Jonathan Mugan, some personal notes and math concepts, without references
March 5, 2015

Contents

1 Questions

1. Why is $\log$ in the definition of entropy?
2. Why do π, $\cos$, and $\sin$ show up so often? Answer, because the circle is the base of Euclidean geometry?
3. What is a geometry?

2 Logarithms and exponents

- $\log_2 y = x$ means that $2^x = y$
- $\log_2 xy = \log_2 x + \log_2 y$.
- $\log_2 \frac{x}{y} = \log_2 x - \log_2 y$. This means that $\log_2 \frac{1}{y} = \log_2 1 - \log_2 y = -\log_2 y$.
- $\log_2 1 = 0$ and $\log_2 y < 0$ where $y < 1$.
- $2^{-x} = \frac{1}{2^x}$ by random definition. Well, dividing is the opposite of multiplying. 3^{-4} means divide 1 by 3 and do it 4 times.
- $3^{\frac{1}{2}} = \sqrt{3}$.

3 Operators

Any function can be called an operator. You could have an addition function and call that an operator. "However, in practice it [the word 'operator'] is most often applied to functions which operate on mathematical entities of higher complexity than real numbers, such as vectors, random variables, or mathematical expressions. The differential and integral operators, for example, have domains and codomains whose elements are mathematical expressions of indefinite complexity." [Wikipedia]

functional A function with a vector-valued domain and a scalar range.

sup Fancy name for "max." The supremum of $f(t)$ over $t \in \mathcal{T}$, denoted $\sup_{t \in \mathcal{T}} f(t)$, is defined as the smallest value x such that $f(t) \leq x$ for all $t \in T$. We say that $\sup_{t \in \mathcal{T}} f(t) = \infty$ if $f(t)$ has arbitrarily large values over $t \in T$.

inf Fancy name for "min." The infimum of $f(t)$ over $t \in \mathcal{T}$, denoted $\inf_{t \in \mathcal{T}} f(t)$, is defined as the largest value x such that $f(t) \geq x$ for all $t \in T$. We say that $\inf_{t \in \mathcal{T}} f(t) = -\infty$ if $f(t)$ has arbitrarily large values over $t \in T$.[1]

4 Cryptography

- Symmetric key cryptography requires a secure way to exchange keys.
- In public key cryptography (asymmetric key cryptography) you publish your public key and have a linked private key. This allows you to receive a message. Consider the scenario of Alice Sending a message to Bob.
  - Bob creates a public key `bob_public_key` and a linked private key `bob_private_key`.
  - Bob publishes `bob_public_key`.
  - Alice encrypts a message using `bob_public_key`.
  - Bob decrypts that message using `bob_private_key`. This works because Bob's public and private keys are linked.
  - Tom tries to read the message but he can't because he doesn't have `bob_private_key`.
- But how does Alice know she really has Bob's public key? Maybe someone pretending to be Bob gave her his public key.
- This is where the certificate authority comes in. From wikipedia: "When the user's web browser receives the public key from www.bank.example it can contact the certificate authority to ask whether the public key does really belong to www.bank.example. Since www.bank.example uses a public key that the certification authority certifies, a fake www.bank.example can only use the same public key. Since the fake www.bank.example does not know the corresponding private key, it cannot decrypt the user's answer."
- SSL uses public key cryptography.

[1] inf and sup taken word for word from http://www-bcf.usc.edu/ mjneely/ee599/liminf-notes.pdf

4.1 Cyptrography: another take

- Each person X has a public encryption key E_X and a private decryption key D_X such that for a plain text message P we have $P = D_X(E_X(P))$. The scheme is set up such that it is also true that $P = E_X(D_X(P))$.
- For A to send a message P to B, A can send $E_B(P)$, and only B can read it because only B can do $D_B(E_B(P))$.
- To sign a message P, A can send both P and the signature $D_A(P)$. Only A can generate this signature.
- A certificate authority C can create a certificate P that says the public key of X is E_X and sign it with the certificate authority's private key D_C to generate $D_C(P)$. Then, since everyone knows E_C, and certificate P is public, anyone can check the validity of the certificate by ensuring that $E_C(D_C(P)) = P$.
- A certificate authority can make a certificate that says anything, such as the hash of this program is 5641. If you run the hash algorithm on the program you downloaded and you don't get 5641, you know the program has been changed. If you know someone's public key, you can ensure that anything they sign is authentic.

5 Algebra

1. $a^2 - 2ab + b^2 = (a - b)^2$
2. You can't just take the square root of both sides.

6 Higher Math

Mathematics Consists of mathematical objects and mathematical operations. The operations map mathematical objects to other objects. We can think of these operations as transformations. We can think of functions as a transformation, elements of a group as a transformation. We can think of category theory as the generalized study of transformations. I've been talking about computation as transformations as well.

Analysis Most math is analysis, which deals with limits, derivatives, integrals, and stuff.

Topology Generalizes the idea of shape so that two items have the same shape (topology) if one can be transormed into the other without tearing it or making a hole. Mobius strip is a classic shape in ontology.

class A collection of sets.

monoid Is like a group. A set of objects M, a function $\star$, and an identity element e. If we are thinking of a monoid as performing actions, the members of M are the possible actions, and the state space is some other set S, and $\star$ applies one of the actions of M on a state in S. Also see http://www.michael-noll.com/blog/2013/12/02/twitter-algebird-monoid-monad-for-large-scala-data-analytics/.

model theory Is about constraints limiting all of the possible ways the world can be. If you know I went to school in Texas in 1992, then you know the world isn't such that I was in the army that year.

6.1 Domain Theory

The mathematical foundation of denotational semantics. In denotational semantics, a consctruct is given meaning by assigning it to an element in a *domain* of possible meanings (Winskel).

6.2 Topology

"Topology is the study of abstract shapes such as 7-dimensional spheres."[2]

[2] http://math.mit.edu/~dspivak/CT4S.pdf

6.3 Category Theory

It appears that a category is a whole class of things. You can have a category of sets, for example. Within that category, you have objects (e.g., the individual sets) and morphisms (e.g., functions between sets). A set of objects and morphisms form a *category* if the associative law holds and everything can be mapped to itself. I.e., one of the morphisms is $Id(a) = a$. The two axioms boil down to associative and identity.

Spivak says the interesting thing about category theory is the paths. From Healy and Caudell, "unlike the situation with graphs, a path through the arrows in a category is associated with a precise notion of cumulative effect of meaning; and, further, different paths whose compositions have the same domain and codomain can have the *same* meaning. When this is true, we have a cummutative diagram."

$Hom_{Set}(X, Y)$ is the set of all functions from X to Y where X and Y are two different sets.

6.3.1 Quotes from xx

Quotes from: http://math.mit.edu/~dspivak/CT4S.pdf

- "Dierent branches of mathematics can be formalized into categories. These categories can then be connected together by functors. And the sense in which these functors provide powerful communication of ideas is that facts and theorems proven in one category can be transferred through a connecting functor to yield proofs of analogous theorems in another category. A functor is like a conductor of mathematical truth."
- "Hierarchies are partial orders, symmetries are group elements, data models are categories, agent actions are monoid actions, local-to-global principles are sheaves, self-similarity is modeled by operads, context can be modeled by monads."
- " In mathematics, a category can also be construed as a collection of things and a type of relationship between pairs of such things. For this kind of thing-relationship duo to count as a category, we need to check two rules, which have the following avor: every thing must be related to itself by simply being itself, and if one thing is related to another and the second is related to a third, then the rst is related to the third. In a category, the things are called objects and the relationships are called morphisms."

6.3.2 Definitions

category A category consists of two classes, objects and morphisms. The objects can be a class of thigns like sets, and the morphisms are mappings between them. Morphisms are transformations.

morphism A structure-preserving mapping from one mathematical structure to another. What does structure-preserving mean? For example, it can be a mapping that preserves an associated binary operation. There are many kinds of morphisms. For example, a homomorphism is a "structure-preserving map between two algebraic structures (such as groups, rings, or vector spaces)" (Wikipedia).

homomorphism A structure-preserving mapping

functor Connects categories together. A link between categories, such as Set and Olog.

6.4 Abstract Algebra

- Algebra is the study of transformations. A symmetry is a transformation that leaves an object invariant. All of the elements in the set (called a group) are different transformations that leave a central object unchanged. For example, all of the ways to flip a triangle.
- Abstracts the numbers to transformations. The example of flipping a mattress. If you rotate and flip, that is the same as flipping it long ways. so $RF = L$. Flipping it twice takes you back to the same place so $I = FF$. In this case, I is the *identity element*. It is like a noop. The number 1 would be an identity element in an algebra with the natural numbers and a multiplication operator. The number 0 would be the identity element if the operator was addition.
- A Symmetry group for a circle is all the transformations that leave the circle invariant.

- You can have a group be a set $\{2, 3, 4\}$ along with an operator $\times$ (this is not an actual group, just for demonstration purposes), but those numbers can be transformations or functions, like flip a mattress. numbers can be transformation or functions instead of
- But what is special about identity (1 for *) and zero (0 for +). Why do these two things matter so much?
- the Algebra we learn in high school is the study of the transformations (e.g., subtract 9 from both sides) that leave the equation the same (symmetry).
- Can also say that "Algebra is the study of mathematical systems that, in some sense, generalize the behavior of numbers." So integers, polynomials, and n by n matrices are all different, but it makes sense to add and subtract them. And if you prove something, it can be said for all. Seems like math is like this, if you prove something for a class it is true for all members of the class.

6.5 Real Analysis

It seems like this kind of math is about setting up a set of objects and a set of operators, both subject to some restrictions. Then proving properties about these kinds of systems.

Topological Space A set X and a collection of subsets τ of the set X such that the union of any two sets of τ is in τ and the intersection of any two sets in τ is in τ. If items of X are in the same subset in τ then they are "next" to each other. They are in the same neighborhood.

Open set Each subset in τ is an open set. An open set is on that you can go a little ways in each direction and still be in the set. E.g. $(-4.0, 5.0)$ is an open set. You can always go a little closer to 5.0.

Linear space (vector space) A set of items with a sum, negative, product operator and a **0** element and a **1** element that meet certain conditions. They also meet the conditions of linearly (the standard definition of linear). The dimension of the set is the largest number of items such that no item is a linear combination of the others. In linear algebra we studied finite linear spaces (the dimension is finite). But the dimension can be infinite, like with Hilbert spaces and Banach spaces.

Null space You can have subspaces of linear spaces. One example is the null space. The null space of a linear operator or function f is the set of objects X such that $f(x) = 0$.

Norm In a general linear space, the norm is an extension of the absolute value of the real numbers. In a vector space, the norm is the length of the vector.

Measure A non-zero number assigned to a set, such as height. Assigns numbers to subsets.

