

A geometric point of view on biological sequences

Enrico Rogora.

SALT LAKE CITY , April 29th 2009

We shall discuss some recent applications of mathematics, in particular algebraic geometry, to **GENETICS**

We shall discuss some recent applications of mathematics, in particular algebraic geometry, to **GENETICS**

Human genome is made up with **desossiribonucleic acid** organized in a **double elic structure** made of about 3 billions of pairs of complementary bases:

(**A**denine, **T**hymine), (**C**ytosine, **G**uanine).

We shall discuss some recent applications of mathematics, in particular algebraic geometry, to **GENETICS**

Human genome is made up with **desossiribonucleic acid** organized in a **double elic structure** made of about 3 billions of pairs of complementary bases:

(**A**denine, **T**hymine), (**C**ytosine, **G**uanine).

Genome **primary structure** can be modeled as a **sequence of letters** taken from the alphabet $\Omega = \{A, C, G, T\}$.

We shall discuss some recent applications of mathematics, in particular algebraic geometry, to **GENETICS**

Human genome is made up with **desossiribonucleic acid** organized in a **double elic structure** made of about 3 billions of pairs of complementary bases:

(**A**denine, **T**hymine), (**C**ytosine, **G**uanine).

Genome **primary structure** can be modeled as a **sequence of letters** taken from the alphabet $\Omega = \{A, C, G, T\}$.

The genome of two individuals in the same **species** is not identical: for human beings, the differences are of about **one bases over a million**, enough to explain the variability between individuals.

Biological sequences

The sequence of bases in a DNA segment, the sequence of amino acids in a protein, etc. are examples of **biological sequences**.

Biological sequences

The sequence of bases in a DNA segment, the sequence of amino acids in a protein, etc. are examples of **biological sequences**.

Mathematics has been applied to solve problems of **recognition** and **information retrieval** in biological sequence analysis.

Biological sequences

The sequence of bases in a DNA segment, the sequence of amino acids in a protein, etc. are examples of **biological sequences**.

Mathematics has been applied to solve problems of **recognition** and **information retrieval** in biological sequence analysis.

Examples

- 1 Distinguishing the protein coding segments in a gene.
- 2 Subdividing a DNA segment in different functional parts.
- 3 Identifying a protein
- 4 Sequence alignments
- 5 Building phylogenetic trees
- 6 Recognizing **ultraconserved sequences**

Probabilistic models vs deterministic ones.

During XX century **probabilistic models** have been developed as an alternative to **deterministic models** in the description of physical phenomena like (**quantum mechanics** and **statistical mechanics**).

Probabilistic models vs deterministic ones.

During XX century **probabilistic models** have been developed as an alternative to **deterministic models** in the description of physical phenomena like (**quantum mechanics** and **statistical mechanics**).

Probabilistic models are very useful also for biology, especially in the comprehension of the evolution of biological sequences.

Probabilistic models vs deterministic ones.

During XX century **probabilistic models** have been developed as an alternative to **deterministic models** in the description of physical phenomena like (**quantum mechanics** and **statistical mechanics**).

Probabilistic models are very useful also for biology, especially in the comprehension of the evolution of biological sequences.

We need to recall some some basic concepts of **probability theory**.

Coin flipping

Let us begin with a quick review of the everyone's first examples of probability theory: **coin flipping**.

Coin flipping

Let us begin with a quick review of the everyone's first examples of probability theory: **coin flipping**.

Let p the probability **to get head** ($0 \leq p \leq 1$). The probability **to get tail** is $1 - p$. If $p \neq \frac{1}{2}$ the coin is **loaded**.

Coin flipping

Let us begin with a quick review of the everyone's first examples of probability theory: **coin flipping**.

Let p the probability **to get head** ($0 \leq p \leq 1$). The probability **to get tail** is $1 - p$. If $p \neq \frac{1}{2}$ the coin is **loaded**.

How is possible to **estimate** p , hence to assess if the coin is loaded? A popular approach is through the **maximum likelihood principle**.

Maximum likelihood estimates

Flip a coin repeatedly and record the outcomes. For example

TCCTCTTCTCCTTT

Maximum likelihood estimates

Flip a coin repeatedly and record the outcomes. For example

TCCTCTTCTCCTTT

Basic assumption

The outcomes are **independent events**, i.e. the outcome of each flipping does not depend on the outcome of the others.

Maximum likelihood estimates

Flip a coin repeatedly and record the outcomes. For example

TCCTCTTCTCCTTT

Basic assumption

The outcomes are **independent events**, i.e. the outcome of each flipping does not depend on the outcome of the others.

The probability for the above sequence is

$$L(p) = p^8(1 - p)^6.$$

Maximum likelihood estimates

Flip a coin repeatedly and record the outcomes. For example

TCCTCTTCTCCTTT

Basic assumption

The outcomes are **independent events**, i.e. the outcome of each flipping does not depend on the outcome of the others.

The probability for the above sequence is

$$L(p) = p^8(1 - p)^6.$$

Maximum likelihood estimates

The **maximum likelihood estimate** for p is the value which maximizes $L(p)$, in $[0, 1]$. In the example is $\frac{4}{7}$.

Probabilistic models for DNA sequences generation

The outcomes of a sequence generation process is any **sequence** from the alphabet $\Omega = \{A, C, G, T\}$.

Probabilistic models for DNA sequences generation

The outcomes of a sequence generation process is any **sequence** from the alphabet $\Omega = \{A, C, G, T\}$.

The simplest model of sequence generation is the **model of the urn**.

Probabilistic models for DNA sequences generation

The outcomes of a sequence generation process is any **sequence** from the alphabet $\Omega = \{A, C, G, T\}$.

The simplest model of sequence generation is the **model of the urn**.

One considers an urn with n_A balls marked A , and analogously for n_C , n_G e n_T .

Probabilistic models for DNA sequences generation

The outcomes of a sequence generation process is any **sequence** from the alphabet $\Omega = \{A, C, G, T\}$.

The simplest model of sequence generation is the **model of the urn**.

One considers an urn with n_A balls marked A , and analogously for n_C , n_G e n_T . To generate a sequence

Probabilistic models for DNA sequences generation

The outcomes of a sequence generation process is any **sequence** from the alphabet $\Omega = \{A, C, G, T\}$.

The simplest model of sequence generation is the **model of the urn**.

