Lecture 1. StatML Welcome and Maths Review

COMP90051 Statistical Machine Learning

Sem1 2021

Lecturer: Trevor Cohn

THE UNIVERSITY OF MELBOURNE

This lecture

- About COMP90051
- Review: Probability theory
- Review: Linear algebra
- Review: Sequences and limits

Subject objectives

- Develop an appreciation for the role of statistical ML, advanced foundations and applications
- Gain an understanding of a representative selection of ML techniques – *how ML works*
- Be able to design, implement and evaluate ML systems
- Become a discerning ML consumer

Subject content

- The subject will cover topics from

  Foundations of statistical learning, linear models, non-linear bases, regularised linear regression, generalisation theory, kernel methods, deep neural nets, multi-armed bandits, Bayesian learning, probabilistic models

- Theory in lectures; hands-on experience with range of toolkits in workshop pracs and projects
- vs COMP90049: much depth, much rigor, so wow

Subject staff / Contact hours

Contacting staff	*Discussion board first; then combined staff email* **comp90051-2021s1-staff@lists.unimelb.edu.au**
Lecturer & Coordinator	Trevor Cohn Professor, Computing & Information Systems *Statistical Machine Learning, Natural Language Processing*
Tutors:	Justin Tan, Jun Wang, Xudong Han, Kazi Adnan. *See Canvas for latest list and contact details.*
Contact:	*Weekly, please attend: 2nd Lecture (live Zoom discussion), 1 Workshop (some live, most Zoom)*
Pre-recorded Lectures:	*Posted to Canvas for you to view safely at home.* Strongly recommend that you keep up, weekly. (viz. quizzes)

COVIDSafe Campus

Stay COVIDSafe on campus

If you haven't already, you must complete the COVIDSafe module and health declaration immediately:
students.unimelb.edu.au/COVIDSafe

WHAT WE DO NOW BECOMES WHAT HAPPENS NEXT

Completing this module is a requirement for being on campus, and you must follow all COVIDSafe guidelines while you're here.

THE UNIVERSITY OF MELBOURNE

unimelb.edu.au/coronavirus

V2 Feb 2021

COVIDSafe Campus

Stay COVIDSafe on campus

If you start to feel unwell while on campus, leave immediately.
Call the University Health Services COVID-19 Hotline at 8344 6905. You may be able to get a COVID-19 test before going home.

We encourage everyone to scan QR codes
in all areas you visit to assist with contact tracing if needed.

1.5m

Where possible, keep at least 1.5 metres between yourself and others.

Remember to sneeze or cough into a tissue or your elbow, followed by hand hygiene.

Wash or sanitise your hands often.

Carry a face mask at all times and wear one as required.
Guidelines are changing often, so please check the latest requirements.

Leave classrooms and surrounding areas promptly.
This assists in maintaining physical distancing during changeover periods between classes.

WHAT WE DO NOW BECOMES WHAT HAPPENS NEXT

STAY SAFE. KEEP EVERYONE SAFE.

COVID-19 hotline: 8344 6905 | services.unimelb.edu.au/health

Stay informed about the latest COVID-19 public health advice, cases and exposure sites at
www.coronavirus.vic.gov.au/coronavirus-covid-19-victoria

THE UNIVERSITY OF MELBOURNE

unimelb.edu.au/coronavirus

V2 Feb 2021

About me (Trevor)

- PhD 2006 – Melbourne
- Several years in research
  - UK: Edinburgh U, Sheffield U.
  - Australia: Melbourne U.
- **Interests**: Structured prediction; graphical models; probabilistic modelling (Bayesian); deep learning; transfer learning
- **Applications to language**: e.g., translation, structure parsing/induction, sequential tagging

Advanced ML: Expected Background

- Why a challenge: Diverse math + CS + coding
- ML: COMP90049 either 2020s1 “new” or earlier (we’ll review gaps throughout semester)
- Alg & complexity: big-oh, termination; basic data structures & algorithms; solid coding ideally experience in Python

…and more…

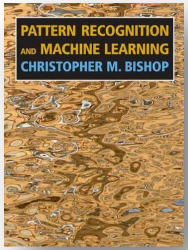
Advanced ML: Expected Background

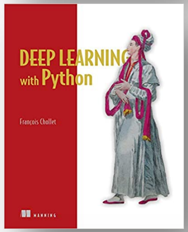
...and more...

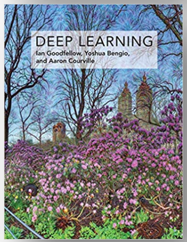
- Maths: Review next videos, but ideally seen most before
  "Matrix **A** *is symmetric & positive definite, hence its eigenvalues..."*
- Probability theory: probability calculus; discrete/continuous distributions; multivariate; exponential families; Bayes rule
- Sequences: sequences, limits, supremum
- Linear algebra: vector inner products & norms; orthonormal bases; matrix operations, inverses, eigenvectors/values
- Calculus & optimisation: partial derivatives; gradient descent; convexity; Lagrange multipliers

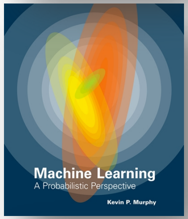
Textbooks

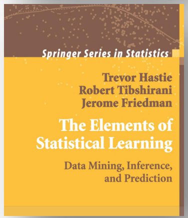
- We don’t have only one reference. We prefer to pick good bits from several. We may also supplement with other readings as we go.
- All are available free online or through the library digitally. See the Canvas lecture outline for links. Therefore, no need to buy.
- Primarily we refer to (good all rounder): Bishop (2007) *Pattern Recognition and Machine Learning*
- Practical Deep Nets: Chollet (2017) *Deep learning with Python*
- More deep learning detail: Goodfellow, Bengio, Courville (2016) *Deep learning*
- For more on PGMs/Bayesian inference: Murphy (2012) *Machine Learning: A Probabilistic Perspective*
- For reference on frequentist ideas, SVMs, lasso, etc.: Hastie, Tibshirani, Friedman (2001) *The Elements of Statistical Learning: Data Mining, Inference and Prediction*