7 Geometry

1. *What is a geometry?*
2. Progression of *Euclidean* (can just move), *affine* (can move and stretch and resize), and *projective* (parallel lines no longer parallel) geometry, with projective being the most general.
3. Two ways of reasoning in geometry. *Synthetic* argue about geometric entities (point, line, etc.) and geometrical relations between them. *Analytical* represent geometrical entities by coordinates and equations so that algebraic manipulation can be used.

8 Calculus

8.1 Differentiation

A *differential* is an infinitesimally small change in a variable. Consider the function $y = f(x)$, the differential of y, denoted by $\mathrm{d}y$, is related to $\mathrm{d}x$ by

$$\mathrm{d}y = \frac{\mathrm{d}y}{\mathrm{d}x}\mathrm{d}x \qquad (1)$$

The derivative of $f(x)$ is given by

$$\frac{\mathrm{d}y}{\mathrm{d}x} = \lim_{\Delta x \to 0} \frac{\Delta y}{\Delta x} = \lim_{\Delta x \to 0} \frac{f(x+\Delta x) - f(x)}{\Delta x} \tag{2}$$

There are multiple notations that all mean the same thing

$$\frac{\mathrm{d}y}{\mathrm{d}x} = f'(x) = y' = D_x[y] = \frac{\mathrm{d}}{\mathrm{d}x}[f(x)] \tag{3}$$

8.2 Integration

If we have the differential equation

$$\frac{\mathrm{d}y}{\mathrm{d}x} = g(x) \tag{4}$$

we can rewrite this as

$$\mathrm{d}y = g(x)\mathrm{d}x \tag{5}$$

and if we anti-differentiate both sides, we get

$$y = \int g(x)\mathrm{d}x = G(x) + C \tag{6}$$

For example, consider $g(x) = 2x$. Then $y = \int 2x\mathrm{d}x = x^2 + C$.

8.3 Partial Derivatives

The definition of the partial derivative of x with respect to y is

$$f_x(x,y) = \lim_{\Delta x \to 0} \frac{f(x+\Delta x, y) - f(x,y)}{\Delta x} \tag{7}$$

(y with respect to x is analogous). Consider the equation

$$f(x,y) = 3x - x^2y^2 + 2x^3y \tag{8}$$

the partial derivative with respect to x (treating y as a constant) is

$$f_x(x,y) = 3 - 2xy^2 + 6x^2y \tag{9}$$

The notation for the partial derivative is

$$f_x(x,y) = \frac{\partial}{\partial x} f(x,y) = z_x = \frac{\partial z}{\partial x} \tag{10}$$

8.4 Differential Equations

A differential equation of order n has a general solution with n arbitrary constants. Remember that $\mathrm{d}y = y'\,\mathrm{d}x$.

Differential equation in x and y is an equation that involves x, y, and derivatives of y, e.g. $\dot{y} = 4x$. The solution to this simple equation is to integrate both sides.

Separation of variables

$$\begin{aligned}
\frac{\mathrm{d}y}{\mathrm{d}x} &= \frac{2x}{y} \\
y\,dy &= 2x\,dx \\
\int y\,\mathrm{d}y &= \int 2x\,\mathrm{d}x \\
\frac{1}{2}y^2 &= x^2 + C_1 \\
y^2 - 2x^2 &= C
\end{aligned}$$

The general form is $M(x) + N(y)\frac{\mathrm{d}y}{\mathrm{d}x} = 0$. With a homogeneous differential equation then it can be separated with a change in variables.

Exponential growth and decay model Often the rate of change of a variable y is proportional to the value of y. An example is radioactive decay and the half-life of a material, where y would be the amount of material. If y is a function of time t, the proportionality can be written as $\mathrm{d}y/\mathrm{d}t = ky$ and the general solution to this differential equation is

$$\begin{aligned}
\dot{y} &= ky \\
\frac{\dot{y}}{y} &= k \\
\int \frac{\dot{y}}{y}\mathrm{d}t &= \int k\mathrm{d}t \\
\int \frac{1}{y}\mathrm{d}y &= \int k\mathrm{d}t \quad (\mathrm{d}y = \dot{y}\mathrm{d}t) \\
\ln y &= kt + C_1 \\
y &= e^{C_1}e^{kt} \\
y &= Ce^{kt} \quad (C = e^{C_1})
\end{aligned}$$

Exponential growth occurs when $k > 0$ and exponential decay occurs when $k < 0$.

Differential equation solution A function $y = f(x)$ is a solution of a differential equation if the equation is satisfied when y and its derivatives are replaced by $f(x)$ and its derivatives. For example, $y = 4e^{-x}$ is a solution to the differential equation $y'' - y = 0$ because $y = 4e^{-x}$, $y' = -4e^{-x}$, and $y'' = 4e^{-x}$. So $4e^{-x} - 4e^{-x} = 0$.

Difference equation Also known as a recurrence relation, is an equation which defines a sequence recursively: each term of the sequence is defined as a function of the preceding terms. For example (the logistic map): $x_{n+1} = rx_n(1 - x_n)$. When solving an ordinary differential equation numerically, one typically encounters a recurrence relation.

A difference equation is a discrete version of a differential equation. For example two difference equations are

$$\begin{aligned}
\mathbf{x}_{t+1} &= \mathbf{x}_t + \Delta t \cdot \dot{\mathbf{x}}_t \\
\dot{\mathbf{x}}_{t+1} &= \dot{\mathbf{x}}_t + \Delta t \frac{\mathbf{f}_t}{m}
\end{aligned}$$

and the corresponding differential equation would be (THESE MAY BE WRONG, NEED TO FIX)

$$\begin{aligned}
\mathbf{x} &= \int \frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t}\mathrm{d}t \\
\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} &= \frac{\mathbf{f}}{m}\mathrm{d}t
\end{aligned}$$

Notice how we need an initial condition for the first differential equation to make sense.

8.5 Other Calculus Topics

Taylor Series Used to give a polynomial (or linear) representation of a function. Approximates the behavior of a function in a small area around the value a and is given by

$$T(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!}(x-a)^n$$

If $a = 0$ then it is called a Maclaurian series.

Directional Derivative Is a scalar, and is the derivative (or rate of change) in a particular direction. The directional derivative of f in the direction of the unit vector $\mathbf{u} = \cos\theta\mathbf{i} + \sin\theta\mathbf{j}$ is

$$D_u f(x,y) = f_x(x,y)\cos\theta + f_y(x,y)\sin\theta$$

Gradient Is a vector, and is the direction of maximum slope, its norm is the value of that slope and is the maximum value of the directional derivative at that point

$$\nabla f(x,y) = f_x(x,y)\mathbf{i} + f_y(x,y)\mathbf{j}$$

Divergence Is a scalar, the rate of particle flow per unit volume. For a vector field $F = M\mathrm{i} + N\mathrm{j} + P\mathrm{k}$ the divergence is give by

$$divF(x,y,z) = \nabla \cdot F(x,y,z) = \frac{\partial M}{\partial x} + \frac{\partial N}{\partial y} + \frac{\partial P}{\partial z} \tag{11}$$

9 Logic

From [**?**], page 240.

sentence Represents some assertion about the world.

knowledge base A knowledge base, KB, is a set of sentences.

model A world state. The set of all models is the set of all possible states.

satisfies A model m satisfies a sentence α if sentence α is true in model m. If a model m satisfies sentence α, then m is a model of α. $M(\alpha)$ means all of the models of α; i.e., all of the modes in which α is true.

entails $\alpha \models \beta \iff M(\alpha) \subseteq M(\beta)$

model checking Check that $M(KB) \subseteq M(\alpha)$. Check that a sentente α is entailed by your knowledge base. Said another way, to do model checking, you check that for every model in which your knowledge base is true, α is true.

fluent An aspect of the world that changes. E.g. $At(s_1)$. A function or relation (predicate) that can vary from one time step to the next.

frame problem Logic moves through time like frames like the frames of a movie. If you move a table at time t_k, is your textbook still at the same physical place t_{k+1}? You don't know. It depends on if it was on the table. You have to explicitly say all of the things that don't change for every action. That is a lot of stuff.

database semantics The closed-world assumption (anything specifically not stated or proved is assumed false), and there is a one-to-one mapping between objects and symbols.

model logic Allows one to reason about beliefs of others.

Propositional logic (like probability) deals only with facts. By contrast, first-order logic deals with objects, relations, and facts. You can propositionalize first-order logic to propositional logic, but the state space blows up. [What, exactly, are the advantages of having it in propositional form? You make the infinite finite?]

9.1 Propositional Logic

model A model in propositional logic is a mapping from every propositional letter to either true or false.

9.2 First Order Logic

First order logic is used to model the real world. It is used to represent the world (knowledge representation) and it is used to determine what is true about the world based on your model and what you already know (inference). Terms from [1] and also from Russel and Norvig page 293.

The weird thing about first-order logic is you have these objects, which may or many not have symbols that match to them. They are just there in the ether.

First order logic has objects and True/False values.

1. Sentences, formulas, predicates, relations, atoms, and literals map to True/False.

2. Terms, constants, variables, or functions mapt to objects.

First order logic is also a kind of generalization—it is a kind of abstraction.

objects A set of things in the real world that you want to model.

term Maps to object. Any expression representing an object in the domain. It can be a constant, variable, or function applied to a tuple of terms.

atom or predicate or relation Maps to truth value. Also called an *atomic formula*. A predicate symbol applied to a tuple of terms.

literal An atom or its negation.

formula or sentence Recursively constructed from atoms using logical connectives (e.g. $\forall$, $\exists$, $\wedge$, $\vee$, and $\neg$).

ground term Term containing no variables.

ground atom Also *ground predicate*. Atom where all arguments are ground terms.

possible world Also called *Herbrand interpretation*. Assigns a truth value to each possible ground atom. Can think of this as an instantiation of the set of predicates.

interpretation Maps every symbol to an object. Specifically, it maps: constant symbols to objects, predicate symbols to relations,function symbols to functions. There can be more objects than there are constants, and vice versa. If you don't know the number of objects, there can be an unbounded number of interpretations.

model A set of objects plus an interpretation.

9.3 Inference

resolution Allows you to do inference in first order logic without grounding the KB (it is *lifted*).

lifted Where you do somthing such as inference whithout having to ground the KB in propositional variables.

9.4 Knowledge Representation

circumscription Allows you to do default reasoning. For example, $Bird(x) \wedge \neg Abnormal(x) \Rightarrow Flies(x)$. Circumstription can assume that $\neg Abnormal(x)$ unless $Abnormal(x)$ is known to be true.

answer set programming A refinement of circumscription. You can also compare answer set programming with Prolog. Prolog gives you one answer, answer set programming gives you all the answers (objects that fit).

semantic network A subset of first-order logic where inference is computationally tractable. A semantic network provides graphical aids for visualizing a knowledge base and an efficient algorithm for inferring the properties of an object on the basis of category membership.

description logic A subset of first-order logic where inference is computationally tractable. Constructs and provides category definitions and provides efficient algorithms for deciding subset and superset relationships.

OWL Web Ontology Language. OWL is a description logic.