One considers an urn with n_A balls marked A , and analogously for n_C , n_G e n_T . To generate a sequence **draw a ball**,

Probabilistic models for DNA sequences generation

The outcomes of a sequence generation process is any **sequence** from the alphabet $\Omega = \{A, C, G, T\}$.

The simplest model of sequence generation is the **model of the urn**.

One considers an urn with n_A balls marked A , and analogously for n_C , n_G e n_T . To generate a sequence **draw a ball**, **copy the marking**,

Probabilistic models for DNA sequences generation

The outcomes of a sequence generation process is any **sequence** from the alphabet $\Omega = \{A, C, G, T\}$.

The simplest model of sequence generation is the **model of the urn**.

One considers an urn with n_A balls marked A , and analogously for n_C , n_G e n_T . To generate a sequence **draw a ball**, **copy the marking**, **put the ball back in the urn**,

Probabilistic models for DNA sequences generation

The outcomes of a sequence generation process is any **sequence** from the alphabet $\Omega = \{A, C, G, T\}$.

The simplest model of sequence generation is the **model of the urn**.

One considers an urn with n_A balls marked A , and analogously for n_C , n_G e n_T . To generate a sequence **draw a ball**, **copy the marking**, **put the ball back in the urn**, **shake well and repeat**.

Probabilistic models for DNA sequences generation

The outcomes of a sequence generation process is any **sequence** from the alphabet $\Omega = \{A, C, G, T\}$.

The simplest model of sequence generation is the **model of the urn**.

One considers an urn with n_A balls marked A , and analogously for n_C , n_G e n_T . To generate a sequence **draw a ball**, **copy the marking**, **put the ball back in the urn**, **shake well and repeat**.

In this model

$$p_A = \frac{n_A}{n_A + n_C + n_G + n_T}$$

is the probability to draw a ball marked A , and the same for the others.

Probabilistic models for DNA sequences generation

The outcomes of a sequence generation process is any **sequence** from the alphabet $\Omega = \{A, C, G, T\}$.

The simplest model of sequence generation is the **model of the urn**.

One considers an urn with n_A balls marked A , and analogously for n_C , n_G e n_T . To generate a sequence **draw a ball**, **copy the marking**, **put the ball back in the urn**, **shake well and repeat**.

In this model

$$p_A = \frac{n_A}{n_A + n_C + n_G + n_T}$$

is the probability to draw a ball marked A , and the same for the others. The probabilities of all sequences can be immediately computed, since the model assume implies independence of different extractions.

Probabilistic models for DNA sequences generation

The outcomes of a sequence generation process is any **sequence** from the alphabet $\Omega = \{A, C, G, T\}$.

The simplest model of sequence generation is the **model of the urn**.

One considers an urn with n_A balls marked A , and analogously for n_C , n_G e n_T . To generate a sequence **draw a ball**, **copy the marking**, **put the ball back in the urn**, **shake well and repeat**.

In this model

$$p_A = \frac{n_A}{n_A + n_C + n_G + n_T}$$

is the probability to draw a ball marked A , and the same for the others. The probabilities of all sequences can be immediately computed, since the model assume implies independence of different extractions. Note that $p_A + p_C + p_G + p_T = 1$, hence the urn model depends on **three essential parametrs**.

Parameters estimate and model adequacy

Given a biological sequence and a probabilistic model of generation (like the urn model) one consider two problems

Parameters estimate and model adequacy

Given a biological sequence and a probabilistic model of generation (like the urn model) one consider two problems

- **Parameters estimate** for example p_A , p_C , etc.

Parameters estimate and model adequacy

Given a biological sequence and a probabilistic model of generation (like the urn model) one consider two problems

- **Parameters estimate** for example p_A , p_C , etc.
- **Model adequacy**

Parameters estimate and model adequacy

Given a biological sequence and a probabilistic model of generation (like the urn model) one consider two problems

- **Parameters estimate** for example p_A , p_C , etc.
- **Model adequacy**

For parameter estimates one usually employs the **principle of maximum likelihood**.

Parameters estimate and model adequacy

Given a biological sequence and a probabilistic model of generation (like the urn model) one consider two problems

- **Parameters estimate** for example p_A , p_C , etc.
- **Model adequacy**

For parameter estimates one usually employs the **principle of maximum likelihood**.

For model adequacy there exists some statistical models: e.g. Akaike information criterion, etc.

Parameters estimate and model adequacy

Given a biological sequence and a probabilistic model of generation (like the urn model) one consider two problems

- **Parameters estimate** for example p_A , p_C , etc.
- **Model adequacy**

For parameter estimates one usually employs the **principle of maximum likelihood**.

For model adequacy there exists some statistical models: e.g. Akaike information criterion, etc.

The urn model **is not useful** to describe the statistical properties of DNA sequences. We need to allow different urns and to model a **selection mechanism**.

Markov Model

Let us consider **two plates** marked \mathcal{T} and \mathcal{C} .

Markov Model

Let us consider **two plates** marked \mathcal{T} and \mathcal{C} . **On every plate there is an urn** containing balls marked A, C, G, T , **and a coin** with symbols \mathcal{T} and \mathcal{C} .

Markov Model

Let us consider **two plates** marked \mathcal{T} and \mathcal{C} . **On every plate there is an urn** containing balls marked A, C, G, T , **and a coin** with symbols \mathcal{T} and \mathcal{C} . The two coins are **loaded differently**.

Markov Model

Let us consider **two plates** marked \mathcal{T} and \mathcal{C} . **On every plate there is an urn** containing balls marked A, C, G, T , **and a coin** with symbols \mathcal{T} and \mathcal{C} . The two coins are **loaded differently**. The probability $p(X, \mathcal{Y})$ to draw X from the urn \mathcal{Y} is a function of the numbers $n_{X, \mathcal{Y}}$ of balls marked X in the urn \mathcal{Y} .

Markov Model

Let us consider **two plates** marked \mathcal{T} and \mathcal{C} . **On every plate there is an urn** containing balls marked A, C, G, T , **and a coin** with symbols \mathcal{T} and \mathcal{C} . The two coins are **loaded differently**. The probability $p(X, \mathcal{Y})$ to draw X from the urn \mathcal{Y} is a function of the numbers $n_{X, \mathcal{Y}}$ of balls marked X in the urn \mathcal{Y} .

- **Inizialization**: one flips **a third coin** to chose the plate where to begin.