Assessment

- Assessment components
  - Two projects – one group (w4-7), one individual (w9-11)
    - Each (25%)
    - Each has ~3 weeks to complete
  - Regular Quiz, ~fortnightly (5 x 2% = 10%)
  - Final Exam (40%)

- 50% hurdles applied to
  both **exam**, and **combined projects** ~~+ quiz~~

*updated 1/3 to reflect handbook; namely hurdle only applied to projects, not the quiz results. Note there are **two** separate hurdles.*

Machine Learning Basics

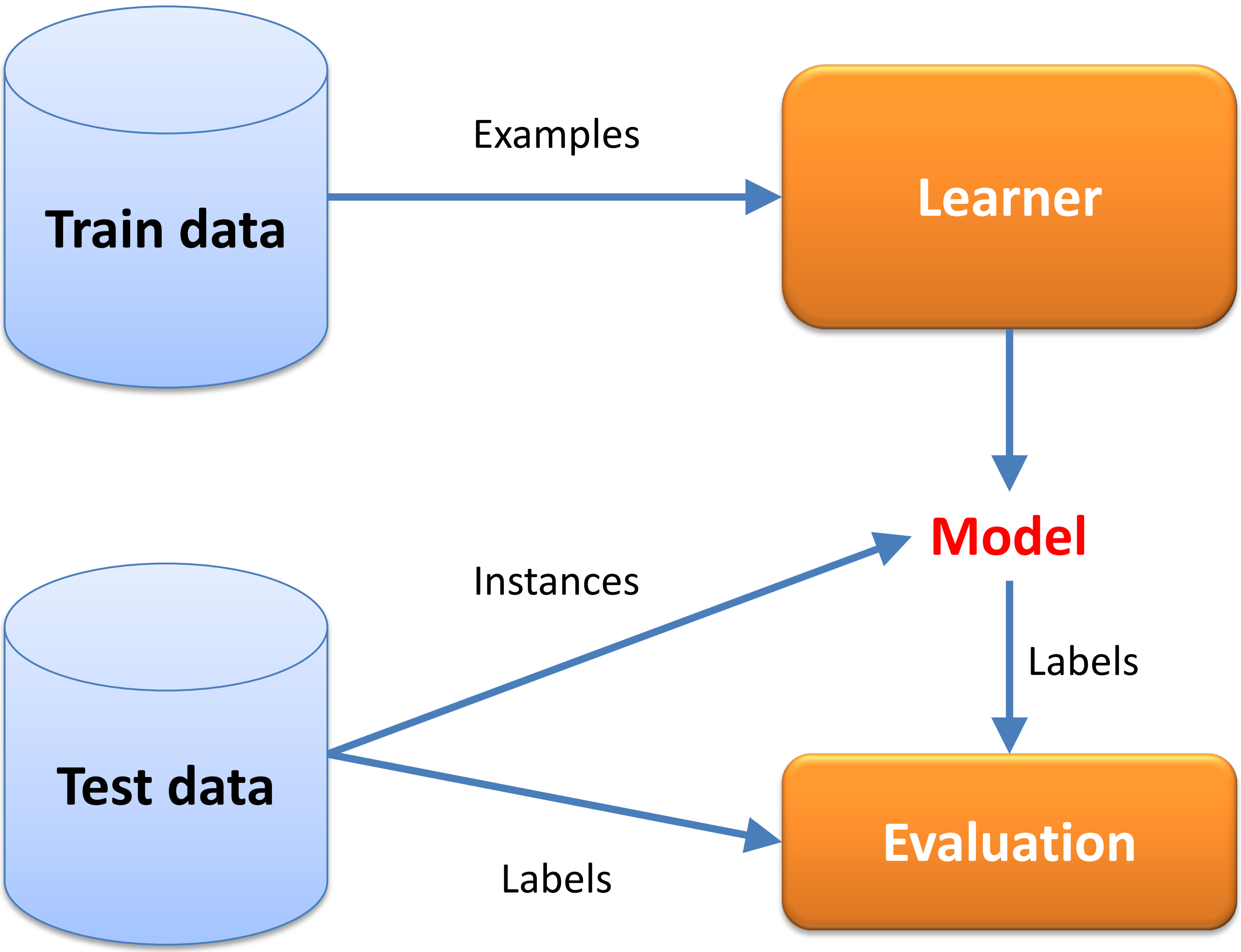
Terminology

- Input to a machine learning system can consist of
  - Instance: measurements about individual entities/objects
    a loan application
  - Attribute (aka Feature, explanatory var.): component of the instances
    the applicant's salary, number of dependents, etc.
  - Label (aka Response, dependent var.): an outcome that is categorical, numeric, etc.
    forfeit vs. paid off
  - Examples: instance coupled with label
    <(100k, 3), "forfeit">
  - Models: discovered relationship between attributes and/or label

Supervised vs unsupervised learning

	Data	Model used for
Supervised learning	Labelled	Predict labels on new instances
Unsupervised learning	Unlabelled	Cluster related instances; Project to fewer dimensions; Understand attribute relationships