10 Planning

- RRT ... something Random Trees.
- STRIPS is a lifted representation (meaning that it can use the abstraction of first-order logic) that through its add list and delete list deals with the frame problem.
- Subgoal Interactions. When achiving a subgoal undoes some other part of the goal you want to achieve.

10.1 Graph-based planning

Convert the problem into a graph with nodes as states and edges as links between them. Then you can do a search.

- But you have to make sure that you don't unset something. *What is this called?*
- Heuristic search speeds things up because you can use the extra information of the estimated distance to the goal.
- GraphPlan appears to be the state of the art.

10.2 Hierarchical Planning

Hierarchical decomposition. Planning algorithms can handle thousands of steps? States?

11 Misc.

Transaction Data Data from transactions, such as purchase orders. Master data is data about static things like customers.

Invariant Something that doesn't change.

Symbol Something that stands for something else. A symbol is a discrete entity that by standing for something else has semantic meaning.

Relation For sets A, B, a subset of $A \times B$ is called a relation from A to B.

Function A special kind of relation in which for each $a \in A$ there is no more than one $b \in B$.

Mahalanobis distance Calculates distance between two objects by taking variance into account. Distance from a mean along a direction where there is a large variance has less weight than distance from the mean along a direction with little variance. $\delta(\mathbf{x}; \mu_k, \Sigma_k)$

Grammar Face $\rightarrow$ 2 eyes, 1 nose, 1 mouth

Abstraction In categorization, is to single out some subset of the sensory input and to ignore the rest. We are always dealing with abstractions. All categories are abstractions. Even a person is an abstraction. Bob looks different today than he did yesterday, and he is in a different place, and his atoms are different, but we still call him Bob.

Emergent behavior A behavior that only involves hidden states (not sure about that definition). You can define wall following manually, or it can emerge by having the robot avoid hitting obstacles but make the robot be attracted to obstacles and be attracted to moving forward. The quintessential example is how ants use scent to make a path. They have no concept of the path they just follow the scent and the path emerges. I'm wondering, would this be an example of an emergent property

$$\begin{aligned} a &\rightarrow b && (12)\\ b &\rightarrow c && (13)\\ \text{set } a &= True && (14)\\ c &\equiv \text{emergent property?} && (15) \end{aligned}$$

Complex numbers Numbers of the form $a + bi$. Have an imaginary y axis and a real x axis. If represented in polar form then $\cos\theta + i\sin\theta = \exp^{i\theta}$. This leads to the famous equation $\exp^{i\pi} + 1 = 0$ that incorporates all of the most fascinating numbers known. These number comes from the fact that the set of real numbers is not closed under polynomial equations. Consider $x^2 = -1$, this equation has no solution in the set of real numbers. So you need complex numbers.

Linear System A linear function $f(x)$ is one that satisfies the following properties

$$\begin{aligned}\text{Additivity: } f(x+y) &= f(x)+f(y) \\ \text{Homogeneity: } f(\alpha x) &= \alpha f(x)\end{aligned}$$

these properties together are called *superposition*, and means that a "net result caused by two or more phenomena is the summation of the results that would have been caused by each phenomenon individually" (wikipedia).

Analogy It can be considered that creativity is reasoning by analogy.

1. a has properties C, D, E, F, and G.
2. b has properties C, D, E, and F.
3. So b probably has property G [**?**].

False analogy is if b and a do not match in the relevant attributes (although they may match on irrelevant attributes). So the trick is if you are trying to understand b is to find an a that you do understand that is similar in the relevant respects to b.

12 Optimization

Notation $J(\theta)$ is a function of θ that gives the value of the function you want to optimize when θ is that value. For example, assume we want to minimize (θ^2+1). We would then have $J(\theta)=\theta^2+1$, so if $\theta=2$ then we know the value, and we are looking for a value of theta that minimizes $J(\theta)$ and thus (θ^2+1). It's confusing because it seems like unnecessary notion. J just stands for the function you want to minimize.

Find where derivative 0 You know this is a local miximum, a local minimimum, a saddle point, or a plateau. You can't use this method if you also have separate constraint functions, such as $g(x,y)=0$.

Lagrange Multipliers Given function $f(\mathbf{x})$ we wish to optimize and a constraint function $g(\mathbf{x})=0$, we can optimize this by optimizing the Lagrangian function $L(\mathbf{x},\lambda)=f(\mathbf{x})+\lambda g(\mathbf{x})$.

Linear Programming Simplex algorithm. [give run time].

Newton's method

Convex Optimization A subfield of optimization, studies the problem of minimizing convex functions over convex sets.Good because if a local minimum exists, it is a global minimum. This ties into duality because many optimization problems can be reformulated as convex minimization problems. For example, the problem of maximizing a concave function f can be re-formulated equivalently as a problem of minimizing the function $-f$, which is convex.

13 Machine Learning

13.1 General Terms

Machine Learning The Bayes classifier tries to calculate $P(\omega_i|\mathbf{x})$ using $P(\mathbf{x}|\omega_i)$. There are a couple of statistical ways to get $P(\mathbf{x}|\omega_i)$, e.g. assuming a Gaussian and trying to calculate Θ_i. But if this approach doesn't work then you need to use things like neural nets or support vector machines. Consider

$$P(\omega_i|\mathbf{x})=\frac{P(\mathbf{x}|\omega_i)P(\omega_i)}{P(\mathbf{x})} \tag{16}$$

In this case $P(\mathbf{x}|\omega_i)$ is the generative part, because given the hypothesis it can generate the data. The discriminative part is $P(\omega_i|\mathbf{x})$. So, if all you need to do is discriminate you can use a SVM and never have to compute $P(\mathbf{x}|\omega_i)$.

Regularization Simple models help overfitting and regularization is used to reduce overfitting. Usually takes the form of a term added to a minimization objective function that serves to dampen down the solution.

Lasso Regularization for regression that minimizes the absolute values of the components of the weight vector.

Ridge regression Regularization for regression that minimizes the squared values of the components of the weight vector.

Logistic regression Is a classifier, gives a value between 0 and 1. You are basically taking the sigmoid of $P(x|C_1)P(C_1)/P(x|C_2)P(C_2)$. (This isn't it exactly, but close enough for now.)

You project the point onto each hyperplane and take the closest one. $P(Y = i|x, W, b) = softmax(Wx + b)$ where each row i of W represents the centroid of class i. You learn W and b by gradient descent by maximizing the likelihood of $P(Y = i|x, W, b)$. Logistic regression is equivalent to a single-layer neural network. Each neuron is represented by W_i, b_i.

Bias-variance tradeoff Frequentist view of model complexity. Flexible models have low bias and high variance because they overfit the choice of dataset.

Loss function Inverse of a utility function. The cost of incorrectly assigning an object of class C_i to class C_i. $L(C_i, C_i) = 0$. C_i could be "has cancer" and C_j could be "cancer free."

Maximum likelihood With a Gaussian likelihood function ... find the parameters that make the data the most likely given those parameters.

Relational Learning Learning tasks in which the data points associated with the same output do not share coordinates and do not cluster together. In parity, the values would not cluster together. In non-relational learning the values do cluster and that's why fence and fill methods work.

Kernel A kernel seems to have two meanings (although I imagine that they are related). The first is that the kernel is like a weighting function. For kernel regression you use all the points but weight them by a kernel, like a Gaussian.

A second meaning of kernel is as in the kernel trick. The kernel trick is a projection into another space where classes are linearly separable. You use the kernel so you don't actually have to do the projection. A common example is x^2, this allows you to separate points that are clustered around 0 form other classes more positive and negative. In this sense a kernel is more associated with the dot product. A kernel performs a transformation that you hope will make the data more easily separable.

Manifold Learning Manifold learning is an optimization problem to find a low-dimensional representation that preserves some property of interest.

Functional $F[y]$ returns a value for function y.

Calculus of Variations Seek a function y that maximizes some functional $F[y]$.

Curse of Dimensionality If we divide a region into spaces a-la fence-and-fill, then the number spaces grows exponentially in the dimensionality of the data. This means that we need an exponentially large amount of training data.

Bayes Optimal Compute $P(C_1|x)$ and $P(C_2|x)$ using Bayes rule. Put a decision boundary at the crossover point.

Bayes Error Plot $P(C_1|x)$ and $P(C_2|x)$ and it is the area of wrong regions.

13.2 Bishop

I want to better understand the relationship between neural networks, regression, and support vector machines.

In the first part of the book, the basis funcitons are fixed. An example is Gaussian mixture models. You set the locations of the basis functions ahead of time. Bishop says that to get good performance, you have to adapt the basis functions to fit the data. Neural networks do this because the hidden nodes get parameters tuned to the data. Support Vector Machines pick the support vectors to represent the basis functions because those are the vectors on the decision boundary.

13.3 Gaussian Process

It's a generalization of Bayesian regression. A Kalman filter is an example of a Gaussian process. The value of the unknown point y for $y_{t+1} = f(x_{t+1}) = x_{t+1} + covariance(x_{t+1}, x_{0:t}) \times (y_{0:t} - x_{0:t}) \times (1/covariance(x_{0:t}))$. [3] Maybe. Probably need to watch it again. But, regardless, this y_{t+1} is given as a Gaussian distribution; it is not a point estimate.

A *Gaussian process* defines a distribution of functions $p(f)$ over a function $f : \mathcal{X} \to \mathbb{R}$.[4] The covariance is between the inputs (the x axis) and not the values for those inputs. The values for those imputs are used later. Following Ebden[5], if we want to know a new point $f(x_*) = y_*$ we have the vector of covariances $K_* = [k(x_*, x_1), k(x_*, x_2), \ldots, k(x_*, x_n)]$, and if K is the covariance matrix on known values $x_i \ldots x_n$, and $\mathbf{y}$ is the y values of the training data, then

$$y_* = K_* K^{-1}\mathbf{y} \tag{17}$$

It looks to be a way to do kernel regression but it is easier to set the parameters. Regression predicts $f(x) = y$ and from the kernel function $k(x, x')$ you build a covariance matrix that gives the covariance between each two training points (each entry in the matrix is a value $k(x, x')$). You also have a mean vector. With a covariance matrix and a mean vector you get a Gaussian. Then to get the prediction for y at a point x^*, you just query the Gaussian. For point x^* you also have a variance.

You represent the set of n training examples as a big n-dimensional Gaussian. You define a kernel function on the distance between data points (which, I guess, is distance between dimensions in the Gaussian). Then, you create a covariance matrix of the inputs using the kernel and use that to make predictions of unseen data. You don't have to specify the model, but you do have to specify the kernel function. So it almost feels like the same thing.

A Gaussian process is a Gaussian distribution of infinite dimension.

$$f(x) \sim \mathcal{GP}(m(x), k(x_i, x_j)) \tag{18}$$

For example, for y_t at different times t,

$$\langle y_1, y_2, y_5, y_{13}, y_{20}\rangle \sim \mathcal{GP}(m(t), k(t, t')) \tag{19}$$

Another way to say this is you have in infinite Gaussian distribution and you marginalize any subset of that to get a multi-variate Gaussian. The infinite distribution is the infinite vector (function).