Markov Model

Let us consider **two plates** marked \mathcal{T} and \mathcal{C} . **On every plate there is an urn** containing balls marked A, C, G, T , **and a coin** with symbols \mathcal{T} and \mathcal{C} . The two coins are **loaded differently**. The probability $p(X, \mathcal{Y})$ to draw X from the urn \mathcal{Y} is a function of the numbers $n_{X, \mathcal{Y}}$ of balls marked X in the urn \mathcal{Y} .

- **Inizialization**: one flips **a third coin** to chose the plate where to begin.
- **Iteration**: One draws a ball from the urn on the **active** plate,

Markov Model

Let us consider **two plates** marked \mathcal{T} and \mathcal{C} . **On every plate there is an urn** containing balls marked A, C, G, T , **and a coin** with symbols \mathcal{T} and \mathcal{C} . The two coins are **loaded differently**. The probability $p(X, \mathcal{Y})$ to draw X from the urn \mathcal{Y} is a function of the numbers $n_{X, \mathcal{Y}}$ of balls marked X in the urn \mathcal{Y} .

- **Initialization:** one flips **a third coin** to chose the plate where to begin.
- **Iteration:** One draws a ball from the urn on the **active** plate, **copies the mark**,

Markov Model

Let us consider **two plates** marked \mathcal{T} and \mathcal{C} . **On every plate there is an urn** containing balls marked A, C, G, T , **and a coin** with symbols \mathcal{T} and \mathcal{C} . The two coins are **loaded differently**. The probability $p(X, \mathcal{Y})$ to draw X from the urn \mathcal{Y} is a function of the numbers $n_{X, \mathcal{Y}}$ of balls marked X in the urn \mathcal{Y} .

- **Initialization:** one flips **a third coin** to chose the plate where to begin.
- **Iteration:** One draws a ball from the urn on the **active** plate, **copies the mark**, **flip the coin on the current plate to chose the new plate**,

Markov Model

Let us consider **two plates** marked \mathcal{T} and \mathcal{C} . **On every plate there is an urn** containing balls marked A, C, G, T , **and a coin** with symbols \mathcal{T} and \mathcal{C} . The two coins are **loaded differently**. The probability $p(X, \mathcal{Y})$ to draw X from the urn \mathcal{Y} is a function of the numbers $n_{X, \mathcal{Y}}$ of balls marked X in the urn \mathcal{Y} .

- **Initialization:** one flips **a third coin** to chose the plate where to begin.
- **Iteration:** One draws a ball from the urn on the **active** plate, **copies the mark**, **flip the coin on the current plate to chose the new plate**, **repeats**.

Markov Model

Let us consider **two plates** marked \mathcal{T} and \mathcal{C} . **On every plate there is an urn** containing balls marked A, C, G, T , **and a coin** with symbols \mathcal{T} and \mathcal{C} . The two coins are **loaded differently**. The probability $p(X, \mathcal{Y})$ to draw X from the urn \mathcal{Y} is a function of the numbers $n_{X, \mathcal{Y}}$ of balls marked X in the urn \mathcal{Y} .

- **Initialization**: one flips **a third coin** to chose the plate where to begin.
- **Iteration**: One draws a ball from the urn on the **active** plate, **copies the mark**, **flip the coin on the current plate to chose the new plate**, **repeats**.

This process depends on **9** independent parameters, and is an example of **Markov model**.

Markov Model

Let us consider **two plates** marked \mathcal{T} and \mathcal{C} . **On every plate there is an urn** containing balls marked A, C, G, T , **and a coin** with symbols \mathcal{T} and \mathcal{C} . The two coins are **loaded differently**. The probability $p(X, \mathcal{Y})$ to draw X from the urn \mathcal{Y} is a function of the numbers $n_{X, \mathcal{Y}}$ of balls marked X in the urn \mathcal{Y} .

- **Initialization**: one flips **a third coin** to chose the plate where to begin.
- **Iteration**: One draws a ball from the urn on the **active** plate, **copies the mark**, **flip the coin on the current plate to chose the new plate**, **repeats**.

This process depends on **9** independent parameters, and is an example of **Markov model**. It can be usefully employed to describe **some aspects** of DNA sequences.

Markov Model

Let us consider **two plates** marked \mathcal{T} and \mathcal{C} . **On every plate there is an urn** containing balls marked A, C, G, T , **and a coin** with symbols \mathcal{T} and \mathcal{C} . The two coins are **loaded differently**. The probability $p(X, \mathcal{Y})$ to draw X from the urn \mathcal{Y} is a function of the numbers $n_{X, \mathcal{Y}}$ of balls marked X in the urn \mathcal{Y} .

- **Initialization:** one flips **a third coin** to chose the plate where to begin.
- **Iteration:** One draws a ball from the urn on the **active** plate, **copies the mark**, **flip the coin on the current plate to chose the new plate**, **repeats**.

This process depends on **9** independent parameters, and is an example of **Markov model**. It can be usefully employed to describe **some aspects** of DNA sequences. In biological applications one usually **does not observe directly the mechanism of urns selection**.

Hidden Markov models

In a probabilistic model, one needs often to consider **hidden states**, i.e. **non observable states**, on which observations depend.

Hidden Markov models

In a probabilistic model, one needs often to consider **hidden states**, i.e. **non observable states**, on which observations depend. These models are introduced in order to **estimate the hidden states given some observations**. Usually many different hidden states may produce the same observations and one choses the **most probable**.

Hidden Markov models

In a probabilistic model, one needs often to consider **hidden states**, i.e. **non observable states**, on which observations depend.

These models are introduced in order to **estimate the hidden states given some observations**. Usually many different hidden states may produce the same observations and one chooses the **most probable**.

In the preceding example we have an hidden state model when one can only observe the marks on the balls which are drawn **without knowing** from which urn they are drawn, i.e. without knowing the outcome of the coins flipping which determines the urns from which we draw the balls.

Hidden Markov chains

An hidden Markov chain is described by the following data:

Hidden Markov chains

An hidden Markov chain is described by the following data:

- 1 An alphabet $N = \{n_1, \dots, n_h\}$ of **hidden states**.

Hidden Markov chains

An hidden Markov chain is described by the following data:

- 1 An alphabet $N = \{n_1, \dots, n_h\}$ of **hidden states**.
- 2 An alphabet $V = \{v_1, \dots, v_k\}$ of **visible symbols**.

