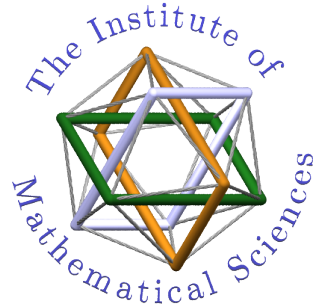
Sequences and Information

Rahul Siddharthan

The Institute of Mathematical Sciences, Chennai, India
http://www.imsc.res.in/~rsidd/

Facets'16, 04/07/2016



This box says something

By looking at the symbols here, you get a message

This box says something

By looking at the symbols here, you get a message

This box says nothing

THHHHHHTHHTHHHTTTHHHHHTHHTTHHTTHHTTHTTTHTHTTTTTT
HTTHTTTTTTTTHHHHHTHHTTTHHTHTTTTHTTHHHTTHHHTHTTHH
THHTTTTHHHTTTHHTTHHHHTTHTTHTTTHTTTTHTHHHHHTTTHHT
THHTTHHHTHTHHTTHHHHTTTHTHTTTHTHTTHTTHTTTTTTTHHHT

A nonrandom sequence

Three “random” sequences

Which is in fact random?

Example 1

THHHHHHTHHTHHHTTTHHHHHTHHTTHHTTHHTTHTTTHTHTTTTTT
HTTHTTTTTTTTHHHHHTHHTTTHHTHTTTTHTTHHHTTHHHTHTTHH
THHTTTTHHHTTTHHTTHHHHTTHTTHTTTHTTTTHTHHHHHTTTHHT
THHTTHHHTHTHHTTHHHHTTTHTHTTTHTHTTHTTHTTTTTTTHHHT

Example 2

HTTHTHHHTHTTHTTHHTHHTHTTHTTTHHHTHTTHHHHTHTTHTHHT
TTHTTTHTHHTTHTHTTHHTHTHHTHTHTTHTHTTHTTHTHHTTHTHT
HHTTTHTHTTTHTTHTHHHTHTTTHTHTHHHHTHHHTHTTTHTHHHTH
TTHTTTHTHHHTHTTHTHHTHTTHTHTHHHHTHHTTTTHTHHHTHTHH

Example 3

TTHHTHTTHTHTTHTHTTHTTHTHHTHTHTTHTHTHTHHTHHTHTTHT
HHTHTHTTHTHTHTTHHTHHTHHTHHTTHHHTTHTHHTHTTTTHHTHH
TTTHTHTHTHTTHHHTHTTHTHHTHTHTTHTHHTHHTHTTHTHTHHTT
HHTTHTHHTTTTHTTHHTHTTHTHTHTHTHHTHHTHTTTHTTHTHHTH

Three "random" sequences

Answer:

Example 1 (made with random number generator)

THHHHHHTHHTHHHTTTHHHHHTHHTTHHTTHHTTHTTTHTHTTTTTT
HTTHTTTTTTTTHHHHHTHHTTTHHTHTTTTHTTHHHTTHHHTHTTHH
THHTTTTHHHTTTHHTTHHHHTTHTTHTTTHTTTTHTHHHHHTTTHHT
THHTTHHHTHTHHTTHHHHTTTHTHTTTHTHTTHTTHTTTTTTTHHHT

Example 2 (made by hand)

HTTHTHHHTHTTHTTHHTHHTHTTHTTTHHHTHTTHHHHTHTTHTHHT
TTHTTTHTHHTTHTHTTHHTHTHHTHTHTTHTHTTHTTHTHHTTHTHT
HHTTTHTHTTTHTTHTHHHTHTTTHTHTHHHHTHHHTHTTTHTHHHTH
TTHTTTHTHHHTHTTHTHHTHTTHTHTHHHHTHHTTTTHTHHHTHTHH

Example 3 (made from English text)

TTHHTHTTHTHTTHTHTTHTTHTHHTHTHTTHTHTHTHHTHHTHTTHT
HHTHTHTTHTHTHTTHHTHHTHHTHHTTHHHTTHTHHTHTTTTHHTHH
TTTHTHTHTHTTHHHTHTTHTHHTHTHTTHTHHTHHTHTTHTHTHHTT
HHTTHTHHTTTTHTTHHTHTTHTHTHTHTHHTHHTHTTTHTTHTHHTH

Communication

- How do you communicate meaning via symbols?
- And knowing that a sequence of symbols has meaning, how do you extract that meaning? (A much harder problem!)

Reprinted with corrections from *The Bell System Technical Journal*,
Vol. 27, pp. 379–423, 623–656, July, October, 1948.

A Mathematical Theory of Communication

By C. E. SHANNON

INTRODUCTION

THE recent development of various methods of modulation such as PCM and PPM which exchange bandwidth for signal-to-noise ratio has intensified the interest in a general theory of communication. A basis for such a theory is contained in the important papers of Nyquist[1] and Hartley[2] on this subject. In the present paper we will extend the theory to include a number of new factors, in particular the effect of noise in the channel, and the savings possible due to the statistical structure of the original message and due to the nature of the final destination of the information.