13.4 Ensemble Methods

Ensemble methods generate many simple models that vote to classify a new instance. Methods vary on two important aspects

- Each model is learned by experiencing different portions (or different weightings) of the training data.
- During voting, each model is weighted by its error on the training data (more accurate models have more weight).

The advantage of ensemble methods is that you get a lot of independent votes, which allows you to be more accurate.

Bagging Trains multiple instances of a classier on different subsamples (bootstrap samples) of the training data. The decision on an unseen test record is made by a majority vote among the base classiers.

Boosting Focuses classification learning on hard-to-classify items. On each round, each item to be classified has a weight based on how well it was classified in previous rounds. And on each round, the classifier with the lowest weighted error is chosen. The final classification method is then to take a weighted average of the vote of the individual best classifiers that were learned in each round. The weight of each classifier's vote is proportional to the accuracy it had in that round.

Random Forest Each tree only sees a subset of the features.

Viola and Jones Use ensemble methods to do face detection. They made each classifier a feature and had to learn the threshold and polarity for each classifier. E.g. for white/black vertical feature may say if face if greater than 20. Viola and Jones use boosting to find the best features by making each feature its own classifier.

[3] `https://www.youtube.com/watch?v=JdZr74mtZkU`
[4] http://mlss2011.comp.nus.edu.sg/uploads/Site/lect1gp.pdf
[5] http://www.robots.ox.ac.uk/ mebden/reports/GPtutorial.pdf

14 Kernels Methods

- A kernel is a function on two vectors that returns a scalar with special properties. It is interpreted as defining a similarity measure between two data points.
- A kernel is an equivalence relation defined over the domain of a function f such that two elements x and y have the same value under f so that $f(x) = f(y)$.

14.1 The Kernel Trick

- Let $\mathbf{x} \cdot \mathbf{y}$ be the inner product (dot product) of two vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$.[6]
- We consider a kernel function $\mathcal{K}(\mathbf{x}, \mathbf{y})$ that corresponds to some mapping $\phi(\mathbf{z})$ of a vector $\mathbf{z} \in \mathbb{R}^n$ into a high dimensional space $\mathbb{R}^m$ where $m > n$.
- If
  - $\mathcal{K}(\mathbf{x}, \mathbf{y}) = \phi(\mathbf{x}) \cdot \phi(\mathbf{y})$ and
  - your learning algorithm for learning in high-dimensional space only needs to compute inner products

  then you don't need to really compute $\phi(\mathbf{z})$ because the algorithm is only using $\phi(\mathbf{x}) \cdot \phi(\mathbf{y})$ and that is equal to $\mathcal{K}(\mathbf{x}, \mathbf{y})$.
- Often, you don't even know what $\phi(\mathbf{z})$ is exactly, you just know that $\mathcal{K}(\mathbf{x}, \mathbf{y}) = \phi(\mathbf{x}) \cdot \phi(\mathbf{y})$.
- Since Support Vector Machines only need to compute inner products, they work well with the kernel trick.

Example: let $\mathbf{x}, \mathbf{y} \in \mathbb{R}^2$ so that $\mathbf{x} = \langle x_1, x_2 \rangle$ and $\mathbf{y} = \langle y_1, y_2 \rangle$. One mapping is $\phi(\mathbf{x}) = \langle x_1^2, x_2^2, x_1x_2, x_2x_1 \rangle$ so that $\phi(\mathbf{x}) \in \mathbb{R}^4$. If the kernel function $\mathcal{K}(\mathbf{x}, \mathbf{y})$ is the square of the dot product of $\mathbf{x}$ and $\mathbf{y}$, then

$$\begin{aligned}
\mathcal{K}(\mathbf{x}, \mathbf{y}) &= (\mathbf{x} \cdot \mathbf{y})^2 \\
&= (x_1y_1 + x_2y_2)^2 \\
&= x_1y_1x_1y_1 + x_1y_1x_2y_2 + x_2y_2x_1y_1 + x_2y_2x_2y_2 \\
&= x_1^2y_1^2 + x_2^2y_2^2 + x_1x_2y_1y_2 + x_2x_1y_2y_1 \\
&= \phi(\mathbf{x}) \cdot \phi(\mathbf{y})
\end{aligned}$$

This kernel function is referred to as second degree polynomial.

15 Deep Learning

But when they say RMM, do they mean recurrent neural network or recursive neural network?

- Cross entropy error. If prediction value is $x \in [0, 1]$ is $-\log(x)$. http://visualstudiomagazine.com/articles/2014/04/01/neural-network-cross-entropy-error.aspx
- Deep Belief Network. Hinton. A stack of Restricted Boltzman machines. Gives you increasingly abstract features as you go up.
- Recursive Neural Network. Tree of grammar. You always represent each node using the same length vector.
- Recurrent neural network. There is a side area that holds state. Really? Like a chain. A recurrent neural network is like a more sophisticated HMM. "Recurrent neural networks are a special case of recursive neural networks that operate on chains and not trees." [Paulus, Socher, and Manning]

[6]This explanation was done in Euclidean space $\mathbb{R}^m$, but more generally applies to any Hilbert space. (The dot product becomes the inner product.) [I need to figure out exactly how all this is related.] [Also, Hilbert space, does this relate to what Matt was saying with trees?]

15.1 Energy Based Model

You have a graph with symmetric connections and weights on the connections. If two nodes are "on" that energy is added (subtracted) to the energy of the model.

Hopfield nets are examples and so are Boltzman machines.

They find stable, low-energy states so they can be used to store memories. They say it gives a content addressable memory because you can set (address) a subset of the nodes to complete a memory. I can tell you a little about something (set some of the nodes) the other nodes will converge to their states and complete the memory. Like reconstructing a dinosaur from a few bones (not my analogy).

16 Probability

Stochastic process Is a sequence of random variables. A process whose behavior is non-deterministic in that the next state of the environment is not fully determined by the previous state of the environment.

Generative model "A model for randomly generating observed data, typically given some observed parameters" (wikipedia). Specifies how causes generate effects. Bayes nets and HMMs are generative models. Q-learning is *model free* because the transition function does not have to be specified.

Bayesian vs. Frequentest Klaus was a frequentest. To a Bayesian (a subjectivist), probabilities can be interpreted as degrees of belief, but a frequentest rejects this and and assigns probabilities only to random events according to their relative frequencies of occurrence. Whereas a frequentest and a Bayesian might both assign a 1/2 probability to the event of getting a head when a coin is tossed, only a Bayesian might assign 1/1000 probability to a personal belief in the proposition that there was life on Mars a billion years ago. This assertion is made without intending to assert anything about relative frequency. One problem with Bayesian probability is the priors. Consider 1) a box you know contains black and white balls 2) a box sampled to have 50% white and 50% black balls, and 3) a box in which you know there are half white and half black balls. A Bayesian would give the probability of 0.5 to pulling a black ball in each case.

Statistical inference Inference about a population from a random sample drawn from it or, more generally, about a random process from its observed behavior during a finite period of time. (Wikipedia)

Bayesian inference A statistical inference in which probabilities are interpreted not as frequencies or proportions or the like, but rather as degrees of belief. (Wikipedia)

Probabilistic inference The computation of the posterior probability distribution for a set of query variables given some observed event–that is some assignment of values to a set of evidence variables.

Random Variable A numerical attribute. It is a function $X : \Omega \to \Re$ that assigns a real number to each outcome of an experiment. Can also have Boolean random variables which take on the values True and False and discrete random variables which take on a countable number of values.

Probability Distribution Assigns to every interval of the real numbers a probability, so that the probability (Kolmogorov) axioms are satisfied. The axioms are (1) the probability of each event is ≥ 0 (2) probability of all events sum to 1 (3) the probability of an event that is the union of a set of disjoint events is the sum of those events.

(Cumulative) Distribution function A function that gives the probability of the value of a random variable X being below a given value x. Given by:

$$F(x) = P(X \leq x)$$

Probability mass function A function $P(x)$ over a random variable x with a finite number of values such that:

$$P(x) \geq 0, \text{ and } \sum_{x \in X} P(x) = 1$$

Probability density function A probability mass function in which X is a continuous random variable. The probability density function $p(x)$ (note lowercase notation) is the derivative of the cumulative distribution function $F(x)$ and must satisfy:

$$p(x) \geq 0, \text{ and } \int_{-\infty}^{\infty} p(x)\mathrm{d}x = 1$$

Gaussian distribution The probability density function for the normal distribution is given by

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

There is no closed form expression for the integral of this this function, so it needs to be integrated numerically and results put into a table to find "the area under the curve." To do this we use the $N(0,1)$ distribution for which the table is calculated and use $Z = (X - \mu)/\sigma$.

Due to the central limit theorem, a distribution is assumed to be approximately normal if it comes from the presence of a large number of small effects acting independently and additively. The sampling distribution of the mean is approximately normal, even if the underlying distribution being sampled is not normal.

Central Limit Theorem The sum of many independent identically distributed random variables with a finite variance will be approximately a normal distribution.

Moment The n^{th} moment of a real-valued function $f(x)$ of a real variable is

$$\mu'_n = \int_{-\infty}^{\infty} x^n f(x)\mathrm{d}x$$

If f is a probability density function then the n^{th} central moment of the probability distribution of a random variable X is given by

$$\mu_n = E((X - \mu'_1)^n)$$

where μ'_1 is just the mean. The second central moment is the variance. The third central moment is the skewness which is negative if most mass on left and positive if most mass on right (normal distribution is 0). The fourth central moment is the kurtosis which is used to find the peakedness or how tall and skinny it is verses how short and fat.

Mixture Distribution A distribution with k components, each of which is a distribution. A data point is generated by first choosing a component and then generating a sample from that component based on its distribution. If C is a random variable denoting the component with values $1 \dots k$ then the mixture distribution is given by:

$$P(\mathbf{x}) = \sum_{i=1}^{k} P(C = i)P(\mathbf{x}|C = i)$$

The the individual components are Guassians then you have a *mixture of Guassians* family of distributions.

16.1 Measure Theory

Consider a sample space Ω of possible events. A σ-algebra $\mathcal{F}$ is a collection of subsets of Ω such that

1. $\emptyset \in \mathcal{F}$
2. if $A \in \mathcal{F}$ then $A^c \in \mathcal{F}$
3. if $A, B \in \mathcal{F}$ then $A \cup B \in \mathcal{F}$

A probability measure P then is a function $P : \mathcal{F} \to [0, 1]$ such that

1. $P(\Omega) = 1$
2. $P(\cup A_i) = \sum_i P(A_i)$ where the sets $\{A_i\}$ are disjoint.

In this case P is a measure and gives a notion of "size." The most common type of σ-algebra is the Borel sigma algebra. You need this fancy measure theory in continous environments because there can be weird inconsistencies if you don't use it.