Hidden Markov chains

An hidden Markov chain is described by the following data:

- 1 An alphabet $N = \{n_1, \dots, n_h\}$ of **hidden states**.
- 2 An alphabet $V = \{v_1, \dots, v_k\}$ of **visible symbols**.
- 3 The vector $p = (p_1, \dots, p_h)$ of **initial probability**: p_i is the probability that the initial state is n_i .

Hidden Markov chains

An hidden Markov chain is described by the following data:

- 1 An alphabet $N = \{n_1, \dots, n_h\}$ of **hidden states**.
- 2 An alphabet $V = \{v_1, \dots, v_k\}$ of **visible symbols**.
- 3 The vector $p = (p_1, \dots, p_h)$ of **initial probability**: p_i is the probability that the initial state is n_i .
- 4 The **transition matrix** $T = (t_{ij})$ between hidden states: t_{ij} is the probability to make a transition from state n_i to state n_j .

Hidden Markov chains

An hidden Markov chain is described by the following data:

- 1 An alphabet $N = \{n_1, \dots, n_h\}$ of **hidden states**.
- 2 An alphabet $V = \{v_1, \dots, v_k\}$ of **visible symbols**.
- 3 The vector $p = (p_1, \dots, p_h)$ of **initial probability**: p_i is the probability that the initial state is n_i .
- 4 The **transition matrix** $T = (t_{ij})$ between hidden states: t_{ij} is the probability to make a transition from state n_i to state n_j .
- 5 The **emission matrix** $E = (e_{is})$: e_{is} is the probability that state n_i emits symbol v_s .

Hidden Markov chain (II)

The mechanism which generates a sequence of visible symbols is the following:

Hidden Markov chain (II)

The mechanism which generates a sequence of visible symbols is the following:

- 1 A **initial hidden state** x_1 is chosen by flipping a coin with h faces, whose probabilities are described by the vector p .

Hidden Markov chain (II)

The mechanism which generates a sequence of visible symbols is the following:

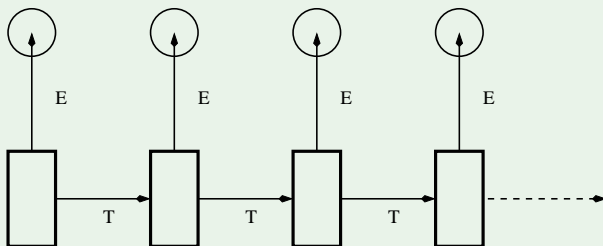
- 1 A **initial hidden state** x_1 is chosen by flipping a coin with h faces, whose probabilities are described by the vector p .
- 2 The first **visible symbol** y_1 is produced from x_1 by drawing from a suitable urn with drawing probabilities described by the rows of E corresponding to x_1 .

Hidden Markov chain (II)

The mechanism which generates a sequence of visible symbols is the following:

- 1 A **initial hidden state** x_1 is chosen by flipping a coin with h faces, whose probabilities are described by the vector p .
- 2 The first **visible symbol** y_1 is produced from x_1 by drawing from a suitable urn with drawing probabilities described by the rows of E corresponding to x_1 .
- 3 The **new hidden state** x_2 is produced from x_1 by flipping a coin with h faces whose probabilities are described by the row of T corresponding to x_1 , and so on.

Graph associated to a Hidden Markov Chain



Probabilities of sequences produced by an Hidden Markov chain

Sequence of hidden states

$$\sigma = (\sigma_1, \dots, \sigma_n) \quad \sigma_i \in N$$

Probabilities of sequences produced by an Hidden Markov chain

Sequence of **hidden states**

$$\sigma = (\sigma_1, \dots, \sigma_n) \quad \sigma_i \in N$$

Sequence of **visible states**

$$\tau = (\tau_1, \dots, \tau_n) \quad \tau_j \in V$$

Probabilities of sequences produced by an Hidden Markov chain

Sequence of **hidden states**

$$\sigma = (\sigma_1, \dots, \sigma_n) \quad \sigma_i \in N$$

Sequence of **visible states**

$$\tau = (\tau_1, \dots, \tau_n) \quad \tau_j \in V$$

The **probability** to see the outcome τ given the hidden states σ is given by the **monomial**

$$p_{\sigma\tau} = p_{\sigma_1} e_{\sigma_1\tau_1} t_{\sigma_1\sigma_2} e_{\sigma_2\tau_2} t_{\sigma_2\sigma_3} e_{\sigma_3\tau_3} \dots t_{\sigma_{n-1}\sigma_n} e_{\sigma_n\tau_n}$$

Probabilities of sequences produced by an Hidden Markov chain

Sequence of **hidden states**

$$\sigma = (\sigma_1, \dots, \sigma_n) \quad \sigma_i \in N$$

Sequence of **visible states**

$$\tau = (\tau_1, \dots, \tau_n) \quad \tau_j \in V$$

The **probability** to see the outcome τ given the hidden states σ is given by the **monomial**

$$p_{\sigma\tau} = p_{\sigma_1} e_{\sigma_1\tau_1} t_{\sigma_1\sigma_2} e_{\sigma_2\tau_2} t_{\sigma_2\sigma_3} e_{\sigma_3\tau_3} \dots t_{\sigma_{n-1}\sigma_n} e_{\sigma_n\tau_n}$$

The **probability** to see τ is given by the **polynomial**

$$p_{\tau} = \sum_{\sigma \in N^n} p_{\sigma\tau}$$

Problems and algorithms

Hidden Markov models for the generation of symbols from an alphabet allow us to deal with the following problems, for which efficient algorithms of **Dynamic Programming** exist

Problems and algorithms

Hidden Markov models for the generation of symbols from an alphabet allow us to deal with the following problems, for which efficient algorithms of **Dynamic Programming** exist

- 1 Determination of a sequence of hidden states which produced a given outcome (eg determination of CpG islands and gene boundaries) **Viterbi algorithm**

Problems and algorithms

Hidden Markov models for the generation of symbols from an alphabet allow us to deal with the following problems, for which efficient algorithms of **Dynamic Programming** exist

- 1 Determination of a sequence of hidden states which produced a given outcome (eg determination of CpG islands and gene boundaries) **Viterbi algorithm**
- 2 Computation of the probability of a given outcome. **Forward algorithm**

Problems and algorithms

Hidden Markov models for the generation of symbols from an alphabet allow us to deal with the following problems, for which efficient algorithms of **Dynamic Programming** exist

