same time/place as lecture (optional, interactive sessions)
- **Readings**: required, online, recommended for after lecture
- **Technologies**: Piazza (discussion), Gradescope (homework), Canvas (gradebook only)
- **Academic Integrity**:
  - Collaboration encouraged, but must be documented
  - Solutions must always be written independently
  - No re-use of found code / past assignments
  - Severe penalties (i.e. failure)
- **Office Hours**: posted on Google Calendar on "People" page

606/607 Syllabus Highlights

- **Grading**: 55% homework, 10% in-class quizzes, 30% final exam, 5% participation
- **Final Exam:**
  - 606: Mini-I final exam week, date TBD
  - 607: Mini-II final exam week, date TBD
- **In-Class Quizzes**: always announced ahead of time
- **Homework**: 4 assignments with written / programming portions
  - 2 grace days for the unexpected
  - Late submissions: 80% day 1, 60% day 2, 40% day 3, 20% day 4
  - No submissions accepted after 4 days w/o extension
  - Extension requests: see syllabus
- **Recitations**: Fridays, same time/place as lecture (optional, interactive sessions)
- **Readings**: required, online, recommended for after lecture
- **Technologies**: Piazza (discussion), Gradescope (homework), Canvas (gradebook only)
- **Academic Integrity**:
  - Collaboration encouraged, but must be documented
  - Solutions must always be written independently
  - No re-use of found code / past assignments
  - Severe penalties (i.e. failure)
- **Office Hours**: posted on Google Calendar on "People" page

606/607 Syllabus Highlights

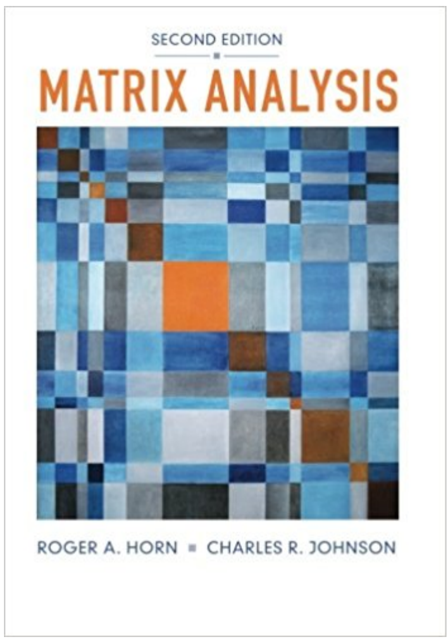
- **Grading:** 55% homework, 10% in-class quizzes, 30% final exam, 5% participation
- **Final Exam:**
  - 606: Mini-I final exam week, date TBD
  - 607: Mini-II final exam week, date TBD
- **In-Class Quizzes:** always announced ahead of time
- **Homework:** 4 assignments with written / programming portions
  - 2 grace days for the unexpected
  - Late submissions: 80% day 1, 60% day 2, 40% day 3, 20% day 4
  - No submissions accepted after 4 days w/o extension
  - Extension requests: see syllabus
- **Recitations:** Fridays, same time/place as lecture (optional, interactive sessions)
- **Readings:** required, online, recommended for after lecture
- **Technologies:** Piazza (discussion), Gradescope (homework), Canvas (gradebook only)
- **Academic Integrity:**
  - Collaboration encouraged, but must be documented
  - Solutions must always be written independently
  - No re-use of found code / past assignments
  - Severe penalties (i.e. failure)
- **Office Hours:** posted on Google Calendar on "People" page