The fundamental problem of communication is that of reproducing at one point either exactly or approximately a message selected at another point. Frequently the messages have *meaning*; that is they refer to or are correlated according to some system with certain physical or conceptual entities. These semantic

Shannon and information theory

How much information does a message contain

Answer:
If the message is selected from N possible, equally likely messages, then N or any monotonic function of N is a measure of the "information content" of the message.

Shannon and information theory

How much information does a message contain

Answer:
If the message is selected from N possible, equally likely messages, then N or any monotonic function of N is a measure of the "information content" of the message.

What function of N?

Answer: $\log N$, and in particular, $\log_2 N$.
Why? Because $\log N$ tends to vary linearly with engineering parameters
Example: if there are n bits in a word, the number of messages that word can represent is $N = 2^n$, so $\log_2 N = n$.

Shannon and information theory

How much information does a message contain

Answer:
If the message is selected from N possible, equally likely messages, then N or any monotonic function of N is a measure of the "information content" of the message.

What function of N?

Answer: $\log N$, and in particular, $\log_2 N$.
Why? Because $\log N$ tends to vary linearly with engineering parameters
Example: if there are n bits in a word, the number of messages that word can represent is $N = 2^n$, so $\log_2 N = n$.

Also:

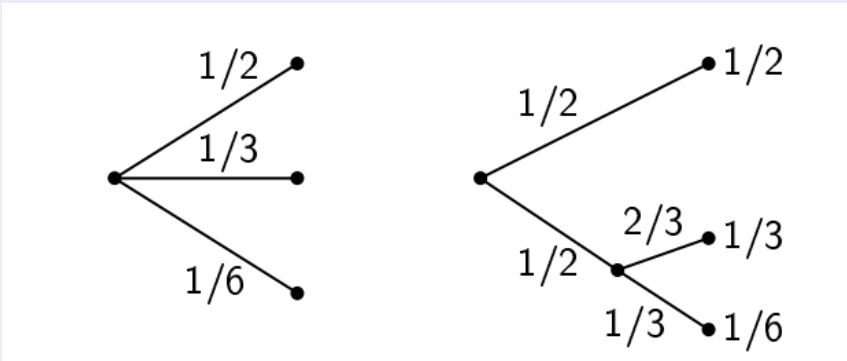
More "intuitive"; mathematically more convenient

Shannon and information theory

Consider a process that produces symbols drawn from an alphabet of n letters. These letters don't occur equally often but have "frequencies" (probabilities) $p_1, p_2 \ldots p_n$.
How much information is being produced with each letter? How do we define an "information score" $H(p_1, p_2 \ldots p_n)$?

Desirable characteristics of H

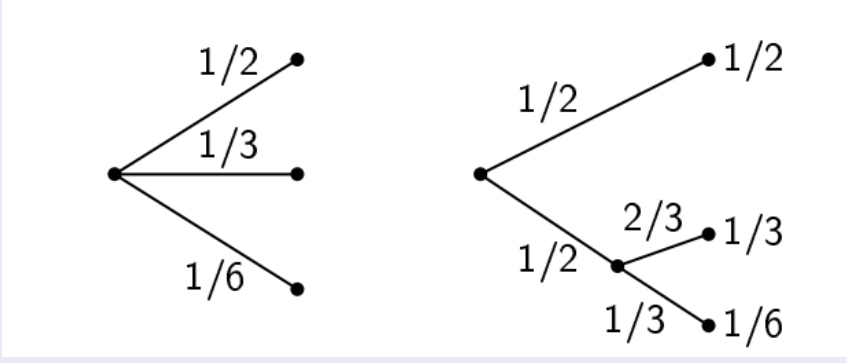
- Continuity
- If all the p's are equal ($p_i = 1/n \; \forall \; i$) H should increase monotonically with n.
- If a choice can be broken down into two choices, the full H should be the weighted sum of the individual values. Eg here



we want $H(\frac{1}{2}, \frac{1}{3}, \frac{1}{6}) = H(\frac{1}{2}) + \frac{1}{2}H(\frac{2}{3}, \frac{1}{3})$.

Desirable characteristics of H

- Continuity
- If all the p's are equal ($p_i = 1/n \; \forall \; i$) H should increase monotonically with n.
- If a choice can be broken down into two choices, the full H should be the weighted sum of the individual values. Eg here



we want $H(\frac{1}{2}, \frac{1}{3}, \frac{1}{6}) = H(\frac{1}{2}) + \frac{1}{2}H(\frac{2}{3}, \frac{1}{3})$.