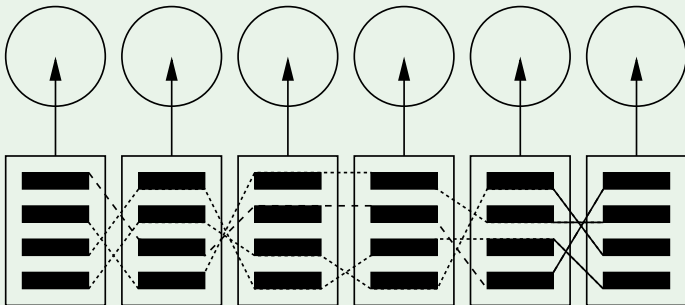
- 1 Determination of a sequence of hidden states which produced a given outcome (eg determination of CpG islands and gene boundaries) **Viterbi algorithm**
- 2 Computation of the probability of a given outcome. **Forward algorithm**
- 3 Computation of conditional probabilities $p(x_i = k|y)$, i.e. posterior probability of state k at time i , given the outcome, **backward algorithm**

Problems and algorithms

Hidden Markov models for the generation of symbols from an alphabet allow us to deal with the following problems, for which efficient algorithms of **Dynamic Programming** exist

- 1 Determination of a sequence of hidden states which produced a given outcome (eg determination of CpG islands and gene boundaries) **Viterbi algorithm**
- 2 Computation of the probability of a given outcome. **Forward algorithm**
- 3 Computation of conditional probabilities $p(x_i = k|y)$, i.e. posterior probability of state k at time i , given the outcome, **backward algorithm**
- 4 Estimation of the parameters of a hidden Markov Model which better describes the data **Baum Welch algorithm**

Viterbi algorithm



The formula

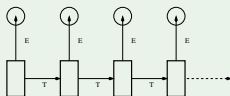
$$p_{\tau} = \sum_{\sigma \in N^n} p_{\sigma\tau}$$

Graphic models

The formula

$$p_{\tau} = \sum_{\sigma \in N^n} p_{\sigma\tau}$$

and the graph

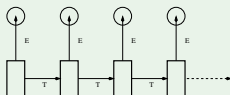


Graphic models

The formula

$$p_{\tau} = \sum_{\sigma \in N^n} p_{\sigma\tau}$$

and the graph



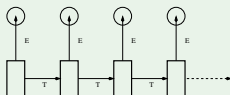
synthesize in equivalent fashion the structure of Hidden Markov models.

Graphic models

The formula

$$p_{\tau} = \sum_{\sigma \in N^n} p_{\sigma\tau}$$

and the graph



synthesize in equivalent fashion the structure of Hidden Markov models.

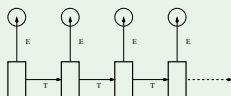
More generally, there exists a large class of probabilistic models, the so called **graphical models**, where **algebraic** and **combinatoric** aspects interconnects in an analogous way and for which useful **geometric interpretations** exist.

Graphic models

The formula

$$p_{\tau} = \sum_{\sigma \in N^n} p_{\sigma\tau}$$

and the graph



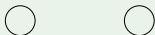
synthesize in equivalent fashion the structure of Hidden Markov models.

More generally, there exists a large class of probabilistic models, the so called **graphical models**, where **algebraic** and **combinatoric** aspects interconnects in an analogous way and for which useful **geometric interpretations** exist.

We discuss briefly some of these models.

The model of independence

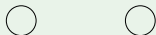
Let us consider the graph



To each of its vertexes is associated an alphabet consisting of two symbols $\{E, I\}$ and the probabilities $p_E^{(i)}, p_I^{(i)}$ of the output E or I at the vertex i .

The model of independence

Let us consider the graph



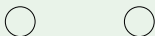
To each of its vertexes is associated an alphabet consisting of two symbols $\{E, I\}$ and the probabilities $p_E^{(i)}, p_I^{(i)}$ of the output E or I at the vertex i .

The **model of independence** assigns to the four possible outcomes the probabilities, given by **monomials**, displayed in the following table

EE	EI	IE	II
$p_E^{(1)} p_E^{(2)}$	$p_E^{(1)} p_I^{(2)}$	$p_I^{(1)} p_E^{(2)}$	$p_I^{(1)} p_I^{(2)}$

The model of independence

Let us consider the graph



To each of its vertexes is associated an alphabet consisting of two symbols $\{E, I\}$ and the probabilities $p_E^{(i)}, p_I^{(i)}$ of the output E or I at the vertex i .

The **model of independence** assigns to the four possible outcomes the probabilities, given by **monomials**, displayed in the following table

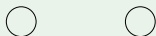
EE	EI	IE	II
$p_E^{(1)} p_E^{(2)}$	$p_E^{(1)} p_I^{(2)}$	$p_I^{(1)} p_E^{(2)}$	$p_I^{(1)} p_I^{(2)}$

The space of probability distributions over the possible outcomes $\{EE, EI, IE, II\}$ is the set Δ of the possible quadruple (x_0, x_1, x_2, x_3) of real numbers such that

$$0 \leq x_i \leq 1 \quad i = 0, \dots, 3; \quad x_0 + x_1 = x_2 + x_3 = 1.$$

The model of independence

Let us consider the graph



To each of its vertexes is associated an alphabet consisting of two symbols $\{E, I\}$ and the probabilities $p_E^{(i)}, p_I^{(i)}$ of the output E or I at the vertex i .

The **model of independence** assigns to the four possible outcomes the probabilities, given by **monomials**, displayed in the following table

EE	EI	IE	II
$p_E^{(1)} p_E^{(2)}$	$p_E^{(1)} p_I^{(2)}$	$p_I^{(1)} p_E^{(2)}$	$p_I^{(1)} p_I^{(2)}$

The space of probability distributions over the possible outcomes $\{EE, EI, IE, II\}$ is the set Δ of the possible quadruple (x_0, x_1, x_2, x_3) of real numbers such that

$$0 \leq x_i \leq 1 \quad i = 0, \dots, 3; \quad x_0 + x_1 = x_2 + x_3 = 1.$$

The model of independence selects in Δ the subset given by the **algebraic equation**

$$x_0 x_3 - x_1 x_2 = 0.$$

The model of independence (II)

In its more general version **the model of independence** is associated to the graph with m vertices



The model of independence (II)

In its more general version **the model of independence** is associated to the graph with m vertices



To vertex i is associated an alphabet of $n_i + 1$ symbols $a_j^{(i)}$, $j = 0, \dots, n_i$, and the probabilities $p_j^{(i)}$ to have the outcome $a_j^{(i)}$ in such vertex.