17 Statistics

Likelihood The likelihood of a set of parameter values given some observed outcomes is equal to the probability of those observed outcomes given those parameter values. If you have a Gaussian model and you notice the average weight of cars is 3,000 pounds, then the mean of 3000 will have high likelihood. Likelihood = $P(\text{observed data}|\text{model}, \theta)$ where θ is the set of parameters for the model (e.g. Gaussian). (Wikipedia)

Likehood talks about the past, probably of past events given a situation, and probability talks about the future.

Also see http://stats.stackexchange.com/questions/2641/what-is-the-difference-between-lik

17.1 Non-parametric Statistics

You don't assume a model, like a Gaussian. A histogram is a simple non-parametric estimation of a distribution.

You just count things in the data. You look at the empirical distribution, which is a cumulative distribution and is computed by counting the number of datapoints with value less than t. You have things like the KolmogorovSmirnov test.

18 The Bayesian Approach

What makes an analysis Bayesian is treating the hypothesis $\mathcal{H}$ that you want to estimate as a random variable. You must clearly identify what the set of hypothesis $\mathcal{H}$ is, and what you end up with is a posterior distribution over $\mathcal{H}$.

Bayesian methods treat everything as probabilities and you can use this to replace cook-book statistical tests.

Using Bayesian methods you calculate the posterior $P(\mathcal{H}|D)$ of the hypothesis $\mathcal{H}$ given the data D. Prediction is then made using all the hypothesis, weighted by their probabilities, rather than using just the best one.

You have a prior $P(\mathcal{H})$ over hypothesis and using the generative model $P(D|\mathcal{H})$ and Bayes rule you get

$$P(\mathcal{H}|D) = \frac{P(D|\mathcal{H})P(\mathcal{H})}{P(D)}$$

where

$$P(D) = \sum_{\mathcal{H}} P(D|\mathcal{H})P(\mathcal{H})$$

The variance of the posterior gives you a sense of the uncertainty in the model. If you use a beta distribution as your prior, then the hyper-parameters are the parameters for the beta distribution.

[In Bayesian methods, what kinds of models are used for $P(D|\mathcal{H})$?]

You can do temporal updates with Bayesian. The posterior $P(\mathcal{H}|D)$ then can become the new prior $P(\mathcal{H})$ for the next experiment. So a way to say this is that given what you currently know $P(\mathcal{H})$, and given some evidence D, you update what you currently know $P(\mathcal{H}|D)$ by looking at the probability of that evidence given what you currently know $P(D|\mathcal{H})$ and $P(\mathcal{H})$ and $P(D)$ so that

$$P(\mathcal{H}') = P(\mathcal{H}|D) = \frac{P(D|\mathcal{H})P(\mathcal{H})}{P(D)}$$

and then when new data D' comes in, you have

$$P(\mathcal{H}'') = P(\mathcal{H}'|D') = \frac{P(D'|\mathcal{H}')P(\mathcal{H}')}{P(D')}$$

(Although $P(D'|\mathcal{H}')$ is just a model of how the world works and probably won't change as your observations change.) This part $P(D'|\mathcal{H}')$ is the crux. If what you saw, D', is likely given how you think the world is, $\mathcal{H}'$, then you are more likely to think the world is that way so you update to $\mathcal{H}''$. If you see the Eiffel Tower, D', then you are more likely to think you are in Paris ($\mathcal{H}'' =$ Paris has higher probability). And this is weighted by how likely it seemed that you were in Paris to begin with ($\mathcal{H}' =$ Paris has high or low probability). And $P(D)$ is just so everything stays a probability. In this example, $\mathcal{H}$ is a distribution over major cities.

Bayes formula Given evidence e:

$$P(Y|Xe) = \frac{P(XYe)}{P(Xe)} = \frac{P(X|Ye)P(Ye)}{P(X|e)P(e)} = \frac{P(X|Ye)P(Y|e)P(e)}{P(X|e)P(e)} = \frac{P(X|Ye)P(Y|e)}{P(X|e)} \tag{20}$$

Maximum a posteriori (MAP) Make predictions based on the single most probable hypothesis, that is the hypothesis h_i that maximizes $P(h_i|\mathbf{d})$.

Maximum likelihood (ML) hypothesis If all the hypothesis are given the same prior probability, then MAP learning reduces to choosing an h_i that maximizes $P(\mathbf{d}|h_i)$

Conditional independence Two variables X and Y are conditionally independent given a third variable Z if

$$P(X,Y|Z) = P(X|Z)P(Y|Z).$$

This means that $P(X|Y,Z) = P(X|Z)$ and $P(Y|X,Z) = P(Y|Z)$. This allows the joint probability distribution of $P(X,Y,Z)$ to be decomposed into $P(X|Z)P(Y|Z)P(Z)$.

This illustrates a commonly occurring pattern in which a single cause directly influences a number of effects, all of which are conditionally independent. Thus the full joint distribution can we written as

$$P(\mathsf{Cause}, \mathsf{Effect_1}, \ldots, \mathsf{Effect_n}) = \mathsf{P(Cause)} \prod_{\mathsf{i}} \mathsf{P(Effect_i|Cause)} \tag{21}$$

This is called the *naive Bayes* model.

Expectation-Maximization (EM) An algorithm for finding maximum likelihood estimates of parameters in probabilistic models, where the model depends on unobserved latent variables. In other words, it computes expected values of hidden variables for each example, then recomputes the parameters using the expected values as if they were the observed values. For K-means the expected value of the hidden variable is the cluster and the parameters are μ and Σ. For Baum-Welch, the expected values of the hidden variables are the probabilities of going from state i to state j at time t, $\gamma_{ij}(t)$ given that the model generated the complete visible sequence V^T, the parameters are the state transition values a_{ij} and the observation values b_{ij}. If y is the observed value and z is the hidden value then

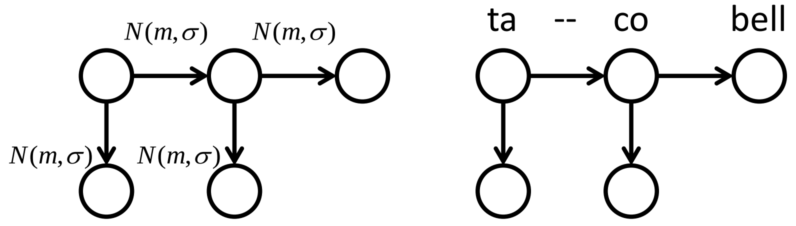
$$\theta_{n+1} = \arg\max_{\theta} \sum_{z} p(z|y,\theta_n) \log p(y,z|\theta)$$

In other words, θ_{n+1} is the value that maximizes (M) the expectation (E) of the complete data log-likelihood with respect to the conditional distribution of the latent data under the previous parameter value. This expectation is usually denoted as $Q(\theta)$:

$$Q(\theta) = \sum_{z} p(z|y,\theta_n) \log p(y,z|\theta)$$

EM is a generalization of Baum-Welch (Baum and Petrie, 1966).

Hidden Markov Models A couple of examples are shown above. The one on the right is used to natural language understanding of the phrae "Taco Bell."



Sequence Learning with a Bayesian Network Assume that the sequences come into level A and that level A feeds to level B and that level B feeds to level C in a hierarchy. There are three steps.

- Learn k most frequent sequences of inputs of length l at level A. Each sequence is assigned an index number.
- Each parent learns k most frequent incidences of children.
- Calculate conditional probabilities of children given parents, $P(A|B)$, and $P(B|C)$.

Then during inference take in a board state and do Bayesian belief propagation.

Hyperparameter A parameter to your prior.

Hyperprior Instead of using a single value for a given hyperparameter, one instead take a probability distribution on the hyperparameter itself; this is called a hyperprior. In principle, one may iterate this, calling parameters of a hyperprior hyperhyperparameters, and so forth.

18.1 Bayesian Networks

A Bayesian network is a directed acyclic graph that represents the conditional probability layout.

Variable elimination Computes one marginal probability in time proportional to the size of the biggest factor.

polytree A directed graph whose underlying undirected graph is acyclic (a tree).

Belief propagation If the network is a polytree then you can do exact inference at all nodes in time proportional to the size of the graph.

Loopy propagation If the network is not a polytree then you can do approximate inference at all nodes.

Other Variational methods. Join tree. Using EM to learn conditional probabilities. How is Gaussian special (own conjugate prior) but how do you do other distributions and why exactly don't they work?

18.2 Temporal Inference

Markov property A model has the Markov property if and only if knowledge of past model states does not help predict future model states [**?**]. Is Markov if everything you need to predict the future is in the current state. A random walk is Markov. A model is said to have the Markov property if future state is conditionally independent of past states and actions given the current state and action,

$$P(s_{t+1}|s_t, a_t, s_{t-1}, a_{t-1}, \ldots, s_0) = P(s_{t+1}|s_t, a_t)$$

Non-Markov A memory of some precept. Shaquille O'Neil remembering that his teammate was open and does a no-look pass.

Filtering The task of computing the belief state—the posterior distribution over the current state, given all evidence to date. (Russel and Norvig) Filtering computes $P(X_t|e_{1:t})$.

Prediction Computing the posterior distribution over the *future* state, given all evidence to date. $P(X_{t+k}|e_{1:t})$ where $k = 1$ usually.

Smoothing The task of computing the posterior distribution over a *past* state given all evidence up to present. $P(X_k|e_{1:t})$ for some k such that $0 \leq k < t$. E.g., the probability of it having rained last Wednesday.

Most Likely Explanation Given a sequence of observations, find the sequence of states that is most likely to have generated those observations. That is compute $\text{argmax}_{x_{1:t}} P(x_{1:t}|e_{1:t})$. Used in speech recognition.

Kalman filter A forward operator that works on Guassians. In a Kalman filter, the belief state is represented by a mean $\mu_\mathbf{t}$ and a covariance matrix Σ_t. Everything is a Gaussian, so all that is needed is to maintain a representation is the mean and the covariance for the prediction and correction phase. Specifically, a Kalman filter is specified by a linear transition model F, a transition noise covariance matrix Σ_x, a linear sensor model H, and a sensor noise covariance matrix Σ_z. Thus we have

$$\begin{aligned} P(\mathbf{x}_{t+1}|\mathbf{x}) &= N(F\mathbf{x_t}, \Sigma_x)(\mathbf{x_{t+1}}) \\ P(\mathbf{z_t}|\mathbf{x_t}) &= N(H\mathbf{x_t}, \Sigma_z)(\mathbf{z_t}) \end{aligned}$$

and given observation $\mathbf{z_t}$ the updates are given by

$$\begin{aligned} \mu_{\mathbf{t+1}} &= F\mu_{\mathbf{t}} + K_{t+1}(\mathbf{z_{t+1}} - HF\mu_{\mathbf{t}}) \\ \Sigma_{t+1} &= (I - K_{t+1})(F\Sigma_t F^T + \Sigma_x) \end{aligned}$$

where the Kalman gain K_t is given by:

$$K_{t+1} = (F\Sigma_t F^T + \Sigma_x)H^T(H(F\Sigma_t F^T + \Sigma_z)H^t + \Sigma_x)^{-1}$$

Particle filtering Used to perform approximate inference in dynamic Bayesian networks. First a population of N samples is created by sampling from that prior distribution at time 0, $\mathbf{P}(\mathbf{x_0})$. Then the update cycle is repeated for each time step.