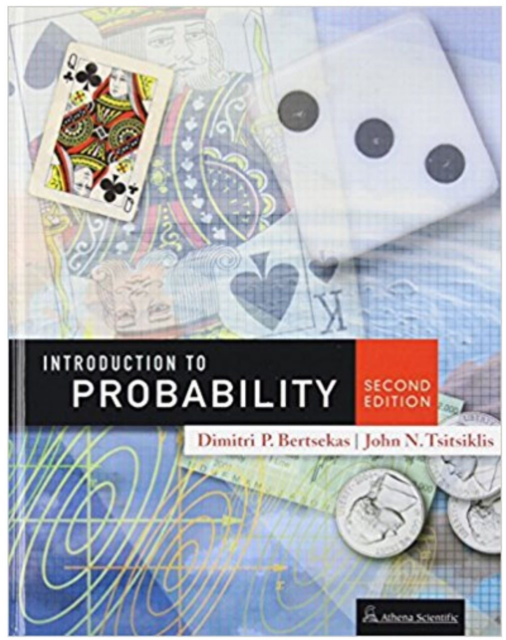
Lectures

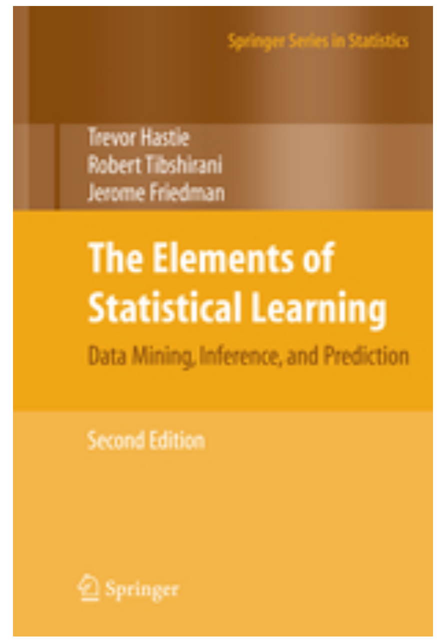
- You should ask lots of questions
  - Interrupting (by raising a hand) to ask your question is strongly encouraged
  - Asking questions later on Piazza is also great
- When I ask a question…
  - I want you to answer
  - Even if you don't answer, think it through as though I'm about to call on you
- Interaction improves learning (both in-class and at my office hours)

Textbooks

These are optional, but highly recommended as an alternate presentation of the material







Expected Background

10-606 (Math Background 4 ML)

You should be familiar with some of the following…

- Calculus:
  - can take scalar derivatives
  - can solve scalar integrals
- Linear Algebra:
  - know basic vector operations
  - seen matrix multiplication
- Probability:
  - seen the basics: conditioning, Bayes Rule, etc.
- Programming:
  - know some Python
    OR
    have sufficient programming background to pick up the basics of Python

But we'll offer practice to make sure you can catch up on your weaker areas

10-607 (CS Background 4 ML)

You should…

- be comfortable with all the topics listed for 10-606
- ideally, have the mathematical maturity of someone who completed 10-606 **because** it will aide in understanding the motivating examples from machine learning

That said, the content of 10-607 is designed stand alone

Q&A

Q: Is this course right for me?

A:
- If you're a Master's or PhD and you lack some of the prerequisite material for 10-601/701, this is definitely the right course for you!
- If you're a Master's or PhD and you studied the prerequisite material for 10-601/701… but it was a long time ago, this is certainly the right course for you.
- If you're an undergrad, I would recommend the usual prereq sequence required for 10-601/701.
- For ugrad/MS/PhD: If you tried taken an Intro ML course here and felt a bit lost in all the math/CS, this is a great place to start.

Q: What calculus textbook would you recommend?

A: Good question… I'm still working on that one.