Architecture of a supervised learner

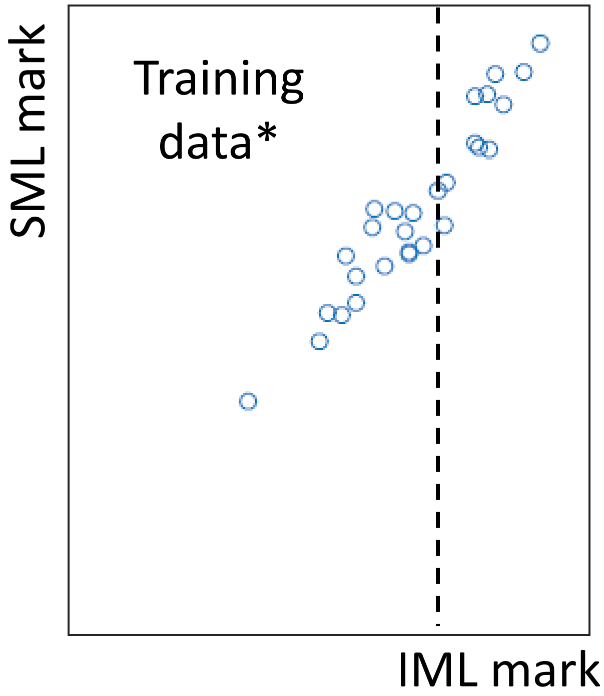


Evaluation (supervised learners)

- How you measure quality depends on your problem!
- Typical process
  - Pick an evaluation metric comparing label vs prediction
  - Procure an independent, labelled test set
  - "Average" the evaluation metric over the test set
- Example evaluation metrics
  - Accuracy, Contingency table, Precision-Recall, ROC curves
- When data poor, cross-validate

Probability theory

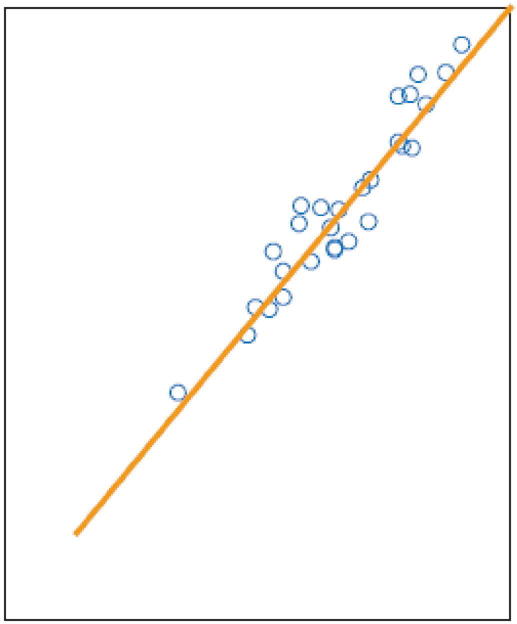
Data is noisy (almost always)



* Example:
  * given mark for Intro ML (IML)
  * predict mark for Stat Machine Learning (SML)

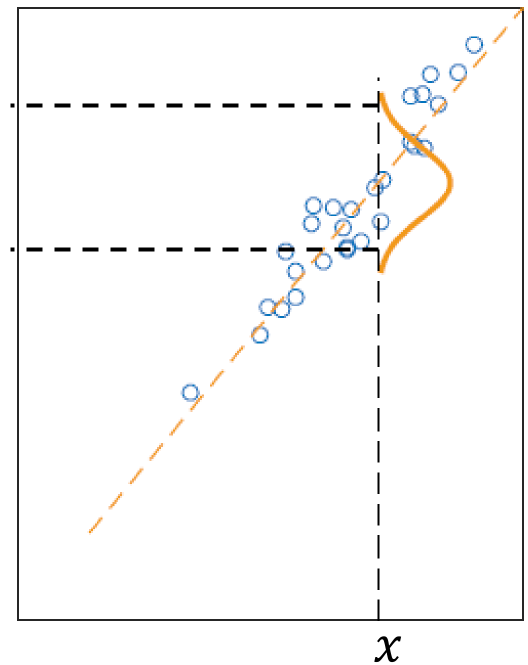
* synthetic data :)

Types of models



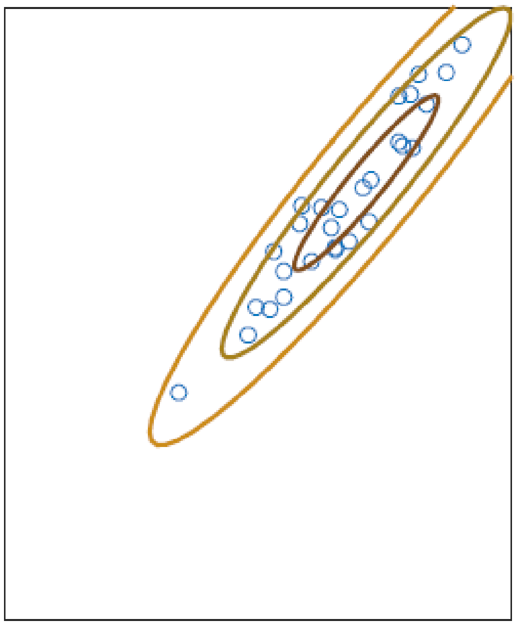
$$\hat{y} = f(x)$$

IntroML mark was 95, SML mark is predicted to be 95



$$P(y|x)$$

IntroML mark was 95, SML mark is likely to be in (92, 97)



$$P(x, y)$$

probability of having $(IML = x, SML = y)$

Basics of probability theory

- A probability space:
  - Set Ω of possible outcomes
  - Set F of events (subsets of outcomes)
  - Probability measure P: $F \rightarrow \mathbf{R}$
- Example: a die roll
  - {1, 2, 3, 4, 5, 6}
  - { φ, {1}, …, {6}, {1,2}, …, {5,6}, …, {1,2,3,4,5,6} }
  - P(φ)=0, P({1})=1/6, P({1,2})=1/3, …

Axioms of probability*

1. F contains all of: Ω; all complements $\Omega \backslash f$, $f \in F$; the union of any countable set of events in F.
2. $P(f) \geq 0$ for every event $f \in F$.
3. $P\left(\bigcup_f f\right) = \sum_f P(f)$ for all countable sets of pairwise disjoint events.
4. $P(\Omega) = 1$

* We won't delve further into advanced probability theory, which starts with measure theory – a beautiful subject and the only way to "fully" formulate probability.