- Each sample is propagated forward by sampling the next state value $\mathbf{x_{t+1}}$ given the current value for x_t for the sample, using the transition model $\mathbf{P}(\mathbf{x_{t+1}}|\mathbf{x_t})$.
- Each sample is weighted by the likelihood it assigns to the new evidence $P(\mathbf{e_{t+1}}|\mathbf{x_{t+1}})$
- The population is resampled (with replacement) to generate a new population of N samples. Each new sample is selected from the current population; the probability that a particular sample is selected is proportional to its weight. The new samples are unweighted.

18.3 Approximate Inference in Bayesian Networks

From [2].

Direct sampling Start at the top and go down generating samples based on the value of its parents. The the probability of a (partial) event is the number of sampled events that match the event divided by the total number of sampled events.

Rejection sampling Like direct sampling except you throw out all samples that do not accord with the evidence e. The problem with rejection sampling is that it may reject too many samples and not leave many left.

Likelihood weighting Avoids the inefficiency of rejection sampling by generating only events that are consistent with the evidence e. But during sampling the parents of e are free, but they should not be if e is fixed. So each event is weighted by the likelihood that the event accords to the evidence. The weight for a event is calculated as the product of the conditional probabilities of each evidence variable given its parents. So events in which the evidence e is unlikely are given less weight.

Markov chain Monte Carlo (MCMC) It does not generate events from scratch. Instead it generates an event by making a random change to the preceding event. At any time the network is in a particular current state. The next state is generated by randomly sampling a value for one of the non-evidence variables X_i, conditioned on the current values of the variables in the Markov blanket of X_i. So it wanders around the state space flipping one variable at a time and keeping the evidence variables fixed.

Gibbs sampler Variant of MCMC. If we sample a new value x_i' for X_i conditionally on all the other variables, including the evidence, we have

$$q(\mathbf{x} \to \mathbf{x}') = q((x_i, \bar{\mathbf{x}}_i \to (x_i', \bar{\mathbf{x}}_i)) = P(x_i'|\bar{\mathbf{x}}_i, \mathbf{e}).$$

The steps are

1. Pick a variable
2. Sample from Markov blanket

18.4 Learning the structure of Bayesian Networks

To learn the best network structures for data D: For each possible structure s calculate the optimal parameters θ_s and then calculate

$$P(s|D,\theta_s) = \frac{P(D|s,\theta_s)P(\theta|s)P(s)}{P(D,\theta_s)}$$

But you can't do this for every possible s so you use some heuristic.

The other approach is a constraint based approach, you find constraints of the form "A is not connected to B." This is justified whenever

$$P(A|C) = P(A|C,B)$$

Then build a graph that meets these criteria.

19 Linear Algebra

- Linear means additive combination of dimensions (or factors). For a mixture model, you can think of the dimensions as the things being mixed.
- Linear algebra is the study of these additive combinations of dimensions.
- A vector transformation T is linear if $T(ax) = aT(x)$ and $T(x_1 + x_2) = T(x_1) + T(x_2)$. That means nothing to me.
- Linear means the curvatire is independent of where you are. Something is linear if the partial derivative in each dimension is a constant. That means it makes a straight line; the value of the derivaitve (how much it curves) doesn't depend on any other dimension. Consider x^2. Its derivative is $2x$. How much curvature there is at any point depends on the value of x. Contrast this with $3x$, which has a derivative of 3. The amount of curvature is always the same at any point, thus a line. Also consider $4x_1x_2$, the paritial derivative relative to x_1 is $4x_2$. It depends on the value of x_2.
- Another dimension is another way to be similar to something. A grocery store is one dimensional. If the anchovies are by the canned fish, they are probably not the pizza toppings.
- Affine function, a function of a vector x of the form $f(x) = w_1x_1 + w_2x_2 + ... + w_nx_n + b$.
- Projection is just the shortest path to the line being projected.
- Covrariance matrix: is positive-semidefinite, which means all eigenvalues ≥ 0. I think all eigenvalues ≥ 0 in PCA because a covariance matrix is positive-semidefinite.[7]

19.1 Special Matricies

Jacobian is a square matrix of first-order partial derivatives of a vector function $F(x_1,\ldots,x_n) = y_1,\ldots,y_m$ where $F_i(x_1,\ldots,x_n) = y_i$. We have $J_{i,j} = \frac{\partial F_i}{\partial x_j}$, which means you get the derivative of component F_i with respect to x_j.

Hessian is a square matrix of second-order partial derivatives of a vector function $f(x_1,\ldots,x_n)$. We have $H_{i,j} = \frac{\partial f^2}{\partial x_i, \partial x_j}$, which means you take the partial derivative of x_i and then the partial derivative of x_j. It is the curvature matrix. For the i,j box, as you travel in i the Hessian tells you how the gradient in the direction of j changes.

[7] http://www.klab.caltech.edu/~harel/share/jh_linalg.pdf

19.2 Spaces

field A field is a set of elements that satisfied the field axioms for division and multiplication: associativity, communativity, distributivity, identity, inverse. And every nonzero element has a multiplicative inverse; e.g., x and $1/x$.

vector space A vector space is subset of fields where the elements are "vectors" with two binary operators that satisfy the vector space axions. Vectors in quotes because they can be anything. Basically a field where you can multiply by a scalar.

inner-product space is a subset of vector spaces where there is an inner product function where all pairs of vectors in the inner product are greater than or equal to 0. This inner product serves as a similarity function in kernel methods.

Hilbert space is a subset of inner project spaces that are separable and complete.

19.3 Matrix Algebra

- Bilinear means $x^T Ay$, so a vector times a matrix times a vector. I'm not sure what is so special about it. Result is a scalar.
- $AB = C$ with dimensions $[r \times m][m \times n] = [r \times n]$.
- pass

19.4 Eigenvectors

Consider the classic eigenvector equation $Ax_i = \lambda_i x_i$ where x_i is eigenvector i and λ_i is its eigenvalue for matrix A. We can think of the square matrix A as a linear transformation. Matrix A encodes a lot of little transformations in many dimensions, but the eigenvectors summarize these little transformations. For line through x_i, matrix A acts like the identity matrix.

"The eigenvectors point in the same direction before and after the transformation is applied, and they are the only vectors to do so."[8] The eigenvalues tell how much the eigenvectors scale during the transformation. A big eigenvalue means the eigenvector is important. In PCA, the covariance matrix is positive-semidefinite, so all of the eigenvalues are positive.

The eigenvectors form a basis. Eigendecomposition is a spectrum method like Fourier analysis.

19.5 Principal Components Analysis

"Principal components are particular linear combinations of the p random variables $X_1, X_2, \ldots, X_p$, with three important properties: (1) the principal components are uncorrelated, (2) the first principal component has the highest variance, the second principal component has the second highest variance, and so on, and (3) the total variation in all the principal components combined equal to the total variation in the original variables $X_1, X_2, \ldots, X_p$" [4].[9]

Much of the following explaination builds on the notation and explanation in [3].

- Let each observation x_t be a feature vector of dimension m.
- If you make each observation a column vector and you have n observations, then the matrix X of observations is of size $m \times n$.
- We want to represent this data in a lower-dimension space $r \ll m$ with minimal loss in a matrix Y of size $r \times n$
- To do this, we need a matrix P of size $r \times m$ so that $PX = Y$.
- The dimensions of $PX = Y$ are $[r \times m][m \times n] = [r \times n]$.

[8] This discussion is based on the web page http://isomorphismes.tumblr.com/post/4261903646/eigenvector.

[9] http://www.ime.unicamp.br/~andreani/matrizes/capitulo7.pdf is also a good reference.

- To compute P, we find the covariance matrix X_C of X. The covariance matrix is taken on the columns of X, where each column is an observation x_t of length m. Thus, X_C will be of size $m \times m$.
- The rows of P are the eigenvectors of X_C. The matrix P is composed of the r eigenvectors with the highest eigenvalues. Those r eigenvectors are the principle components.
- Covariance is relevant here because if two features have high covariance they don't both need to be used.
- Along the diagonal of X_C will be the simple variance. We are assuming that those with high variance are most important.

A succinct way to describe PCA is that it is an eigendecomposition of the covariance matrix created from the sample vectors.

Good reference `http://phd.gccis.rit.edu/discovery/proj4/PCA.pdf`.

19.6 In light of matrix factorization

- Let each observation x_t be a feature vector of dimension m.
- If you make each observation a row vector and you have n observations, then the matrix X of observations is of size $n \times m$.
- We want to represent this data in a lower-dimension space $r \ll m$ with minimal loss in a matrix Y of size $n \times r$.
- We need a matrix P such that $X = YP$ where the dimensions are $[n \times m] = [n \times r][r \times m]$
- trail off ...

19.6.1 Eigenfaces

1. Take n images of faces, each of size $u \times v$, and convert each face image into a face vector of size $m = uv$.
2. Put the n face vectors as columns in a matrix X of size $m \times n$.
3. Find the mean of each pixel value and create a mean vector μ of length m.
4. Subtract μ from each column of matrix X to create matrix $\hat{X}$.
5. Find the covariance matrix X_C of $\hat{X}$.
6. Find all of the eigenvectors of X_C; these are the eigenfaces.

20 MDP and Reinforcement Learning

As opposed to an MDP, in reinforcement learning you don't know what the rewards are for each state or even what all of the states are, and you don't know the transition function $T(s, a, s')$.

20.1 MDP

Policy evaluation For a given policy π calculate the value of each state $U(s)$.

Value iteration Used to calculate the optimal policy in an MDP. Calculate the maximum utility for each state, then use the state utilities to select an optimal action for each state. To calculate the utility for each state, it uses an iterative process using the Bellman equation beginning with each U having a random initial value with each iteration of the form:

$$U_{i+1}(s) = R(s) + \gamma \max_a \sum_{s'} T(s, a, s') U_i(s')$$

Policy iteration Another way to calculate an optimal policy in an MDP. It is a loop with two parts that begins with an initial policy π. First evaluate π. Since the policy is fixed can do a faster way then in value iteration, can just solve a set of linear equations. Then for each state look one ahead and see if there is a better action to take than the one dictated by π. Repeat until there is a loop in which the policy does not change.

20.2 Reinforcement Learning

There are two credit assignment problems.

Structural credit assignment How is reward distributed over similar world states. In environments with large state spaces the agent can't possibly visit every state. This allows the agent to generalize.