Q&A

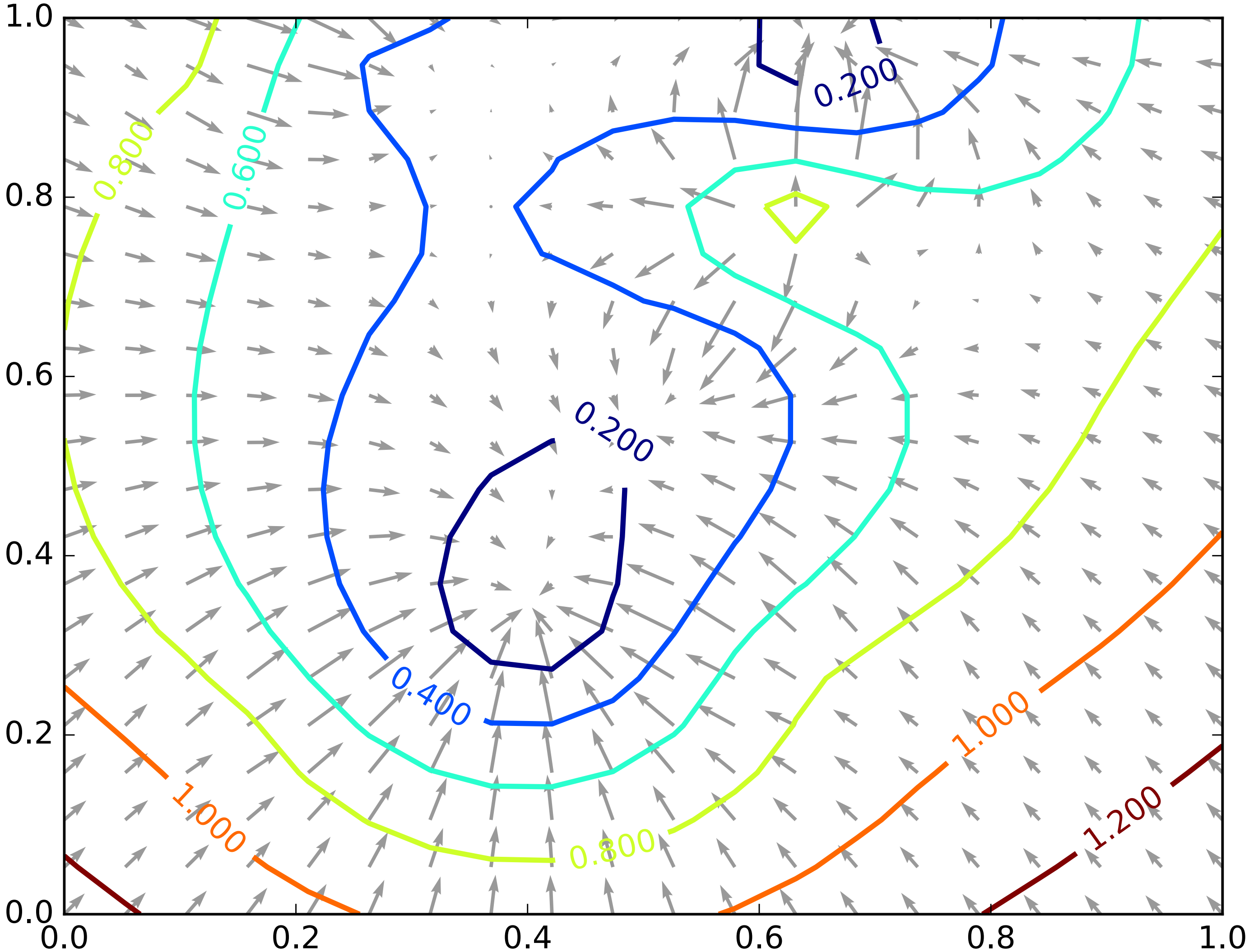
Q: What do I do if I'm feeling a bit lost in the math and CS in 10-606/607?

A: Let me know ASAP! (Do so in office hours, Piazza note, the middle of class, etc.) Our goal is to provide you with a learning environment in which you thrive. We'll certainly make adjustments if we need to.