Random variables (**r.v.'s**)

- A random variable *X* is a numeric function of outcome $X(\omega) \in \boldsymbol{R}$
- $P(X \in A)$ denotes the probability of the outcome being such that *X* falls in the range *A*
- Example: *X* winnings on $5 bet on even die roll
  - *X* maps 1,3,5 to -5
    X maps 2,4,6 to 5
  - P(*X*=5) = P(*X*=-5) = ½

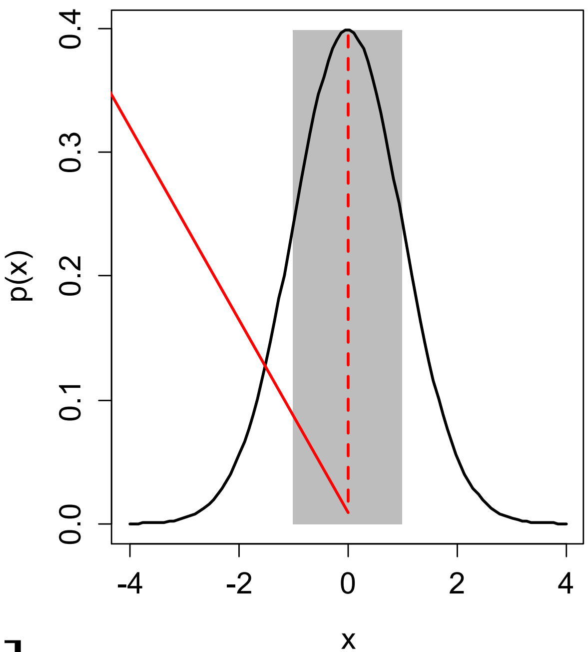
Discrete vs. continuous distributions

- Discrete distributions
  - Govern r.v. taking discrete values
  - Described by probability mass function p(x) which is P(X=x)
  - $P(X \leq x) = \sum_{a=-\infty}^{x} p(a)$
  - **Examples**: Bernoulli, Binomial, Multinomial, Poisson

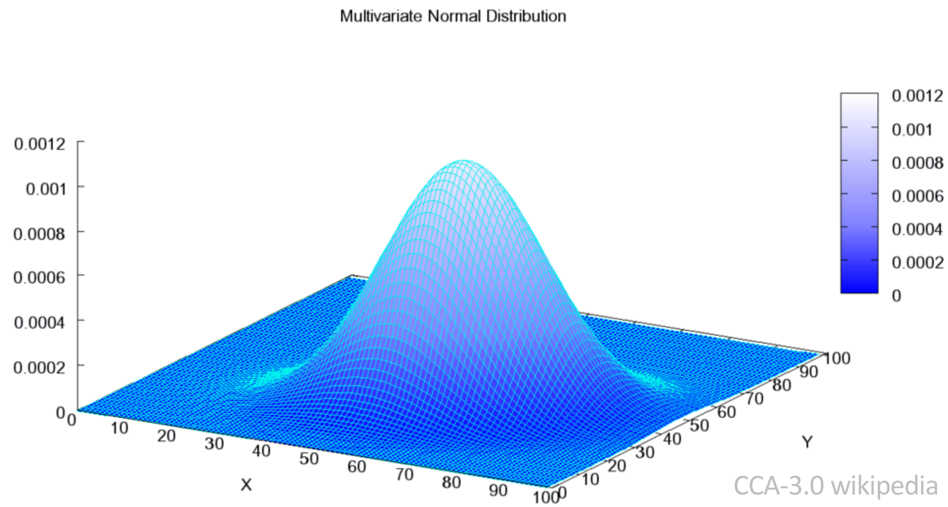
- Continuous distributions
  - Govern real-valued r.v.
  - Cannot talk about PMF but rather probability density function p(x)
  - $P(X \leq x) = \int_{-\infty}^{x} p(a)da$
  - **Examples**: Uniform, Normal, Laplace, Gamma, Beta, Dirichlet

Expectation

- Expectation $E[X]$ is the r.v. X's "average" value
  - Discrete: $E[X] = \sum_x x\, P(X = x)$
  - Continuous: $E[X] = \int_x x\, p(x)\, dx$
- Properties
  - Linear: $E[aX + b] = aE[X] + b$
    $E[X + Y] = E[X] + E[Y]$
  - Monotone: $X \geq Y \Rightarrow E[X] \geq E[Y]$
- Variance: $Var(X) = E[(X - E[X])^2]$



Multivariate distributions



- Specify joint distribution over multiple variables
- Probabilities are computed as in univariate case, we now just have repeated summations or repeated integrals
- Discrete: $P(X, Y \in A) = \sum_{(x,y) \in A} p(x, y)$
- Continuous: $P(X, Y \in A) = \int_A \; p(x, y) dx dy$

Independence and conditioning

- *X*, *Y* are independent if
  - $P(X \in A, Y \in B) = P(X \in A)P(Y \in B)$
  - Similarly for densities: $p_{X,Y}(x, y) = p_X(x)p_Y(y)$
  - **Intuitively**: knowing value of *Y* reveals nothing about *X*
  - **Algebraically**: the joint on *X*,*Y* factorises!
- Conditional probability
  - $P(A|B) = \frac{P(A \cap B)}{P(B)}$
  - Similarly for densities $p(y|x) = \frac{p(x,y)}{p(x)}$
  - **Intuitively**: probability event *A* will occur given we know event *B* has occurred
  - X,Y independent equiv to $P(Y = y|X = x) = P(Y = y)$