The model of independence (II)

In its more general version **the model of independence** is associated to the graph with m vertices



To vertex i is associated an alphabet of $n_i + 1$ symbols $a_j^{(i)}$, $j = 0, \dots, n_i$, and the probabilities $p_j^{(i)}$ to have the outcome $a_j^{(i)}$ in such vertex.

The **model of independence** assigns to the outcome $a_{i_1}^{(1)}, \dots, a_{i_m}^{(m)}$ the probability given by the **monomial**

$$p(i_1, \dots, i_m) = p_{i_1}^{(1)} \cdots p_{i_m}^{(m)}$$

The model of independence: algebraic - geometric aspects

The space of probability distribution over all passible outcomes is the set Δ of $(n_1 + 1) \dots (n_m + 1)$ -ples

$$(x_{i_1 \dots i_m}), \quad i_j = 0, \dots, n_j, \quad j = 1, \dots, m$$

satisfying suitable **algebraic constraints** which we will not consider now.

The model of independence: algebraic - geometric aspects

The space of probability distribution over all passible outcomes is the set Δ of $(n_1 + 1) \dots (n_m + 1)$ -ples

$$(x_{i_1 \dots i_m}), \quad i_j = 0, \dots, n_j, \quad j = 1, \dots, m$$

satisfying suitable **algebraic constraints** which we will not consider now.

The model of independence selects in Δ a subset given by a **system of homogeneous algebraic equations** of second degree.

The model of independence: algebraic - geometric aspects

The space of probability distribution over all passible outcomes is the set Δ of $(n_1 + 1) \dots (n_m + 1)$ -ples

$$(x_{i_1 \dots i_m}), \quad i_j = 0, \dots, n_j, \quad j = 1, \dots, m$$

satisfying suitable **algebraic constraints** which we will not consider now.

The model of independence selects in Δ a subset given by a **system of homogeneous algebraic equations** of second degree. For example, if $m = 2$ the set of equations become

$$\text{rk}(x_{ij}) = 1$$

The model of independence: algebraic - geometric aspects

The space of probability distribution over all passible outcomes is the set Δ of $(n_1 + 1) \dots (n_m + 1)$ -ples

$$(x_{i_1 \dots i_m}), \quad i_j = 0, \dots, n_j, \quad j = 1, \dots, m$$

satisfying suitable **algebraic constraints** which we will not consider now.

The model of independence selects in Δ a subset given by a **system of homogeneous algebraic equations** of second degree.

For example, if $m = 2$ the set of equations become

$$\text{rk}(x_{ij}) = 1$$

These systems of equations define famous **algebraic varieties**, the so called **Segre varieties**: they represents products of projective spaces.

The model of independence: algebraic - geometric aspects

The space of probability distribution over all passible outcomes is the set Δ of $(n_1 + 1) \dots (n_m + 1)$ -ples

$$(x_{i_1 \dots i_m}), \quad i_j = 0, \dots, n_j, \quad j = 1, \dots, m$$

satisfying suitable **algebraic constraints** which we will not consider now.

The model of independence selects in Δ a subset given by a **system of homogeneous algebraic equations** of second degree. For example, if $m = 2$ the set of equations become

$$\text{rk}(x_{ij}) = 1$$

These systems of equations define famous **algebraic varieties**, the so called **Segre varieties**: they represents products of projective spaces.

The case $m = 2, n_0 = n_1 = 1$, already considered, gives a **quadric** in three dimensional projective space, which is the product of two projective lines.

A simple model of dependence

Let us consider the graph



A simple model of dependence

Let us consider the graph



To any of its three vertices is associated an alphabet :

0	a_0, \dots, a_h
1	b_0, \dots, b_n
2	c_0, \dots, c_m

A simple model of dependence

Let us consider the graph



To any of its three vertices is associated an alphabet :

0	a_0, \dots, a_h
1	b_0, \dots, b_n
2	c_0, \dots, c_m

The **parameters** of the model are:

- The probability p_0, \dots, p_h to see a_0, \dots, a_h in 0.

A simple model of dependence

Let us consider the graph



To any of its three vertices is associated an alphabet :

0	a_0, \dots, a_h
1	b_0, \dots, b_n
2	c_0, \dots, c_m

The **parameters** of the model are:

- The probability p_0, \dots, p_h to see a_0, \dots, a_h in 0.
- The $n \times h$ matrix T , for which t_{ij} is the probability to see b_i in 1 if a_j was seen in 0.

A simple model of dependence

Let us consider the graph



To any of its three vertices is associated an alphabet :

0	a_0, \dots, a_h
1	b_0, \dots, b_n
2	c_0, \dots, c_m

The **parameters** of the model are:

- The probability p_0, \dots, p_h to see a_0, \dots, a_h in 0.
- The $n \times h$ matrix T , for which t_{ij} is the probability to see b_i in 1 if a_j was seen in 0.
- The $m \times h$ matrix S where, s_{ij} is the probability to see c_i in 2 se è stato osservato a_j in 0.

A simple model of dependence

Let us consider the graph



To any of its three vertices is associated an alphabet :

0	a_0, \dots, a_h
1	b_0, \dots, b_n
2	c_0, \dots, c_m

The **parameters** of the model are:

- The probability p_0, \dots, p_h to see a_0, \dots, a_h in 0.
- The $n \times h$ matrix T , for which t_{ij} is the probability to see b_i in 1 if a_j was seen in 0.
- The $m \times h$ matrix S where, s_{ij} is the probability to see c_i in 2 se è stato osservato a_j in 0.

Under this model, for 0 is a **hidden state**, the probability of (b_i, c_j) is given by **polynomial**

$$p_{ij} = \sum_{\alpha=0}^h p_{\alpha} t_{i\alpha} s_{j\alpha}.$$

A simple model of dependence (II)

The space of probability distributions for the possible outcomes under this model is the set Δ of $(n+1)(m+1)$ -ples

$$(x_{ij}), \quad i = 0, \dots, n, \quad j = 0, \dots, m.$$

A simple model of dependence (II)

The space of probability distributions for the possible outcomes under this model is the set Δ of $(n+1)(m+1)$ -ples

$$(x_{ij}), \quad i = 0, \dots, n, \quad j = 0, \dots, m.$$

This model selects in Δ a subset given by a system of algebraic equations

$$\text{rk}(x_{ij}) \leq h + 1$$

A simple model of dependence (II)

The space of probability distributions for the possible outcomes under this model is the set Δ of $(n+1)(m+1)$ -ples

$$(x_{ij}), \quad i = 0, \dots, n, \quad j = 0, \dots, m.$$

This model selects in Δ a subset given by a system of algebraic equations

$$\text{rk}(x_{ij}) \leq h + 1$$

These systems of equations define important algebraic varieties i.e. **Varieties of secant spaces to Segre varieties** $\mathbb{P}^n \times \mathbb{P}^m$.