MOTIVATION: SETS & TYPES

Sets, Types, and Functions show up **everywhere** in Machine Learning

(Negative) Gradients

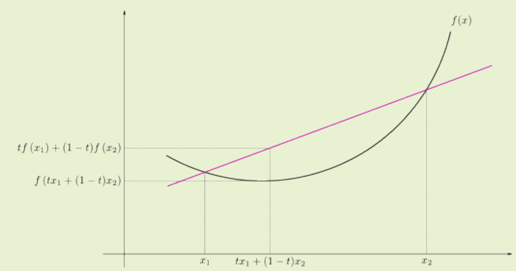


These are the **negative** gradients that
Gradient **Descent** would follow.

Convexity

Suppose we wish to define convexity of a function…

We could draw a picture.



… but that's a bit informal.

So instead, we could offer a mathematical definition.

Function $f : \mathbb{R}^M \to \mathbb{R}$ is **convex**
if $\forall\ \mathbf{x}_1 \in \mathbb{R}^M, \mathbf{x}_2 \in \mathbb{R}^M, 0 \leq t \leq 1$:

$$f(t\mathbf{x}_1 + (1-t)\mathbf{x}_2) \leq t f(\mathbf{x}_1) + (1-t) f(\mathbf{x}_2)$$

… but even this definition requires some carefully defined objects (the vectors, a function, the set of reals, the set of real-valued vectors of length M, etc.)

Slide adapted from William Cohen

Data for ML

What is the object we talk about more in Machine Learning than anything else?

Our data!

$$\mathcal{D} = \{(\mathbf{x}^{(n)}, y^{(n)})\}_{n=1}^{N}$$

The data consists of a set of tuples

$$\mathcal{D} = \{(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), \ldots, (\mathbf{x}^{(N)}, y^{(N)})\}$$

The importance of sets…

- Gaussian Discriminant Analysis and Gaussian Mixture Models are almost identical
- There’s really only one practical difference
- Can you spot it?

See next two slides…

Gaussian Discriminant Analysis

Data: $\mathcal{D} = \{(\mathbf{x}^{(i)}, \mathbf{z}^{(i)})\}_{i=1}^{N}$ where $\mathbf{x}^{(i)} \in \mathbb{R}^M$ and $z^{(i)} \in \{1, \ldots, K\}$

Generative Story:

$$z \sim \text{Categorical}(\phi)$$

$$\mathbf{x} \sim \text{Gaussian}(\boldsymbol{\mu}_z, \boldsymbol{\Sigma}_z)$$

Model:

Joint: $p(\mathbf{x}, z; \boldsymbol{\phi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = p(\mathbf{x}|z; \boldsymbol{\mu}, \boldsymbol{\Sigma})p(z; \boldsymbol{\phi})$

Log-likelihood:

$$\begin{aligned}\ell(\boldsymbol{\phi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) &= \log \prod_{i=1}^{N} p(\mathbf{x}^{(i)}, z^{(i)}; \boldsymbol{\phi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) \\ &= \sum_{i=1}^{N} \log p(\mathbf{x}^{(i)}|z^{(i)}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) + \log p(z^{(i)}; \boldsymbol{\phi})\end{aligned}$$

Gaussian Mixture-Model

Data: $\mathcal{D} = \{\mathbf{x}^{(i)}\}_{i=1}^{N}$ where $\mathbf{x}^{(i)} \in \mathbb{R}^M$

Generative Story:

$z \sim \text{Categorical}(\phi)$

$\mathbf{x} \sim \text{Gaussian}(\boldsymbol{\mu}_z, \boldsymbol{\Sigma}_z)$

Model:

Joint: $p(\mathbf{x}, z; \boldsymbol{\phi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = p(\mathbf{x}|z; \boldsymbol{\mu}, \boldsymbol{\Sigma}) p(z; \boldsymbol{\phi})$