Inverting conditioning: Bayes' Theorem

Bayes

- In terms of events *A*, *B*
  - $P(A \cap B) = P(A|B)P(B) = P(B|A)P(A)$
  - $P(A|B) = \dfrac{P(B|A)\,P(A)}{P(B)}$
- Simple rule that lets us swap conditioning order
- Probabilistic and Bayesian inference make heavy use
  - Marginals: probabilities of individual variables
  - Marginalisation: summing away all but r.v.'s of interest

$$P(A) = \sum_b P(A, B = b)$$

Mini Summary

- Probability spaces, axioms of probability
- Discrete vs continuous; Univariate vs multivariate
- Expectation, Variance
- Independence and conditioning
- Bayes rule and marginalisation

Next: Linear algebra primer/review

Vectors

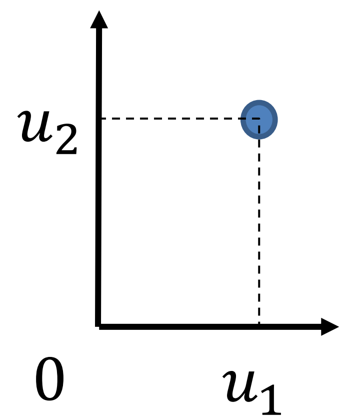
Link between geometric and algebraic interpretation of ML methods

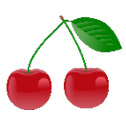
What are vectors?

Suppose $\boldsymbol{u} = [u_1, u_2]'$. What does $\boldsymbol{u}$ really represent?

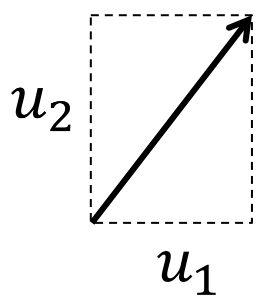
Ordered set of numbers $\{u_1, u_2\}$

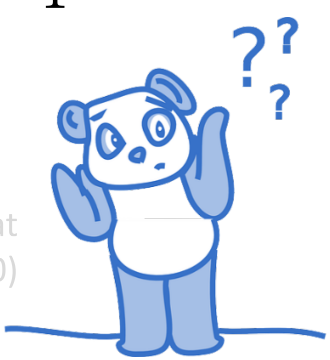
Cartesian coordinates of a point





A direction





art: OpenClipartVectors at pixabay.com (CC0)

Dot product: Algebraic definition

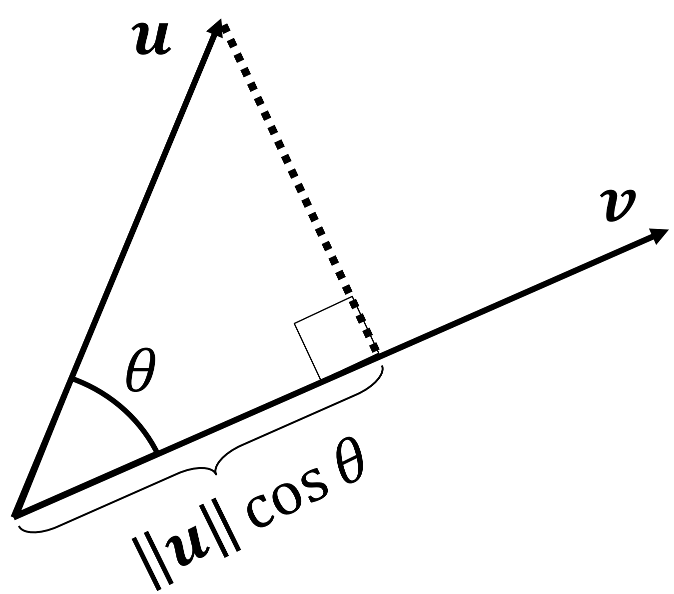
- Given two m-dimensional vectors $\boldsymbol{u}$ and $\boldsymbol{v}$, their dot product is $\boldsymbol{u} \cdot \boldsymbol{v} \equiv \boldsymbol{u}'\boldsymbol{v} \equiv \sum_{i=1}^{m} u_i v_i$
  - E.g., weighted sum of terms is a dot product $\boldsymbol{x}'\boldsymbol{w}$
- If k is a scalar, $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}$ are vectors then

$$(k\boldsymbol{a})'\boldsymbol{b} = k(\boldsymbol{a}'\boldsymbol{b}) = \boldsymbol{a}'(k\boldsymbol{b})$$

$$\boldsymbol{a}'(\boldsymbol{b} + \boldsymbol{c}) = \boldsymbol{a}'\boldsymbol{b} + \boldsymbol{a}'\boldsymbol{c}$$

Dot product: Geometric definition

- Given two m-dimensional Euclidean vectors $\boldsymbol{u}$ and $\boldsymbol{v}$, their dot product is $\boldsymbol{u} \cdot \boldsymbol{v} \equiv \boldsymbol{u}'\boldsymbol{v} \equiv \|\boldsymbol{u}\|\|\boldsymbol{v}\| \cos\theta$
  - $\|\boldsymbol{u}\|, \|\boldsymbol{v}\|$ are L_2 *norms* for $\boldsymbol{u}, \boldsymbol{v}$ also written as $\|\boldsymbol{u}\|_{\mathbf{2}}$
  - θ is the angle between the vectors