A simple model of dependence (II)

The space of probability distributions for the possible outcomes under this model is the set Δ of $(n+1)(m+1)$ -ples

$$(x_{ij}), \quad i = 0, \dots, n, \quad j = 0, \dots, m.$$

This model selects in Δ a subset given by a system of algebraic equations

$$\text{rk}(x_{ij}) \leq h + 1$$

These systems of equations define important algebraic varieties i.e. **Varieties of secant spaces to Segre varieties** $\mathbb{P}^n \times \mathbb{P}^m$.

This model can be generalized by considering the graph



A simple model of dependence (II)

The space of probability distributions for the possible outcomes under this model is the set Δ of $(n+1)(m+1)$ -ples

$$(x_{ij}), \quad i = 0, \dots, n, \quad j = 0, \dots, m.$$

This model selects in Δ a subset given by a system of algebraic equations

$$\text{rk}(x_{ij}) \leq h + 1$$

These systems of equations define important algebraic varieties i.e. **Varieties of secant spaces to Segre varieties** $\mathbb{P}^n \times \mathbb{P}^m$.

This model can be generalized by considering the graph



From a geometro-algebraic point of view this amounts to considering **varieties of secant spaces to Segre products with several factors**.

Algebraic varieties associated to Hidden Markov Chains

In an hidden Markov Chain, the algebraic expressions for probability **parametrize** an algebraic variety.

Algebraic varieties associated to Hidden Markov Chains

In an hidden Markov Chain, the algebraic expressions for probability **parametrize** an algebraic variety.

Many of these varieties have never been studied before and suggest **interesting new problems for algebraic geometry**.

Algebraic varieties associated to Hidden Markov Chains

In an hidden Markov Chain, the algebraic expressions for probability **parametrize** an algebraic variety.

Many of these varieties have never been studied before and suggest **interesting new problems for algebraic geometry**. For example the problem to find a **system of equations for the variety**.

Algebraic varieties associated to Hidden Markov Chains

In an hidden Markov Chain, the algebraic expressions for probability **parametrize** an algebraic variety.

Many of these varieties have never been studied before and suggest **interesting new problems for algebraic geometry**. For example the problem to find a **system of equations for the variety**.

Viceversa, recent **combinatorial and computational techniques** in algebraic geometry (**Gröbner basis**, **toric geometry**, **tropical geometry**) suggest new algorithms to solve problems of **parameter estimates** and **model adequacy verification**.

Floyd-Warshall algorithm

To give an idea of the dynamic programming algorithms used in the application of the graphical models, we consider **Floyd-Warshall algorithm**

Floyd-Warshall algorithm

To give an idea of the dynamic programming algorithms used in the application of the graphical models, we consider **Floyd-Warshall algorithm**

Problem

Find the minimal path in a weighted graph.

Floyd-Warshall algorithm

To give an idea of the dynamic programming algorithms used in the application of the graphical models, we consider **Floyd-Warshall algorithm**

Problem

Find the minimal path in a weighted graph.

Let G be a **weighted oriented graph** on n vertices. Let d_{ij} be the **weight** of the arc joining i with j . $d_{ij} = \infty$ if there is no arc between i and j . $d_{ii} = 0$.

Floyd-Warshall algorithm

To give an idea of the dynamic programming algorithms used in the application of the graphical models, we consider **Floyd-Warshall algorithm**

Problem

Find the minimal path in a weighted graph.

Let G be a **weighted oriented graph** on n vertices. Let d_{ij} be the **weight** of the arc joining i with j . $d_{ij} = \infty$ if there is no arc between i and j . $d_{ii} = 0$. The matrix $D = (d_{ij})$, called the **adjacency matrix**, is defined in $\mathbb{R} \cup \{\infty\}$.

Floyd-Warshall algorithm

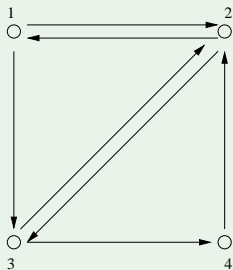
To give an idea of the dynamic programming algorithms used in the application of the graphical models, we consider **Floyd-Warshall algorithm**

Problem

Find the minimal path in a weighted graph.

Let G be a **weighted oriented graph** on n vertices. Let d_{ij} be the **weight** of the arc joining i with j . $d_{ij} = \infty$ if there is no arc between i and j . $d_{ij} = 0$. The matrix $D = (d_{ij})$, called the **adjacency matrix**, is defined in $\mathbb{R} \cup \{\infty\}$. Any non oriented weighted graph gives canonically rise to an oriented one with a symmetric adjacency matrix.

Example



$$\begin{pmatrix} 0 & d_{12} & d_{13} & \infty \\ d_{21} & 0 & d_{23} & \infty \\ \infty & d_{32} & 0 & d_{34} \\ \infty & d_{42} & \infty & 0 \end{pmatrix}$$

The algorithm

The **weight of a path** is the sum of the weights of all its arcs.

The algorithm

The **weight of a path** is the sum of the weights of all its arcs.

A **path of minimal length** between two vertices visits each vertex at most once. Hence it is made up with **at most** $n - 1$ arcs.

The algorithm

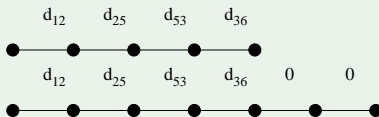
The **weight of a path** is the sum of the weights of all its arcs.

A **path of minimal length** between two vertices visits each vertex at most once. Hence it is made up with **at most** $n - 1$ arcs. We can assume it has length $n - 1$ by adding length zero arcs.

The algorithm

The **weight of a path** is the sum of the weights of all its arcs.