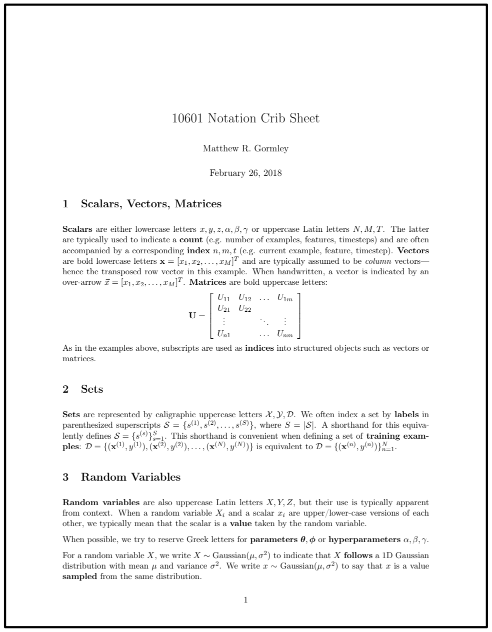
Marginal: $p(\mathbf{x}; \boldsymbol{\phi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{z=1}^{K} p(\mathbf{x}|z; \boldsymbol{\mu}, \boldsymbol{\Sigma}) p(z; \boldsymbol{\phi})$

(Marginal) Log-likelihood:

$$\begin{aligned} \ell(\boldsymbol{\phi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) &= \log \prod_{i=1}^{N} p(\mathbf{x}^{(i)}; \boldsymbol{\phi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) \\ &= \sum_{i=1}^{N} \log \sum_{z=1}^{K} p(\mathbf{x}^{(i)}|z; \boldsymbol{\mu}, \boldsymbol{\Sigma}) p(z; \boldsymbol{\phi}) \end{aligned}$$

Notation for ML

Machine Learning is notorious for requiring lots of notation…and not always being terribly consistent about it!



10601 Notation Crib Sheet

Matthew R. Gormley

February 26, 2018

1 Scalars, Vectors, Matrices

Scalars are either lowercase letters $x, y, z, \alpha, \beta, \gamma$ or uppercase Latin letters N, M, T. The latter are typically used to indicate a **count** (e.g. number of examples, features, timesteps) and are often accompanied by a corresponding **index** n, m, t (e.g. current example, feature, timestep). **Vectors** are bold lowercase letters $\mathbf{x} = [x_1, x_2, \ldots, x_M]^T$ and are typically assumed to be *column* vectors—hence the transposed row vector in this example. When handwritten, a vector is indicated by an over-arrow $\vec{x} = [x_1, x_2, \ldots, x_M]^T$. **Matrices** are bold uppercase letters:

$$\mathbf{U} = \begin{bmatrix} U_{11} & U_{12} & \ldots & U_{1m} \\ U_{21} & U_{22} & & \\ \vdots & & \ddots & \vdots \\ U_{n1} & & \ldots & U_{nm} \end{bmatrix}$$

As in the examples above, subscripts are used as **indices** into structured objects such as vectors or matrices.

2 Sets

Sets are represented by caligraphic uppercase letters $\mathcal{X}, \mathcal{Y}, \mathcal{D}$. We often index a set by **labels** in parenthesized superscripts $\mathcal{S} = \{s^{(1)}, s^{(2)}, \ldots, s^{(S)}\}$, where $S = |\mathcal{S}|$. A shorthand for this equivalently defines $\mathcal{S} = \{s^{(s)}\}_{s=1}^S$. This shorthand is convenient when defining a set of **training examples**: $\mathcal{D} = \{(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), \ldots, (\mathbf{x}^{(N)}, y^{(N)})\}$ is equivalent to $\mathcal{D} = \{(\mathbf{x}^{(n)}, y^{(n)})\}_{n=1}^N$.