Only possible solution

$H(p_1, p_2 \ldots p_n) = -K \sum_{i=1}^{n} p_i \log p_i$

"Entropy" of probabilities

$H(p_1, p_2 \ldots p_n) = -K \sum_{i=1}^{n} p_i \log p_i$

Conventional choice in information theory:

$H(p_1, p_2 \ldots p_n) = -\sum_{i=1}^{n} p_i \log_2 p_i$

"Entropy" of probabilities

$H(p_1, p_2 \ldots p_n) = -K \sum_{i=1}^{n} p_i \log p_i$

Conventional choice in information theory:

$H(p_1, p_2 \ldots p_n) = -\sum_{i=1}^{n} p_i \log_2 p_i$

"Entropy" of probabilities

$H(p_1, p_2 \dots p_n) = -K \sum_{i=1}^{n} p_i \log p_i$

Conventional choice in information theory:

$H(p_1, p_2 \dots p_n) = -\sum_{i=1}^{n} p_i \log_2 p_i$

Properties

- $H = 0$ iff one p is 1 and all other p's are 0.
- H is maximum if all p's are equal. Then $H = \log_2 n$.
-

What is a probability?

We have used the “probability” but how do we define it and how do we calculate it?

What is a probability?

We have used the "probability" but how do we define it and how do we calculate it?

Simplest example: Bernoulli process

Suppose you have only two possible outcomes, say S and F. Then say $P(S) = p$ and $P(F) = q = 1 - p$.
For this process, $H = -p \log_2 p - q log_2 q$. But how do you learn p and q?

Definitions

- Joint probability $P(x, y)$ of two events: probability of both events occurring
- Independent events: the probability of one event is not influenced by the outcome of the other
  $P(x, y) = P(x)P(y)$
- Probability distribution: the set of values p_i that define probabilities of outcomes
- Bernoulli process: *independent, identically distributed* (i.i.d.) events each with two possible outcomes (eg, coin tosses)
- Multinomial process: i.i.d events each with $n > 2$ possible outcomes

Which sequence comes from a Bernoulli process?

Example 1 (made with random number generator)

THHHHHHTHHTHHHTTTHHHHHTHHTTHHTTHHTTHTTTHTHTTTTTT
HTTHTTTTTTTTHHHHHTHHTTTHHTHTTTTHTTHHHTTHHHTHTTHH
THHTTTTHHHTTTHHTTHHHHTTHTTHTTTHTTTTHTHHHHHTTTHHT
THHTTHHHTHTHHTTHHHHTTTHTHTTTHTHTTHTTHTTTTTTTHHHT

Example 2 (made by hand)

HTTHTHHHTHTTHTTHHTHHTHTTHTTTHHHTHTTHHHHTHTTHTHHT
TTHTTTHTHHTTHTHTTHHTHTHHTHTHTTHTHTTHTTHTHHTTHTHT
HHTTTHTHTTTHTTHTHHHTHTTTHTHTHHHHTHHHTHTTTHTHHHTH
TTHTTTHTHHHTHTTHTHHTHTTHTHTHHHHTHHTTTTHTHHHTHTHH

Example 3 (made from English text)

TTHHTHTTHTHTTHTHTTHTTHTHHTHTHTTHTHTHTHHTHHTHTTHT
HHTHTHTTHTHTHTTHHTHHTHHTHHTTHHHTTHTHHTHTTTTHHTHH
TTTHTHTHTHTTHHHTHTTHTHHTHTHTTHTHHTHHTHTTHTHTHHTT
HHTTHTHHTTTTHTTHHTHTTHTHTHTHTHHTHHTHTTTHTTHTHHTH

Which sequence comes from a Bernoulli process?

Example 1 (made with random number generator)

THHHHHHTHHTHHHTTTHHHHHTHHTTHHTTHHTTHTTTHTHTTTTTT
HTTHTTTTTTTTHHHHHTHHTTTHHTHTTTTHTTHHHTTHHHTHTTHH
THHTTTTHHHTTTHHTTHHHHTTHTTHTTTHTTTTHTHHHHHTTTHHT
THHTTHHHTHTHHTTHHHHTTTHTHTTTHTHTTHTTHTTTTTTHHHT

Example 2 (made by hand)

HTTHTHHHTHTTHTTHHTHHTHTTHTTTHHHTHTTHHHHTHTTHTHHT
TTHTTTHTHHTTHTHTTHHTHTHHTHTHTTHTHTTHTTHTHHTTHTHT
HHTTTHTHTTTHTTHTHHHTHTTTHTHTHHHHTHHHTHTTTHTHHHTH
TTHTTTHTHHHTHTTHTHHTHTTHTHTHHHHTHHTTTTHTHHHTHTHH

Example 3 (made from English text)

TTHHTHTTHTHTTHTHTTHTTHTHHTHTHTTHTHTHTHHTHHTHTTHT
HHTHTHTTHTHTHTTHHTHHTHHTHHTTHHHTTHTHHTHTTTTHHTHH
TTTHTHTHTHTTHHHTHTTHTHHTHTHTTHTHHTHHTHTTHTHTHHTT
HHTTHTHHTTTTHTTHHTHTTHTHTHTHTHHTHHTHTTTHTTHTHHTH

Calculate $P(x)$ for the two symbols $x =$ H, T. Also calculate $P(x|y)$ where y is X's predecessor. For a Bernoulli process, $P(x|y) = P(x)$.