The *scalar projection* of $\boldsymbol{u}$ onto $\boldsymbol{v}$ is given by

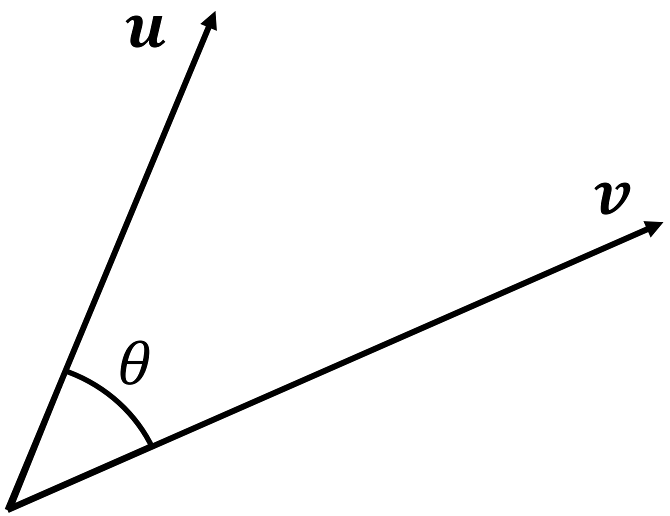
$$u_{\boldsymbol{v}} = \|\boldsymbol{u}\| \cos\theta$$

Thus dot product is

$$\boldsymbol{u}'\boldsymbol{v} = u_{\boldsymbol{v}}\|\boldsymbol{v}\| = v_{\boldsymbol{u}}\|\boldsymbol{u}\|$$

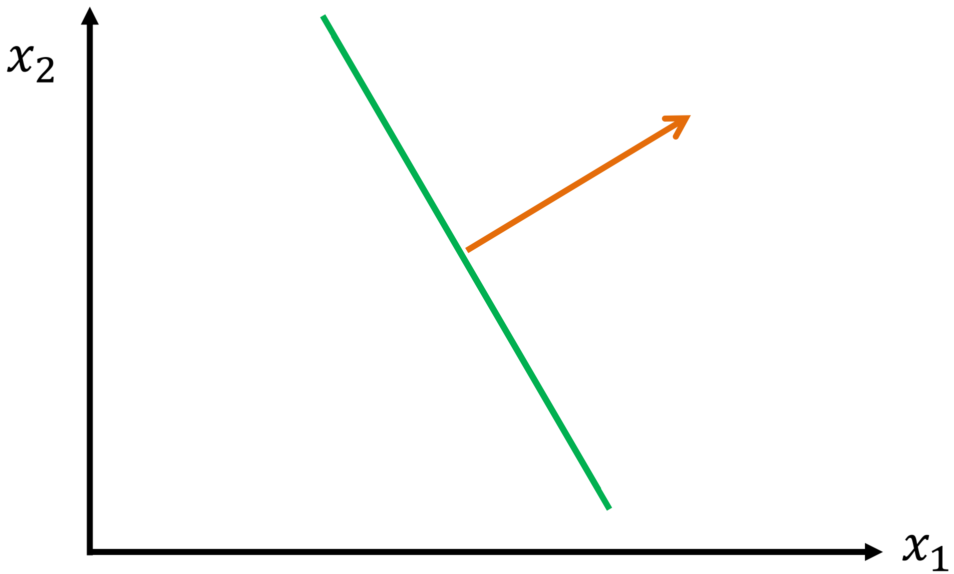
Geometric properties of the dot product

- If the two vectors are orthogonal then $\boldsymbol{u}'\boldsymbol{v} = 0$
- If the two vectors are parallel then $\boldsymbol{u}'\boldsymbol{v} = \|\boldsymbol{u}\|\|\boldsymbol{v}\|$, if they are anti-parallel then $\boldsymbol{u}'\boldsymbol{v} = -\|\boldsymbol{u}\|\|\boldsymbol{v}\|$
- $\boldsymbol{u}'\boldsymbol{u} = \|\boldsymbol{u}\|^2$, so $\|\boldsymbol{u}\| = \sqrt{u_1^2 + \cdots + u_m^2}$ defines the Euclidean vector length



Hyperplanes and normal vectors

- A ***hyperplane*** defined by parameters $\boldsymbol{w}$ and b is a set of points $\boldsymbol{x}$ that satisfy $\boldsymbol{x}'\boldsymbol{w} + b = 0$
- In 2D, a hyperplane is a line: a line is a set of points that satisfy $w_1 x_1 + w_2 x_2 + b = 0$



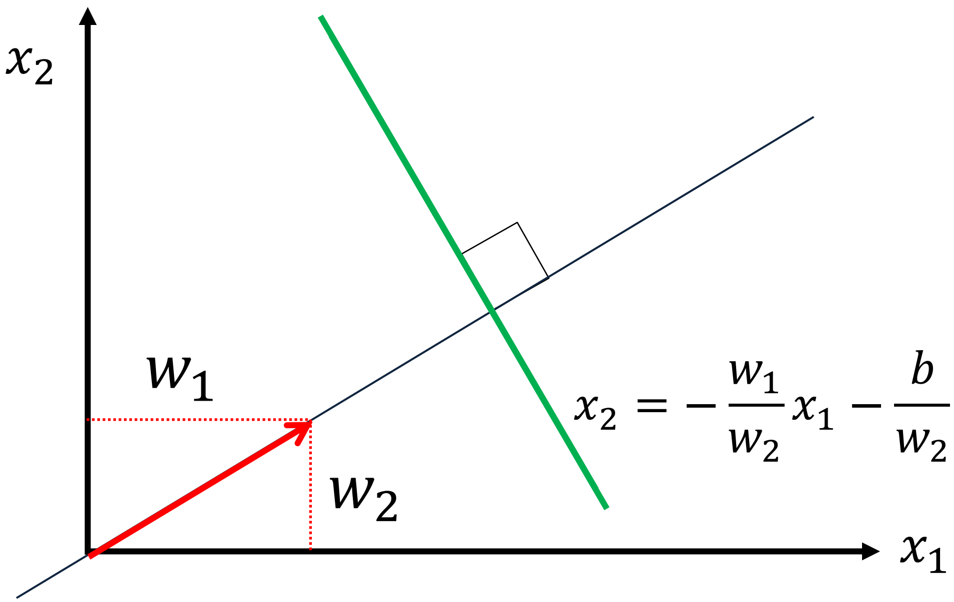
A *normal vector* for a hyperplane is a vector perpendicular to that hyperplane