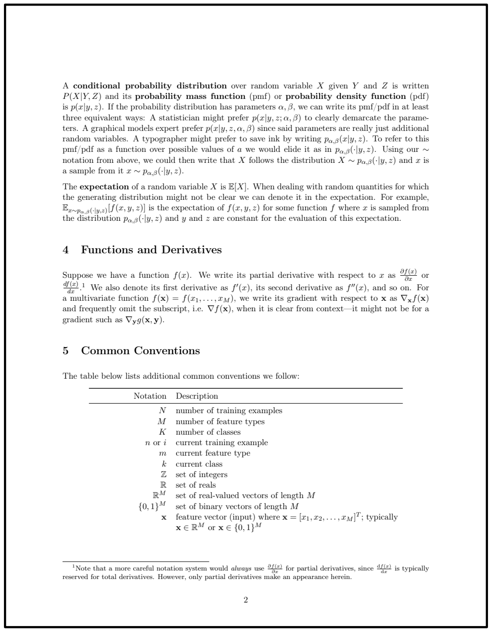
3 Random Variables

Random variables are also uppercase Latin letters X, Y, Z, but their use is typically apparent from context. When a random variable X_i and a scalar x_i are upper/lower-case versions of each other, we typically mean that the scalar is a **value** taken by the random variable.

When possible, we try to reserve Greek letters for **parameters** $\boldsymbol{\theta}, \boldsymbol{\phi}$ or **hyperparameters** α, β, γ.

For a random variable X, we write $X \sim \text{Gaussian}(\mu, \sigma^2)$ to indicate that X **follows** a 1D Gaussian distribution with mean μ and variance σ^2. We write $x \sim \text{Gaussian}(\mu, \sigma^2)$ to say that x is a value **sampled** from the same distribution.

1



A **conditional probability distribution** over random variable X given Y and Z is written $P(X|Y, Z)$ and its **probability mass function** (pmf) or **probability density function** (pdf) is $p(x|y, z)$. If the probability distribution has parameters α, β, we can write its pmf/pdf in at least three equivalent ways: A statistician might prefer $p(x|y, z; \alpha, \beta)$ to clearly demarcate the parameters. A graphical models expert prefer $p(x|y, z, \alpha, \beta)$ since said parameters are really just additional random variables. A typographer might prefer to save ink by writing $p_{\alpha,\beta}(x|y, z)$. To refer to this pmf/pdf as a function over possible values of a we would elide it as in $p_{\alpha,\beta}(\cdot|y, z)$. Using our $\sim$ notation from above, we could then write that X follows the distribution $X \sim p_{\alpha,\beta}(\cdot|y, z)$ and x is a sample from it $x \sim p_{\alpha,\beta}(\cdot|y, z)$.

The **expectation** of a random variable X is $\mathbb{E}[X]$. When dealing with random quantities for which the generating distribution might not be clear we can denote it in the expectation. For example, $\mathbb{E}_{x \sim p_{\alpha,\beta}(\cdot|y,z)}[f(x, y, z)]$ is the expectation of $f(x, y, z)$ for some function f where x is sampled from the distribution $p_{\alpha,\beta}(\cdot|y, z)$ and y and z are constant for the evaluation of this expectation.

4 Functions and Derivatives

Suppose we have a function $f(x)$. We write its partial derivative with respect to x as $\frac{\partial f(x)}{\partial x}$ or $\frac{df(x)}{dx}$.[1] We also denote its first derivative as $f'(x)$, its second derivative as $f''(x)$, and so on. For a multivariate function $f(\mathbf{x}) = f(x_1, \ldots, x_M)$, we write its gradient with respect to $\mathbf{x}$ as $\nabla_{\mathbf{x}} f(\mathbf{x})$ and frequently omit the subscript, i.e. $\nabla f(\mathbf{x})$, when it is clear from context—it might not be for a gradient such as $\nabla_{\mathbf{y}} g(\mathbf{x}, \mathbf{y})$.