Calculating $P(x)$ for a Bernoulli process

Suppose you have two outcomes, S and F, with $P(S) = p$.
Suppose you have *no idea* what the value of p is. It can be anything between 0 and 1.
Suppose you have N observations, with S occurring n times.
What is the probability of S next time?

Calculating $P(x)$ for a Bernoulli process

Suppose you have two outcomes, S and F, with $P(S) = p$.
Suppose you have *no idea* what the value of p is. It can be anything between 0 and 1.
Suppose you have N observations, with S occurring n times.
What is the probability of S next time?
Answer: $P(S) = \frac{n+1}{N+2}$

Calculating $P(x)$ for a Bernoulli process

Suppose you have two outcomes, S and F, with $P(S) = p$.
Suppose you have *no idea* what the value of p is. It can be anything between 0 and 1.
Suppose you have N observations, with S occurring n times.
What is the probability of S next time?
Answer: $P(S) = \frac{n+1}{N+2}$

Proof in outline

$D =$ given data (N trials, n S's)

Calculating $P(x)$ for a Bernoulli process

Suppose you have two outcomes, S and F, with $P(S) = p$.
Suppose you have *no idea* what the value of p is. It can be anything between 0 and 1.
Suppose you have N observations, with S occurring n times.
What is the probability of S next time?
Answer: $P(S) = \frac{n+1}{N+2}$

Proof in outline

$D =$ given data (N trials, n S's)
$P(S|D) = \frac{P(S,D)}{P(D)}$

Calculating $P(x)$ for a Bernoulli process

Suppose you have two outcomes, S and F, with $P(S) = p$.
Suppose you have *no idea* what the value of p is. It can be anything between 0 and 1.
Suppose you have N observations, with S occurring n times.
What is the probability of S next time?
Answer: $P(S) = \frac{n+1}{N+2}$

Proof in outline

$D =$ given data (N trials, n S's)

$P(S|D) = \frac{P(S,D)}{P(D)}$

$P(D) = \binom{N}{n} p^n (1-p)^{N-n}$

Calculating $P(x)$ for a Bernoulli process

Suppose you have two outcomes, S and F, with $P(S) = p$.
Suppose you have *no idea* what the value of p is. It can be anything between 0 and 1.
Suppose you have N observations, with S occurring n times.
What is the probability of S next time?
Answer: $P(S) = \frac{n+1}{N+2}$

Proof in outline

$D =$ given data (N trials, n S's)

$P(S|D) = \frac{P(S,D)}{P(D)}$

$P(D) = \binom{N}{n} p^n (1-p)^{N-n}$

$P(S,D) = \binom{N}{n} p^n + 1(1-p)^{N-n}$

Calculating $P(x)$ for a Bernoulli process

Suppose you have two outcomes, S and F, with $P(S) = p$.
Suppose you have *no idea* what the value of p is. It can be anything between 0 and 1.
Suppose you have N observations, with S occurring n times.
What is the probability of S next time?
Answer: $P(S) = \frac{n+1}{N+2}$

Proof in outline

$D =$ given data (N trials, n S's)

$P(S|D) = \frac{P(S,D)}{P(D)}$

$P(D) = \binom{N}{n} p^n (1-p)^{N-n}$

$P(S,D) = \binom{N}{n} p^n + 1(1-p)^{N-n}$

But since we don't know p, both of these have to be integrated over p from 0 to 1 with uniform weight.
These are "beta function" integrals and can be done exactly.

Beta prior, pseudocounts

Suppose you have two outcomes, S and F, with $P(S) = p$.
Suppose you have *some idea* what the value of p is. It has the specific form $P(p) \propto p^{c_1-1}(1-p)^{c_2-1}$, where c_1 and c_2 are constants.
Suppose you have N observations, with S occurring n times.
What is the probability of S next time?

Beta prior, pseudocounts

Suppose you have two outcomes, S and F, with $P(S) = p$.
Suppose you have *some idea* what the value of p is. It has the specific form $P(p) \propto p^{c_1-1}(1-p)^{c_2-1}$, where c_1 and c_2 are constants.
Suppose you have N observations, with S occurring n times.
What is the probability of S next time?
Answer: $P(S) = \frac{n+c_1}{N+c_1+c_2}$
c_1 and c_2 are often called "pseudocounts".

Beta prior, pseudocounts

Suppose you have two outcomes, S and F, with $P(S) = p$.
Suppose you have *some idea* what the value of p is. It has the specific form $P(p) \propto p^{c_1-1}(1-p)^{c_2-1}$, where c_1 and c_2 are constants.
Suppose you have N observations, with S occurring n times.
What is the probability of S next time?
Answer: $P(S) = \frac{n+c_1}{N+c_1+c_2}$
c_1 and c_2 are often called "pseudocounts".