Hyperplanes and normal vectors

- Consider a hyperplane defined by parameters $\boldsymbol{w}$ and b. Note that $\boldsymbol{w}$ is itself a vector
- Lemma: Vector $\boldsymbol{w}$ is normal to the hyperplane
- Proof sketch:
  * Choose any two points $\boldsymbol{u}$ and $\boldsymbol{v}$ on the hyperplane. Note that vector $(\boldsymbol{u}-\boldsymbol{v})$ lies on the hyperplane
  * Consider dot product $(\boldsymbol{u}-\boldsymbol{v})'\boldsymbol{w} = \boldsymbol{u}'\boldsymbol{w} - \boldsymbol{v}'\boldsymbol{w}$
    $$= (\boldsymbol{u}'\boldsymbol{w} + b) - (\boldsymbol{v}'\boldsymbol{w} + b) = 0$$
  * Thus $(\boldsymbol{u}-\boldsymbol{v})$ lies on the hyperplane, but is perpendicular to $\boldsymbol{w}$, and so $\boldsymbol{w}$ is a vector normal

Example in 2D

- Consider a line defined by w_1, w_2 and b
- Vector $\boldsymbol{w} = [w_1, w_2]'$ is a normal vector



L_1 and L_2 norms

- Throughout the subject we will often encounter norms that are functions $\mathbb{R}^n \to \mathbb{R}$ of a particular form
  * Intuitively, norms measure lengths of vectors in some sense
  * Often component of objectives or stopping criteria in optimisation-for-ML
- More specifically, we will often use the L_2 norm (*aka* Euclidean distance)

$$\|\boldsymbol{a}\| = \|\boldsymbol{a}\|_2 \equiv \sqrt{a_1^2 + \cdots + a_n^2}$$

- And also the L_1 norm (*aka* absolute norm or Manhattan distance)

$$\|\boldsymbol{a}\|_1 \equiv |a_1| + \cdots + |a_n|$$

Vector Spaces and Bases

Useful in interpreting matrices and some algorithms like PCA

Linear combinations, Independence

- For formal definition of vector spaces:
  https://en.wikipedia.org/wiki/Vector_space#Definition
- A linear combination of vectors $v_1, \dots, v_k \in V$ some vector space, is a new vector $\sum_{i=1}^{k} a_i v_i$ for some scalars $a_1, \dots, a_k$
- A set of vectors $\{v_1, \dots, v_k\} \subseteq V$ is called linearly dependent if one element v_j can be written as a linear combination of the other elements
- A set that isn't linearly dependent is linearly independent

Spans, Bases

- The span of vectors $v_1, \ldots, v_k \in V$ is the set of all obtainable linear combinations (ranging over all scalar coefficients) of the vectors
- A set of vectors $\{v_1, \ldots, v_k\} \subseteq V$ is called a basis for a vector subspace $V' \subseteq V$ if
  1. The set is linearly independent; and
  2. Every $\mathrm{v} \in V'$ is a linear combination of the set.
- An orthonormal basis is a basis in which each
  1. Pair of basis vectors are orthogonal (zero dot prod); and
  2. Basis vector has norm equal to 1.

Matrices

Some useful facts for ML

Basic matrices

- See more: https://en.wikipedia.org/wiki/Matrix_(mathematics)
  - Including matrix-matrix and matrix-vector products
- A rectangular array, often denoted by upper-case, with two indices first for row, second for column
- Square matrix has equal dimensions (numbers of rows and columns)
- Matrix transpose $\mathbf{A}'$ or $\mathbf{A}^T$ of m by n matrix $\mathbf{A}$ is an n by m matrix with entries $A'_{ij}=A_{ji}$
- A square matrix $\mathbf{A}$ with $\mathbf{A}=\mathbf{A}'$ is called symmetric
- The (square) identity matrix $\mathbf{I}$ has 1 on the diagonal, 0 off-diagonal
- Matrix inverse $\mathbf{A}^{-1}$ of square matrix $\mathbf{A}$ (if it exists) satisfies $\mathbf{A}^{-1}\mathbf{A}=\mathbf{I}$

Matrix eigenspectrum

- Scalar, vector pair (λ, v) are called an eigenvalue-eigenvector pair of a square matrix **A** if $\mathbf{Av} = \lambda\mathbf{v}$
  - Intuition: matrix **A** doesn't rotate **v** it just stretches it
  - Intuition: the eigenvalue represents stretching factor
- In general eigenvalues may be zero, negative or even complex (imaginary) – we'll only encounter reals

Spectra of common matrices

- Eigenvalues of symmetric matrices are always real (no imaginary component)
- A matrix with linear dependent columns has some zero eigenvalues (called rank deficient) → no matrix inverse exists

Positive (semi)definite matrices

- A symmetric square matrix **A** is called positive semidefinite if for all vectors **v** we have $\mathbf{v}'\mathbf{A}\mathbf{v} \geq 0$.
  - Then **A** has non-negative eigenvalues
  - For example, any $\mathbf{A} = \mathbf{X}'\mathbf{X}$ since: $\mathbf{v}'\mathbf{X}'\mathbf{X}\mathbf{v} = \|\mathbf{X}\mathbf{v}\|^2 \geq 0$
- Further if $\mathbf{v}'\mathbf{A}\mathbf{v} > 0$ holds as a strict inequality then **A** is called positive definite
  - Then **A** has (strictly) positive eigenvalues

Mini Summary

- Vectors: Vector spaces, dot products, independence, hyperplanes
- Matrices: Eigenvalues, positive semidefinite matrices