Temporal credit assignment This is the classic reinforcement learning problem. Solved by propagating the reward or punishment back.

There are two basic kinds of learners. Plays into the debate of how important knowledge is in artificial intelligence.

Utility-based Agent learns a utility function on states and uses it to select actions that maximize the expected outcome utility. It must have a model of its environment, $T(s, a, s')$, in order to make decisions.

Q-learning Agent learns action-value function (Q-function) giving the expected utility of taking a given action in a given state. Q-learners, are *model free*, meaning they don't need to know the effects of their actions represented by the function $T(s, a, s')$. However since they don't know the effects of their actions they can't look ahead, and this hinders learning.

There are two basic algorithms for learning. However learning can be active or passive. In active learning a given policy π is followed to choose actions. This policy allows the learner to use the method of solving linear equations to do policy evaluation. In active learning an action is chosen based on a combination of the expected utility and how often it has been chosen in the past. It tries to choose new actions to explore new parts of the state space.

Adaptive dynamic programming For a given policy π learns by iteratively creating a transition model $T(s, \pi(s), s')$ and then uses that transition model to perform *policy evaluation* to solve for the utility function $U(s)$. Since the policy is fixed (a passive learner has a fixed policy) policy evaluation can by accomplished by solving linear equations (pg. 625). I don't see where the dynamic programming comes in. Also, I don't see how to do this with active learning since policy evaluation requires a policy.

Temporal difference (TD) learning For each action it updates the utility function $U(s)$ for utility-based learning

$$U[s] = U[s] + \alpha(R(s) + \gamma U[s'] - U[s])$$

or the function $Q(a, s)$ for Q-learning

$$Q(a, s) = Q(a, s) + \alpha(R(s) + \gamma \max_{a'} Q(a', s') - Q(a, s))$$

Sometimes the space is too big to learn a table like value function U, this is the structural credit assignment problem. In such a case U has to be approximated as for example, $\hat{U}_\theta(x, y) = \theta_0 + \theta_1 x + \theta_2 y$. The challenge then is to determinate an appropriate hypothesis space and then to learn the parameters θ. It is important to note that by using $\hat{U}$ instead of U that the algorithm can then generalize to states it has never seen before.

Policy search just tries to improve the policy directly by changing the parameters that make up the policy function.

21 Systems

System (1) is an assemblage of elements comprising a whole with each element related to other elements. Any element which has no relationship with any other element of the system, cannot be a part of that system (wikipedia).

System (2) any assemblage which accepts an input, processes it, and produces an output. That is, an open system has an external interface in which matter, energy, information goes from outside to one or more internal element(s), which transduces this input via the original or other element(s) (e.g., by passing it among internal elements via internal interfaces), and an external interface through which results flow from some one or more internal element(s) to outside the system (wikipedia).

Dynamical System A system (definition 2) that changes over time. Thus a particular input could give a different output at different points in time. The output for a given input is determined by the *state* of the system.

Chaotic system A chaotic system can be deterministic but still can't be predicted. You just have to watch it play out. But little subsystems can be carved out that are predictive for a while. [xx I need to make sure I get a clean, real definition for this.]

Cybernetics A living system. A living system is one that self-regulates.

Semantics of a system then would be an specification of the output for each input. a_1 is a mathematical expression, a_2 is also a mathematical expression, then $c = a_1 + a_2$ if $a_1 + a_2 = c$. (Go back and look at programming language notes for this.) A semantics for a dynamical system would take the state into account. How does this jive with other definitions of "semantics"?

21.1 Dynamical Systems

Dynamical System A concept in mathematics where a fixed rule describes the time dependence of a point in a geometrical space (Wikipedia). A dynamical system has a state determined by a collection of real numbers. Small changes in the state of the system correspond to small changes in the numbers. The numbers are also the coordinates of a geometrical space-a manifold. The evolution rule of the dynamical system is a fixed rule that describes what future states follow from the current state. The rule is deterministic: for a given time interval only one future state follows from the current state.

Chaos Theory Simple nonlinear dynamical systems and even piecewise linear systems can exhibit a completely unpredictable behavior, which might seem to be random. (Remember that we are speaking of completely deterministic systems!). This unpredictable behavior has been called chaos.

Attractor In dynamical systems, an attractor is a set of states to which the system evolves after a long enough time. For the set to be an attractor trajectories that get close enough to the attractor must remain close even if slightly disturbed. Geometrically, an attractor can be a point, a curve, a manifold, or even a complicated set with fractal structures known as a strange attractor. A strange attractor moves around a point but never quite reaches it.

Predicting Non-Linear Systems You can't predict way ahead, you can only simulate forward. That's we we can't really predict the future in our environment. You can't abstract the system either becuase the butterfly effect means that small changes in the variable values means big changes in the future states.

22 Psychology

Classical conditioning Two things that occur together get associated together. Also called Pavlovian conditioning because the dogs who heard a tone as their food was served started to salivate even when only the tone was heard. So instead of the food triggering salivation, the tone took on that ability. Deals with involuntary behavior triggered by antecedents.

Operant conditioning Also called instrumental learning or instrumental conditioning, is the modification in behavior due to the consequences of the behavior. Reinforcement learning. Is voluntary as opposed to classical conditioning which is involuntary.

Object recognition People can recognize 10,000 objects

Vgotski Zone of proximal distance. We learn in that zone, and need a talk that is in that zone for maximal learning. He also says that internal speech is a tool.

22.1 Piaget

Assimilation Fitting new information into what you already know. You may have to warp the information to fit into your structure. The more you do this warping the less your structure fits reality.

Accommodation Changing your structure due to new information.

Equilibrium The combination of assimilation and accommodation necessary to internalize new information. You get info that doesn't quite fit, and then more that doesn't quite fit and your assimilation gets out of whack, you do an accommodation to restructure things and then you are back at equilibrium. This happens to me in computer science about every three months.

We learn everything in the framework of what we already know.

22.2 Consciousness

Stanislas Dehaene gave an Edge talk about consciousness. Consciousness is what is available globally to the brain. You have all of these parts of the brain that do parallel processing, and when something is important and they need to talk to one another, then it is available to consciousness. Making it available to consciousness allows them to talk to each other and settle on interpretations of input. Daniel Dennett: consciousness is "fame in the brain." Dehaene: consciousness is "global information in the brain." Bernard Baars: "global workspace."

Fits with my cat and bottle story. Walking in a parking lot I saw a cat. After a few steps I realized it was a bottle. But before, I had "seen" a cat. My brain jumpted from one interpretaton to another.

23 Vision

Compressive Sensing Take a set of random pixels from an image. Then try to fit shapes and colors over them such that you do this in the simplest way possible. If you have a triangle of green pixels, you fit a green triangle so all the pixels in between become green. You first try to fit big objects where they go, then get increasingly smaller. (This of course works with other kinds of signals as well.) So, instead of compression, you end up with decompression. Using computation, you take a sparse object and fill in detail.

More formally, compressive sensing uses a model of the form

$$y = Ax \tag{22}$$

where $x \in R^n$ and $y \in R^m$ with $m < n$. Matrix A is of the form $[m \times n]$. Vector y is the compressed version of the object x. What you do is you sense y (which is cheaper than sensing the bigger object x) and infer what x must have been. The system $y = Ax$ is underdetermined, so there are a lot of x's that match the equation (or close enough when you consider noise), and you choose to infer the x that is the most sparse.

This y you sense could be a from an MRI, where you don't want to leave the child in there very long, so you sense y and infer what x is.

HSV colorspace Hue, saturation, and value. Hue is the actual color, the wavelength of light. Saturation is the amount of white mixed in. My favorite color green has high saturation. And value does not have to do with the actual color of the object (it is not intrinsic to the surface as hue and saturation are), it is determined by the amount of light that hits it. If value is 0 then all colors are black. Imagine my green under moonlight, it would look grey because it would have low value.

Differential geometry The study of geometry using calculus. The apparatus of differential geometry is that of calculus on manifolds.

Gauss Map In differential geometry, the Gauss map maps a surface in Euclidean space R^3 to the unit sphere S^2. Namely, given a surface S lying in R^3, the Gauss map is a continuous map $N : S \to S^2$ such that $N(p)$ is orthogonal to S at p Each point on the surface has a normal. That is, a vector orthogonal to the surface at that point. Now, move this vector to the origin. Do this for all such vectors on the surface. What we get is a surface on the sphere (possibly with overlaps). This is called the Gauss map. A similar concept in 2 dimensions with curves is the radial of a curve..

Aspect Graph A structured graph of the set of aspects of an object, where the edges of the graph are the transitions between two neighboring stable views and a change between aspects is called a visual event.

Gabor filter A linear filter (what makes it linear?) whose impulse response is defined by a Gaussian multiplied by sin.

Visual Slam 1. Find features. 2. Get rough estimate of epipolar geometry between image pairs (odometry, joint position, least squares) 3. Remove noise (Ransac). At this point you have features and odometry. 4. Refine estimate of motion (any of 3 ways, Kalman filter, particle filter, third one?).

24 Digital Signal Processing

Mathematical transformations are applied to signals to obtain a further information from that signal that is not readily available in the raw signal.

Low-pass filtering Let the low frequency stuff pass through, filter out high frequency stuff. Smoothing operation, reduces noise and blurs images, and introduces delay in the time domain. Includes averaging summation and convolution with a Gaussian.

High-pass filtering Filter out low frequency stuff, let high frequency stuff pass through. Edge detection or sharpening, is sharpens edges and increases noise. Operations include differentiation, subtraction, and convolution with the second derivative of the Gaussian.

Fourier Transform (FT) The most popular transform. Gives the frequency-amplitude representation of the signal. For 50 Hz electricity there is only one spike at 50, but usually the signal is more complicated than that. FT gives only what frequency components exist in the signal, if the signal is not stationary then information is lost.

Stationary signal Signals whose frequency does not change in time. Signals that always have the same frequency, like a sine wave.

Short Time Fourier Transform (STFT) Provides a time-frequency transform.

Wavelet Transform Provides a time-frequency representation, developed to overcome some problems with STFT.

25 Information Theory

Taken from [5]. The amount of information in a message is measured relative to what the receiver already knows, there is no absolute measure of information. So if you tell someone that it is about to rain and that person already knows that, then you have conveyed no information.

Information Something that reduces uncertainty among alternatives. The amount of information I in a message m, denoted $I(m)$, in inversely related to the probability of that message from the receivers point of view, denoted P_m,

$$I(m) = -\log(P_m).$$

Negentropy (entropy) Entropy is simply the number of different ways a bunch of particles can arrange themselves. Consider a deck of cards, unordered cards can be in a lot of states and still be unordered (many different ways to arrange the particles) and so high entropy, but ordered cards can only be in one state (low entropy).

The entropy of a distribution of any variable X with a probability distribution $P(x)$ is given by

$$H(X) = -\int P(x)\ln P(x)\mathrm{d}x$$

The normal distribution has the maximum entropy of all distributions having a given mean and variance.