Multinomial generalization

If you have k possible outcomes, and N observations with n_i occurrences of the i'th outcome, then
$P(j) = \frac{n_j+c_j}{N+C}$
where $C = \sum_{i=1}^{k} c_i$.

Bigrams, n-grams, conditional probabilities

If x and y each have k possible outcomes, to calculate $P(x|y)$ we can use
$P(x|y) = P(x,y)/P(y)$
and calculate $P(y)$ as above, and $P(x,y)$ by treating each "bi-gram" as a single unit with k^2 possible outcomes.

Factorizing joint probabilities, Markov processes

Suppose you have a sequence of letters of length L,
$S = S_1 S_2 S_3 \ldots S_L$
with each letter S_i drawn from the same alphabet, but not i.i.d. Then we can write
$P(S) = P(S_1)P(S_2|S_1)P(S_3|S_1S_2)] \ldots P(S_L|S_1S_2 \ldots S_{L-1})$.
If we assume that each letter depends on only its predecessor, we have a (first-order) *Markov model*
$P(S) = P(S_1)P(S_2|S_1)P(S_3|S_2)] \ldots P(S_L|S_{L-1})$.
If we assume that each letter depends on n predecessors, we have an n'th order Markov model.

Example: Shannon, 1948

(C. E. Shannon, *A mathematical theory of communication*, 1948)

3. The Series of Approximations to English

To give a visual idea of how this series of processes approaches a language, typical sequences in the approximations to English have been constructed and are given below. In all cases we have assumed a 27-symbol "alphabet," the 26 letters and a space.

1. Zero-order approximation (symbols independent and equiprobable).

   XFOML RXKHRJFFJUJ ZLPWCFWKCYJ FFJEYVKCQSGHYD QPAAMKBZAACIBZL-HJQD.

2. First-order approximation (symbols independent but with frequencies of English text).

   OCRO HLI RGWR NMIELWIS EU LL NBNESEBYA TH EEI ALHENHTTPA OOBTTVA NAH BRL.

3. Second-order approximation (digram structure as in English).

   ON IE ANTSOUTINYS ARE T INCTORE ST BE S DEAMY ACHIN D ILONASIVE TU-COOWE AT TEASONARE FUSO TIZIN ANDY TOBE SEACE CTISBE.

4. Third-order approximation (trigram structure as in English).

   IN NO IST LAT WHEY CRATICT FROURE BIRS GROCID PONDENOME OF DEMONS-TURES OF THE REPTAGIN IS REGOACTIONA OF CRE.

Example: Shannon, 1948

(C. E. Shannon, *A mathematical theory of communication*, 1948)

5. First-order word approximation. Rather than continue with tetragram, . . . , n-gram structure it is easier and better to jump at this point to word units. Here words are chosen independently but with their appropriate frequencies.

   REPRESENTING AND SPEEDILY IS AN GOOD APT OR COME CAN DIFFERENT NATURAL HERE HE THE A IN CAME THE TO OF TO EXPERT GRAY COME TO FURNISHES THE LINE MESSAGE HAD BE THESE.

6. Second-order word approximation. The word transition probabilities are correct but no further structure is included.

   THE HEAD AND IN FRONTAL ATTACK ON AN ENGLISH WRITER THAT THE CHARACTER OF THIS POINT IS THEREFORE ANOTHER METHOD FOR THE LETTERS THAT THE TIME OF WHO EVER TOLD THE PROBLEM FOR AN UNEXPECTED.

The resemblance to ordinary English text increases quite noticeably at each of the above steps. Note that these samples have reasonably good structure out to about twice the range that is taken into account in their construction. Thus in (3) the statistical process insures reasonable text for two-letter sequences, but four-letter sequences from the sample can usually be fitted into good sentences. In (6) sequences of four or more words can easily be placed in sentences without unusual or strained constructions. The particular sequence of ten words "attack on an English writer that the character of this" is not at all unreasonable. It appears then that a sufficiently complex stochastic process will give a satisfactory representation of a discrete source.

Language models

Definition

A model that assigns a probability distribution $P(T)$ to a "text" T, ie, a sequence of words
$T = W_1 W_2 W_3 \ldots W_L$

Language models

Definition

A model that assigns a probability distribution $P(T)$ to a "text" T, ie, a sequence of words
$T = W_1 W_2 W_3 \ldots W_L$

Examples

- Simplest: $P(T) = \prod_i P(W_i)$
- N-gram model (Markov model):
  $P(T) = \prod_i P(W_i | W_{i-n+1}, W_{i-n+2}, \ldots, W_{i-1})$

N-gram methods are widely used but overfitting is a problem. Sophisticated "smoothing" methods have been developed, eg Good-Turing, Witten-Bell, Kneser-Ney, etc.