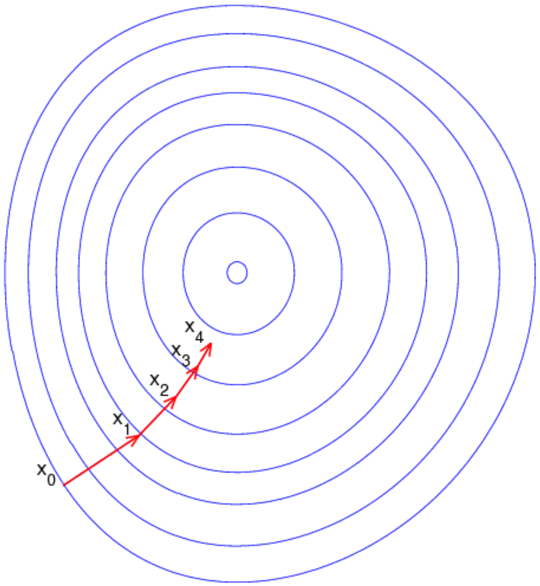
Next: Sequences and limits review/primer

Sequences and Limits

Sequences arise whenever we have iterations (e.g. training loops, growing data sample size). Limits tell us about where sequences tend towards.

Infinite Sequences

- Written like $x_1, x_2, \ldots$ or $\{x_i\}_{i \in \mathbb{N}}$
- Index set: subscript set e.g. $\mathbb{N}$



Wikipedia public domain

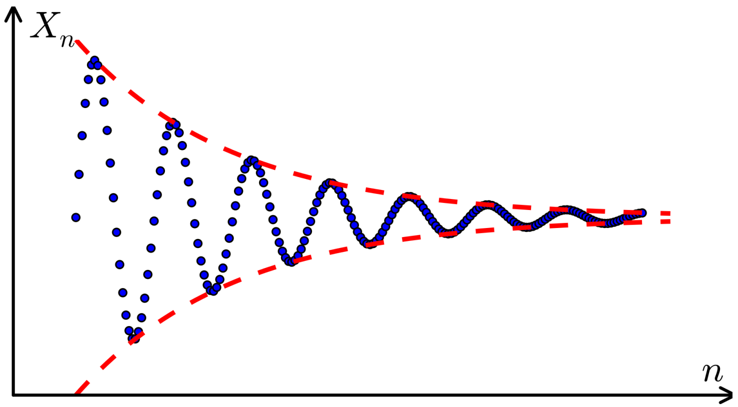
- Sequences allow us to reason about test error when training data grows indefinitely, or training error (or a stopping criterion) when training runs arbitrarily long

Limits and Convergence

- A sequence $\{x_i\}_{i \in \mathbb{N}}$ converges if its elements become and remain arbitrarily close to a fixed limit point L.
- Formally: $x_i \to L$ if, for all $\varepsilon > 0$, there exists $N \in \mathbb{N}$ such that for all $n \geq N$ we have $\|x_n - L\| < \varepsilon$

Notes:

- Epsilon ε represents distance of sequence to limit point
- Distance can be arbitrarily small
- Definition says we eventually get that close (at some finite N) and we stay *at least* that close for ever more



Wikipedia public domain

Supremum

Generalising the maximum: When a sequence never quite peaks.

When does the Maximum Exist?

- Can you always take a max of a set?
- Finite sets: what's the max of {1, 7, 3, 2, 9}?
- Closed, bounded intervals: what's the max of [0,1]?
- Open, bounded intervals: what's the max of [0,1)?
- Open, unbounded intervals: what's the max of [0,∞)?

What about “Least Upper Bound”?

- Can you always take a least-upper-bound of a set? (much more often!)
- Finite sets: what’s the max of {1, 7, 3, 2, 9}?

  max=9 LUB=9

- Closed, bounded intervals: what’s the max of [0,1]?

  max=1 LUB=1

- Open, bounded intervals: what’s the max of [0,1)?

  max=N/A LUB=1

- Open, unbounded intervals: what’s the max of [0,∞)?

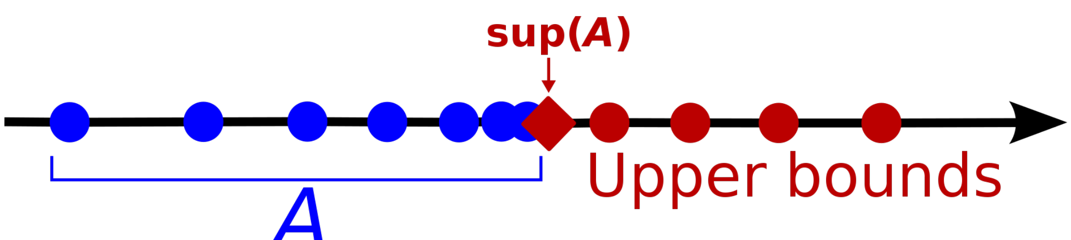
  max=N/A LUB=∞

The Supremum

- Consider any subset S of the reals
- Upper bound $u \in \mathbb{R}^+$ of set S has: $u \geq x$ for all $x \in S$
- If u is no bigger than any other upper bound of S then it's called a least upper bound or supremum of S, written as $\sup(S)$ and pronounced "soup":
  * $z \geq u$ for all upper bounds $z \in \mathbb{R}^+$ of S

FreeSVG public domain

- When we don't know, or can't guarantee, that a set or sequence has a max, it is better to use its sup



Wikipedia public domain

Infimum

- The greatest lower bound or infimum is generalisation of the minimum
- Written inf(S) pronounced “inf”
- Useful if we’re minimising training error but don’t know if the minimum is ever attained.

Mini Summary

- Sequences
- Limits of sequences
- Supremum is the new maximum
- Stochastic convergence

Next time: L02 Statistical schools

Homework week #1: Watch all week 1 recordings.
Jupyter notebooks setup and launch (at home)