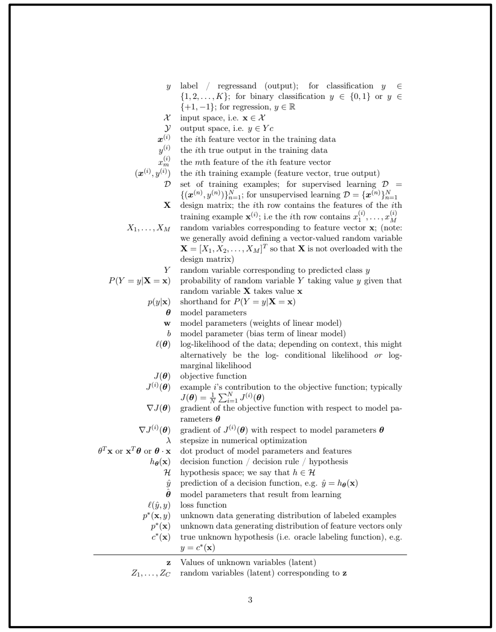
5 Common Conventions

The table below lists additional common conventions we follow:

Notation	Description
N	number of training examples
M	number of feature types
K	number of classes
n or i	current training example
m	current feature type
k	current class
$\mathbb{Z}$	set of integers
$\mathbb{R}$	set of reals
$\mathbb{R}^M$	set of real-valued vectors of length M
$\{0,1\}^M$	set of binary vectors of length M
$\mathbf{x}$	feature vector (input) where $\mathbf{x} = [x_1, x_2, \ldots, x_M]^T$; typically $\mathbf{x} \in \mathbb{R}^M$ or $\mathbf{x} \in \{0,1\}^M$

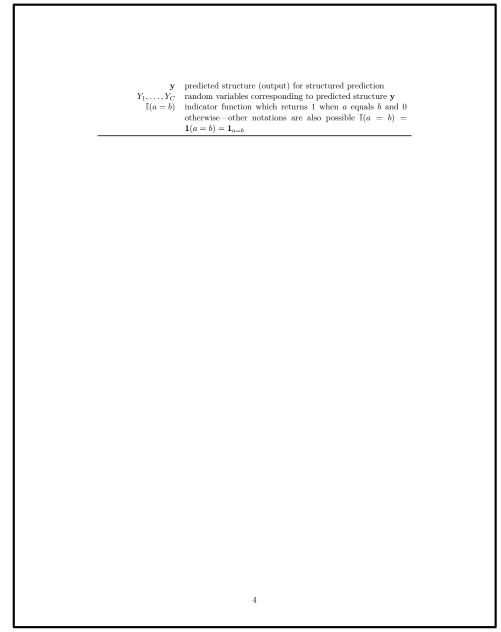
[1] Note that a more careful notation system would *always* use $\frac{\partial f(x)}{\partial x}$ for partial derivatives, since $\frac{\mathrm{d} f(x)}{\mathrm{d} x}$ is typically reserved for total derivatives. However, only partial derivatives make an appearance herein.