We can calculate the expected amount of information the receiver expects to receive by summing over all the possible messages $m \in \mathcal{M}$ and multiplying each by the probability of its occurrence, P_m. This is called the *negentropy* (entropy) of the message source.

$$\begin{aligned}\text{negentropy} &= \sum_{m\in\mathcal{M}} P_m(-\log(P_m)) \\ &= -\sum_{m\in\mathcal{M}} P_m\log(P_m)\end{aligned}$$

Mutual Information Is a measure the the total uncertainty-reducing potential of X on T and is given by $I(T,X) = H(T) - H(T|X)$. If the entropy of a variable T is given by

$$H(T) = -\sum_t P(t)\log_2 P(t)$$

then the entropy of a variable T given a value $X = x$ is given by

$$H(T|x) = -\sum_t P(t|x)\log_2 P(t|x)$$

then

$$\begin{aligned}
H(T|X) &= \sum_x H(T|x)P(x) \\
&= -\sum_x \sum_t P(t,x)\log_2 P(t|x)
\end{aligned}$$

Thus we have

$$\begin{aligned}
I(T,X) &= H(T) - H(T|X) \\
&= -\sum_t P(t)\log_2 P(t) + \sum_x \sum_t P(t,x)\log_2 P(t|x) \\
&= -\sum_t \log_2 P(t) \sum_x P(t,x) + \sum_x \sum_t P(t,x)\log_2 P(t|x) \\
&= \sum_x \sum_t P(t,x)\log_2 P(t|x) - \log_2 P(t)P(t,x) \\
&= \sum_x \sum_t P(t,x)\log_2 \frac{P(t|x)}{P(t)} \\
&= \sum_x \sum_t P(t,x)\log_2 \frac{P(t,x)}{P(t)P(x)}
\end{aligned}$$

It is symmetric so $I(T,X) = I(X,T)$.

Kullback-Leibler distance Also known as the KL divergence or the *relative entropy* is a measure of the "distance" between two distributions. If we have two distributions p and q over a variable x then

$$D_{KL}(p(x), q(x)) = \sum_x q(x)\ln\frac{q(x)}{p(x)}$$

Another way to look at mutual information is as the relative entropy between the joint distribution and the product distribution $I(X,Y) = D_{KL}(p(x,y), p(x)p(y))$. The relative entropy is 0 if X and Y are independent. How does this relate to the covariance?

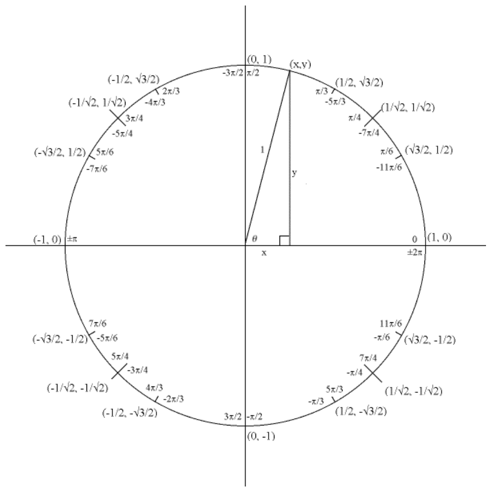
Using binary Using binary, the number of bits in the minimal message is the information content of the message.

Minimum Description Length Principle Takes into account the number of bits needed to specify the hypothesis and the number of bits needed to specify the data given the hypothesis.

$$h_{MDL} = \text{argmin}_{h\in H}\, L_{C_1}(h) + L_{C_2}(D|h)$$

where $L_{C_k}(m)$ is the number of bits needed to encode m using code C_k. Since the code that minimizes the expected message length assigns $-\log_2 p_i$ bits to encode a message with probability p_i, the h_{MDL} hypothesis is equal to the h_{MAP} hypothesis

$$\begin{aligned}
h_{MAP} &= \underset{h\in H}{\text{argmax}}\ \log_2 P(D|h) + \log_2 P(h) \\
&= \underset{h\in H}{\text{argmin}}\ -\log_2 P(D|h) - \log_2 P(h)
\end{aligned}$$



$$\begin{tikzcd}
A \arrow[r] \arrow[d, "\pi(A)"'] & A' \arrow[d, "\pi(A')"] \\
B \arrow[r] & B'
\end{tikzcd}$$

Figure 1: (a) Unit Circle. (b) Diagram of abstraction. The raw input is A and the abstracted representation is B. An abstraction π is valid if both the abstraction of A' and the transition of B lead to B'.

26 Induction

Mathematical induction Used to prove a property P of the natural numbers. To show $P(n)\ \forall n \in I\!N$, must show the *base case* $P(0)$, then using the *induction hypothesis* $P(m)$, the *induction step* that $P(m) \Rightarrow P(m+1)$ must be shown.

Strong induction A variant of mathematical induction in which the induction hypothesis instead of being $P(m)$ is $\forall i,\ 0 \leq i \leq m, P(m)$. This is a strong assumption, so anything provable with regular mathematical induction is provable with strong induction. But you can create $Q(n)$ which means $P(m),\ m \leq n$ to do a strong induction using regular mathematical induction. With strong induction there is no need for a base case.

Structural induction (also called complete induction) Used to show $P(x)$ where x is in some recursively defined structure. To show this, first must prove that P holds for all minimal structures (those not defined in terms of other structures). Then must show that if P holds for all substructures of a structure S then P also holds for S.

Well-founded induction Mathematical and structural induction are both special cases of this. To show a property holds for all expressions, it is sufficient to show that the property holds for an arbitrary expression if it holds for all it subexpressions. Start by showing that the property holds for the expressions with no subexpressions, then show that for each type of transition that the property is preserved. Must be using a *well-founded relation* on a set A, the relation is well-founded if any $Q \subseteq A$ has a minimal element.

Induction on derivations Sometimes structural induction cannot be used because something is defined in terms of itself (like a while loop). The proof is by well-founded induction on the derivation relation. In this case to prove $P(x)\ \forall x$ must show P for all rules of the form $(/y)$ and show that if $P(x_1), P(x_2), ...P(x_n)$ for all rules of the form $(x_1, x_2, ..., x_n/y)$ then $P(y)$.

Rule induction Looks just like induction on derivations. Let I_R be defined by rule instances R. Let P be a property. Then $\forall x \in I_R.\ P(x)$ iff for all rule instances (X/y) in R for which $X \subseteq I_R$ we have $(\forall x \in X.\ P(x)) \Rightarrow P(y)$.

27 Statistical and Bayesian Hypothesis Testing Example

We have a rule r that can predict the future or not. If it correctly predicts the future, it is successful. If we let θ be the probability of success for r and n be the number of trials, and x the number of times successful, then we have a binomial distribution

$$f(X = x|\theta, n) = \binom{n}{x}\theta^x(1-\theta)^{n-x} \tag{23}$$

and the likelihood of θ is

$$L(\theta) = \binom{n}{x}\theta^x(1-\theta)^{n-x} \tag{24}$$

We want to find the best estimate $\hat{\theta}$ of the probability θ of success of r. We do this by differentiating L and finding where $L(\theta) = 0$. We take the log because it is easier to work with (we can do this by that "not crossing over"theorem)

$$\log L(\theta) = \binom{n}{x} + x\log\theta + n - x\log(1-\theta) \tag{25}$$

We then differentiate

$$\frac{\partial \log L}{\partial \theta} = x\frac{1}{\theta} + (n-x)\frac{-1}{1-\theta} \tag{26}$$

and set to 0

$$\begin{aligned} \frac{x}{\theta} - (n-x)\theta &= 0 \quad (27)\\ x - n\theta &= 0 \quad (28) \end{aligned}$$

so that

$$\hat{\theta} = \frac{x}{n} \tag{29}$$

is the maximum likelihood estimate of θ.

The distribution of x is a binomial, but it is approximately normal if n is large. To find the confidence of x/n we use a normal approximately with $\mu = x$ and $\sigma^2 = n\hat{\theta}(1-\hat{\theta})$. We can then find the confidence that x/n is greater than a threshold using the standard method with normal distributions. That is normalize and look it up in a table. If n is small then we have to use the binomial distribution.

If we want to compare the reliability of two rules r_1 and r_2 then we create a new variable Y where

$$Y = \frac{x_1}{n_1} - \frac{x_2}{n_2} \tag{30}$$

If n_1 and n_2 are large then we can subtract two normal distributions to get another normal distribution for Y. Otherwise, we have to look at joint binomial distribution, which gets pretty messy.

The Bayesian method does not calculate the point estimate $\hat{\theta}$ but rather the distribution $P(\theta|X)$ where X is the data.

$$P(\theta|X) = \frac{P(\theta, X)}{P(X)} = \frac{P(X|\theta)P(\theta)}{P(X)} \propto P(X|\theta)P(\theta) \tag{31}$$

where $P(\theta|X)$ is the *posterior*, $P(X|\theta)$ is the *likelihood*, and $P(\theta)$ is the *prior*. The probability of the data $P(X)$ is hard to calculate, but if we use a *conjugate prior* then it is simple. A *conjugate prior* is a prior where the posterior is in the same family. The likelihood $P(X|\theta)$ is

$$P(X|\theta) = \binom{n}{x}\theta^x(1-\theta)^{n-x} \tag{32}$$

and if we use the uniform distribution that gives a value 1 (note: I think that it would be less than 1 but still an constant) then

$$P(\theta|X) \propto \binom{n}{x}\theta^x(1-\theta)^{n-x}\cdot 1 \tag{33}$$

and since $\binom{n}{x}$ is a constant we have the beta distribution $Beta(x+1, n-x+1)$. Then to find the interval that $P(\theta|X)$ is greater than a threshold we use the cumulative distribution of $Beta(x+1, n-x+1)$.

For the case with two rules r_1 and r_2 where we want to determine if one is more reliable than the other, we simulate using the beta distribution for each.

27.1 Note on student-t distribution

The student-t distribution is only used when your underlying distribution is normal. It is used when you don't have enough data to estimate the true normal distribution. So you take your estimate of μ and σ and use the student-t, which has thicker tails. The thicker tails make up for lack of data. If you have more than 30 data points you can use the normal distribution.

The case described here uses binomial data, this means that the student-t distribution is not applicable.

References

[1] M. Richardson and P. Domingos. Markov logic networks. *Machine Learning*, 62(1):107–136, 2006.

[2] S.J. Russell and P. Norvig. *Artificial intelligence: a modern approach*. Prentice hall, 2003.

[3] J. Shlens. A tutorial on principal component analysis. *Systems Neurobiology Laboratory, University of California at San Diego*, 2005.

[4] Mei-Ling Shyu, Shu-Ching Chen, Kanoksri Sarinnapakorn, and LiWu Chang. A novel anomaly detection scheme based on principal component classifier. Technical report, DTIC Document, 2003.

[5] Chris Thornton. *Truth from trash: how learning makes sense*. MIT Press, 2002.