Syntax versus semantics

Various words in a sentence each belong to a different "part of speech": noun, verb, adjective, etc.
Syntax relates to how these words can occur in relation to one another, based on their part of speech.
Semantics relates to the meaning of the words.
Computational linguistics often ignores semantics entirely!

Syntax versus semantics

Various words in a sentence each belong to a different "part of speech": noun, verb, adjective, etc.
Syntax relates to how these words can occur in relation to one another, based on their part of speech.
Semantics relates to the meaning of the words.
Computational linguistics often ignores semantics entirely!

Example

Structure of sentence:
Adjective adjective noun verb adverb

Syntactically correct, semantically meaningful sentence:
Large muscular cheetahs run fast

Syntax versus semantics

Various words in a sentence each belong to a different "part of speech": noun, verb, adjective, etc.
Syntax relates to how these words can occur in relation to one another, based on their part of speech.
Semantics relates to the meaning of the words.
Computational linguistics often ignores semantics entirely!

Example

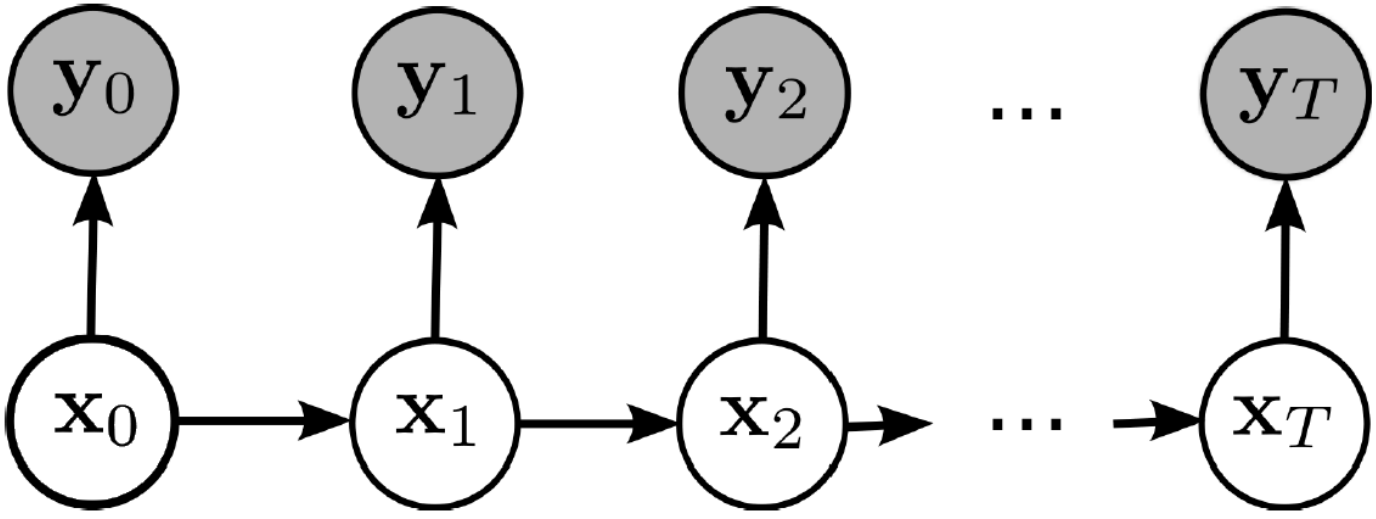
Structure of sentence:
Adjective adjective noun verb adverb

Syntactically correct, semantically meaningful sentence:
Large muscular cheetahs run fast

Syntactically correct, semantically meaningless sentence (Chomsky):
Colourless green ideas sleep furiously

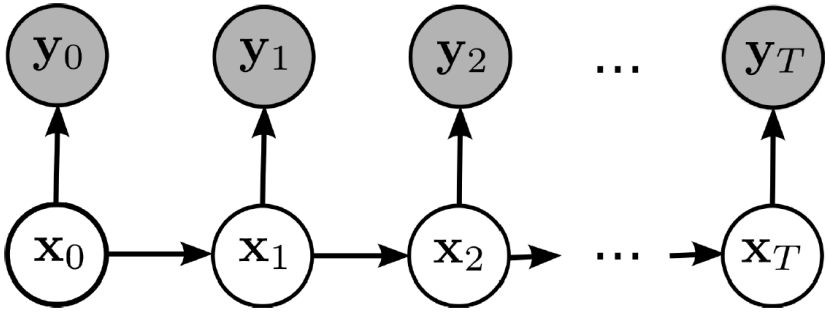
Hidden Markov models

Semantics can be important in, eg, translation:
"Cheetahs run fast"
"Gandhi sat on a fast"
How does the computer translate "fast" correctly? One way is if it "knows" whether "fast" is a noun or a verb.
In a "hidden Markov model", the Markov sequence is a sequence of "hidden states" x_i (eg, noun, verb, adjective, etc) but each hidden state "emits" a visible output y_i ("Cheetahs", "run", etc).