2



y	label / regressand (output); for classification $y \in \{1, 2, \ldots, K\}$; for binary classification $y \in \{0, 1\}$ or $y \in \{+1, -1\}$; for regression, $y \in \mathbb{R}$
$\mathcal{X}$	input space, i.e. $\mathbf{x} \in \mathcal{X}$
$\mathcal{Y}$	output space, i.e. $y \in Yc$
$\boldsymbol{x}^{(i)}$	the ith feature vector in the training data
$y^{(i)}$	the ith true output in the training data
$x_m^{(i)}$	the mth feature of the ith feature vector
$(\boldsymbol{x}^{(i)}, y^{(i)})$	the ith training example (feature vector, true output)
$\mathcal{D}$	set of training examples; for supervised learning $\mathcal{D} = \{(\boldsymbol{x}^{(n)}, y^{(n)})\}_{n=1}^N$; for unsupervised learning $\mathcal{D} = \{\boldsymbol{x}^{(n)}\}_{n=1}^N$
$\mathbf{X}$	design matrix; the ith row contains the features of the ith training example $\mathbf{x}^{(i)}$; i.e the ith row contains $x_1^{(i)}, \ldots, x_M^{(i)}$
$X_1, \ldots, X_M$	random variables corresponding to feature vector $\mathbf{x}$; (note: we generally avoid defining a vector-valued random variable $\mathbf{X} = [X_1, X_2, \ldots, X_M]^T$ so that $\mathbf{X}$ is not overloaded with the design matrix)
Y	random variable corresponding to predicted class y
$P(Y = y \mid \mathbf{X} = \mathbf{x})$	probability of random variable Y taking value y given that random variable $\mathbf{X}$ takes value $\mathbf{x}$
$p(y \mid \mathbf{x})$	shorthand for $P(Y = y \mid \mathbf{X} = \mathbf{x})$
$\boldsymbol{\theta}$	model parameters
$\mathbf{w}$	model parameters (weights of linear model)
b	model parameter (bias term of linear model)
$\ell(\boldsymbol{\theta})$	log-likelihood of the data; depending on context, this might alternatively be the log- conditional likelihood *or* log-marginal likelihood
$J(\boldsymbol{\theta})$	objective function
$J^{(i)}(\boldsymbol{\theta})$	example i's contribution to the objective function; typically $J(\boldsymbol{\theta}) = \frac{1}{N}\sum_{i=1}^N J^{(i)}(\boldsymbol{\theta})$
$\nabla J(\boldsymbol{\theta})$	gradient of the objective function with respect to model parameters $\boldsymbol{\theta}$
$\nabla J^{(i)}(\boldsymbol{\theta})$	gradient of $J^{(i)}(\boldsymbol{\theta})$ with respect to model parameters $\boldsymbol{\theta}$
λ	stepsize in numerical optimization
$\theta^T\mathbf{x}$ or $\mathbf{x}^T\boldsymbol{\theta}$ or $\boldsymbol{\theta} \cdot \mathbf{x}$	dot product of model parameters and features
$h_{\boldsymbol{\theta}}(\mathbf{x})$	decision function / decision rule / hypothesis
$\mathcal{H}$	hypothesis space; we say that $h \in \mathcal{H}$
$\hat{y}$	prediction of a decision function, e.g. $\hat{y} = h_{\boldsymbol{\theta}}(\mathbf{x})$
$\hat{\boldsymbol{\theta}}$	model parameters that result from learning
$\ell(\hat{y}, y)$	loss function
$p^*(\mathbf{x}, y)$	unknown data generating distribution of labeled examples
$p^*(\mathbf{x})$	unknown data generating distribution of feature vectors only
$c^*(\mathbf{x})$	true unknown hypothesis (i.e. oracle labeling function), e.g. $y = c^*(\mathbf{x})$
$\mathbf{z}$	Values of unknown variables (latent)
$Z_1, \ldots, Z_C$	random variables (latent) corresponding to $\mathbf{z}$

3



$\mathbf{y}$	predicted structure (output) for structured prediction
$Y_1, \ldots, Y_C$	random variables corresponding to predicted structure $\mathbf{y}$
$\mathbb{I}(a = b)$	indicator function which returns 1 when a equals b and 0 otherwise—other notations are also possible $\mathbb{I}(a = b) = \mathbf{1}(a = b) = \mathbf{1}_{a=b}$

4

PRELIMINARIES: SETS AND TYPES

Sets

Chalkboard

- Definitions: Set, element of, equality, subset
- Example: Sets of sets
- Set builder notation
- Python list/set comprehentions
- Exercise: Set builder notation
- Definitions: Union, intersection, difference, complement
- Exercise: Set complement
- Tuples and set product
- Exercise: Set product

Data Types and Functions

Chalkboard

- Data types, structs, unions
- Tagged unions
- Exercise: Tagged unions
- Functions
- Anonymous functions
- Exercises: Functions