P. Protopapas, http://iacs-courses.seas.harvard.edu/courses/am207/blog/lecture-18.html

Hidden markov models



Widely used for modelling text, speech, music, biological sequences...
Typical use:

- The model consists of K possible "hidden states" and N possible "output states"
- There are "transition probabilities" a_{kl} governing the passage from hidden state k to hidden state l, and "emission probabilities" $e_k(a)$ for emitting symbol a while in hidden state k
- You are given a corpus of "training data" (outputs) consisting of emitted symbols, which may or may not be annotated with hidden states
- You are given new data consisting of only emitted states, and have to infer hidden states, or the probability

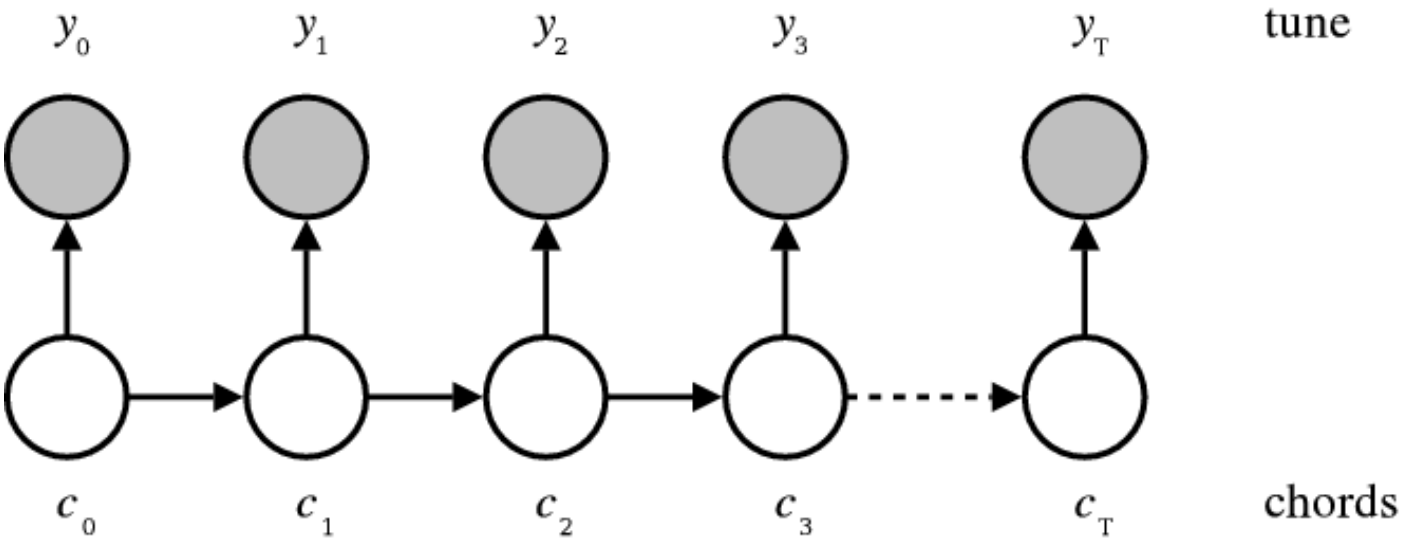
Efficient algorithms exist for all these tasks.

HMM algorithms

- **When transition/emission probabilities are known**: the Viterbi algorithm infers the most probable hidden states in linear time. The forward/backward algorithm calculates the probability of the sequence, summed over all possible hidden state paths, in linear time.
- **When transition/emission probabilities are *not* known**: the Baum-Welch (EM) algorithm infers these from training data.

Music example: harmonization

http://www.anc.inf.ed.ac.uk/demos/hmmbach/theory.html
Given a melody, infer the chords, after training



Folk music classification

Chai, Vercoe, Proc. of International Conference on Artificial Intelligence, 2001

Folk Music Classification Using Hidden Markov Models

Wei Chai
Media Laboratory
Massachusetts Institute of Technology
Cambridge, MA, U.S.A.

Barry Vercoe
Media Laboratory
Massachusetts Institute of Technology
Cambridge, MA, U.S.A.

Abstract

Automatic music classification is essential for implementing efficient music information retrieval systems; meanwhile, it may shed light on the process of human's music perception. This paper describes our work on the classification of folk music from different countries based on their monophonic melodies using hidden Markov models. Music corpora of Irish, German and Austrian folk music in various symbolic formats were used as the data set. Different representations and HMM structures were tested and compared. The classification performances achieved 75%, 77% and 66% for 2-way classifications and 63% for 3-way classification using 6-state left-right HMM with the interval representation in the experiment. This shows that the melodies of folk music do carry some statistical features to distinguish them. We expect that the result will improve if we use a more discriminable data set and the approach should be applicable to other music classification tasks and acoustic musical signals. Furthermore, the results suggest to us a new way to think about musical style similarity.

Biological sequence analysis

HMMER

DOWNLOAD DOCUMENTATION SEARCH PUBLICATIONS BLOG

HMMER: biosequence analysis using profile hidden Markov models

Get the latest version

v3.1b2

Download (Linux / Intel x86_64)

Alternative Download Options

HMMER is used for searching sequence databases for sequence homologs, and for making sequence alignments. It implements methods using probabilistic models called profile hidden Markov models (profile HMMs).

HMMER is often used together with a profile database, such as Pfam or many of the databases that participate in Interpro. But HMMER can also work with query *sequences*, not just profiles, just like BLAST. For example, you can search a protein query sequence against a database with **phmmer**, or do an iterative search with **jackhmmer**.

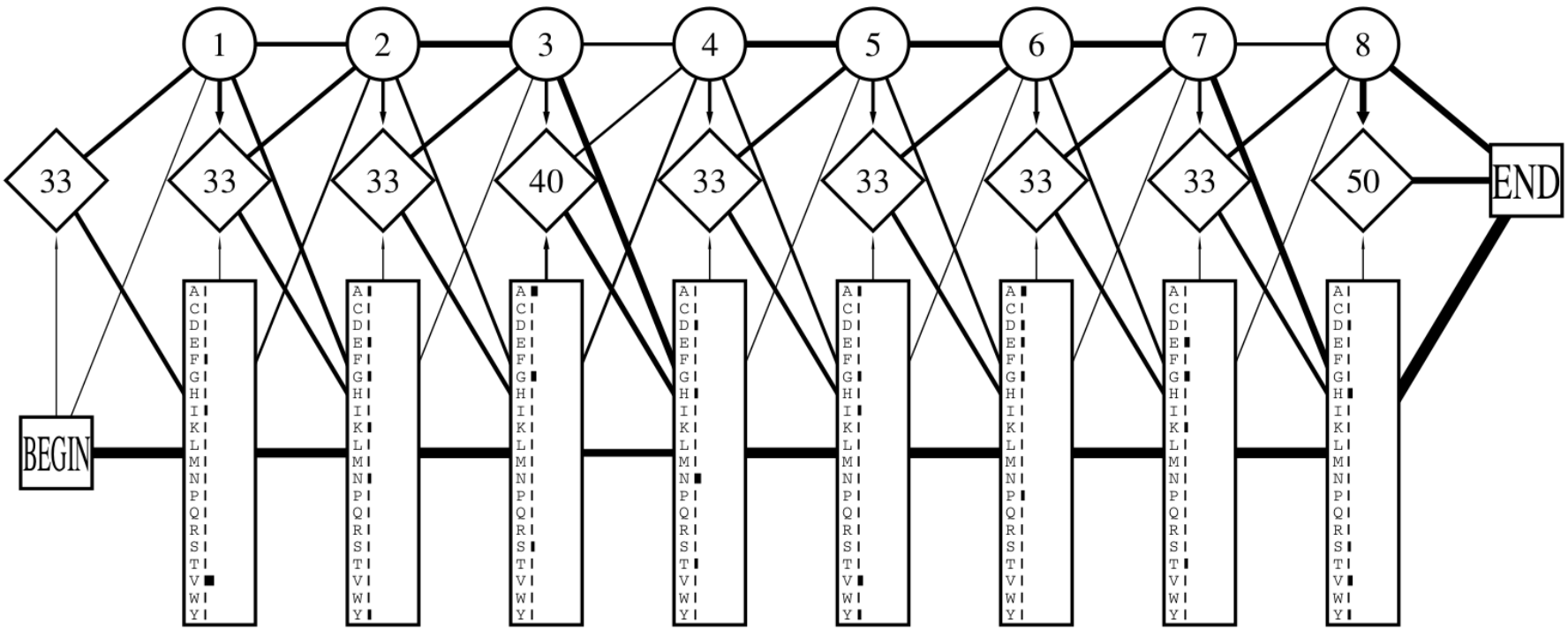
HMMER is designed to detect remote homologs as sensitively as possible, relying on the strength of its underlying probability models. In the past, this strength came at significant computational expense, but as of the new HMMER3 project, HMMER is now essentially as fast as BLAST.

HMMER can be downloaded and installed as a command line tool on your own hardware, and now it is also more wide accessible to the scientific community via new search servers at the European Bioinformatics Institute.

Biological sequence analysis

Example: HMM of a globin protein alignment (from Durbin et al, "Biological sequence analysis")

```
HBA_HUMAN    ...VGA--HAGEY...
HBB_HUMAN    ...V----NVDEV...
MYG_PHYCA    ...VEA--DVAGH...
GLB3_CHITP   ...VKG------D...
GLB5_PETMA   ...VYS--TYETS...
LGB2_LUPLU   ...FNA--NIPKH...
GLB1_GLYDI   ...IAGADNGAGV...
                ***   *****
```



Message: “Sequence analysis” has some universal features that apply to a wide variety of different types of sequences.

Thank you