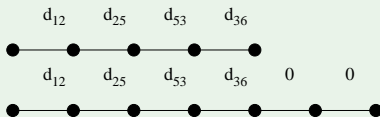
A **path of minimal length** between two vertices visits each vertex at most once. Hence it is made up with **at most** $n - 1$ arcs. We can assume it has length $n - 1$ by adding length zero arcs.



The algorithm

The **weight of a path** is the sum of the weights of all its arcs.

A **path of minimal length** between two vertices visits each vertex at most once. Hence it is made up with **at most** $n - 1$ arcs. We can assume it has length $n - 1$ by adding length zero arcs.



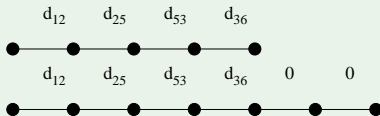
The weight of a minimal path joining vertex i with vertex j is

$$d_{ij}^{(n-1)} = \min_{\alpha_1, \dots, \alpha_n \in \{1, \dots, n\}} (d_{i\alpha_1} + d_{\alpha_1\alpha_2} + \dots + d_{\alpha_{n-2}j}). \quad (1)$$

The algorithm

The **weight of a path** is the sum of the weights of all its arcs.

A **path of minimal length** between two vertices visits each vertex at most once. Hence it is made up with **at most** $n - 1$ arcs. We can assume it has length $n - 1$ by adding length zero arcs.



The weight of a minimal path joining vertex i with vertex j is

$$d_{ij}^{(n-1)} = \min_{\alpha_1, \dots, \alpha_n \in \{1, \dots, n\}} (d_{i\alpha_1} + d_{\alpha_1\alpha_2} + \dots + d_{\alpha_{n-2}j}). \quad (1)$$

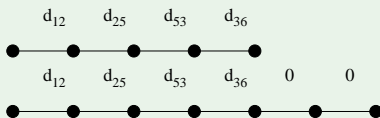
The iterative formula

$$d_{ij}^{(k)} = \min\{d_{ir}^{(k-1)} + d_{rj} : r = 1, 2, \dots, n\}, \quad (2)$$

is the **Floyd-Warshall Algorithm** for computing $d_{ij}^{(n-1)}$.

The algorithm

The **weight of a path** is the sum of the weights of all its arcs.
A **path of minimal length** between two vertices visits each vertex at most once. Hence it is made up with **at most** $n - 1$ arcs. We can assume it has length $n - 1$ by adding length zero arcs.



The weight of a minimal path joining vertex i with vertex j is

$$d_{ij}^{(n-1)} = \min_{\alpha_1, \dots, \alpha_n \in \{1, \dots, n\}} (d_{i\alpha_1} + d_{\alpha_1\alpha_2} + \dots + d_{\alpha_{n-2}j}). \quad (1)$$

The iterative formula

$$d_{ij}^{(k)} = \min\{d_{ir}^{(k-1)} + d_{rj} : r = 1, 2, \dots, n\}, \quad (2)$$

is the **Floyd-Warshall Algorithm** for computing $d_{ij}^{(n-1)}$.

Useful notation: $D^{\odot k} = (d_{ij}^{(k)})$. The element (i, j) of $D^{\odot(n-1)}$ is the length of the minimal path joining i with j .

Path length and matrix algebra

The iterative formula $d_{ij}^{(k)} = \min\{d_{ir}^{(k-1)} + d_{rj} : r = 1, 2, \dots, n\}$ can be usefully interpreted as the result of a **new kind of algebraic operation** over the adjacency matrix.

Path length and matrix algebra

The iterative formula $d_{ij}^{(k)} = \min\{d_{ir}^{(k-1)} + d_{rj} : r = 1, 2, \dots, n\}$ can be usefully interpreted as the result of a **new kind of algebraic operation** over the adjacency matrix. Let ϵ be an indeterminate and let $A(\epsilon)$ be the matrix whose (i, j) -th element is $\epsilon^{d_{ij}}$.

Path length and matrix algebra

The iterative formula $d_{ij}^{(k)} = \min\{d_{ir}^{(k-1)} + d_{rj} : r = 1, 2, \dots, n\}$ can be usefully interpreted as the result of a **new kind of algebraic operation** over the adjacency matrix. Let ϵ be an indeterminate and let $A(\epsilon)$ be the matrix whose (i, j) -th element is $\epsilon^{d_{ij}}$.

The matrix $A(\epsilon)^k$ gives information on **all** paths of k arcs which can be traveled in the graph G .

Path length and matrix algebra

The iterative formula $d_{ij}^{(k)} = \min\{d_{ir}^{(k-1)} + d_{rj} : r = 1, 2, \dots, n\}$ can be usefully interpreted as the result of a **new kind of algebraic operation** over the adjacency matrix. Let ϵ be an indeterminate and let $A(\epsilon)$ be the matrix whose (i, j) -th element is $\epsilon^{d_{ij}}$.

The matrix $A(\epsilon)^k$ gives information on **all** paths of k arcs which can be traveled in the graph G .

The ij element of the matrix $A(\epsilon)^k$ is

$$a_{ij}^{(k)}(\epsilon) = \sum_{h_1, \dots, h_{k-1}} \epsilon^{d_{ih_1} + \dots + d_{h_{k-1}j}}.$$

Path length and matrix algebra

The iterative formula $d_{ij}^{(k)} = \min\{d_{ir}^{(k-1)} + d_{rj} : r = 1, 2, \dots, n\}$ can be usefully interpreted as the result of a **new kind of algebraic operation** over the adjacency matrix. Let ϵ be an indeterminate and let $A(\epsilon)$ be the matrix whose (i, j) -th element is $\epsilon^{d_{ij}}$.

The matrix $A(\epsilon)^k$ gives information on **all** paths of k arcs which can be traveled in the graph G .

The ij element of the matrix $A(\epsilon)^k$ is

$$a_{ij}^{(k)}(\epsilon) = \sum_{h_1, \dots, h_{k-1}} \epsilon^{d_{ih_1} + \dots + d_{h_{k-1}j}}.$$

Hence, each path of length k joining the vertices $i, h_1, \dots, h_{k-1}, j$ contributes with $\epsilon^{d_{ih_1} + \dots + d_{h_{k-1}j}}$ at $a_{ij}^{(k)}(\epsilon)$.

Path length and matrix algebra

The iterative formula $d_{ij}^{(k)} = \min\{d_{ir}^{(k-1)} + d_{rj} : r = 1, 2, \dots, n\}$ can be usefully interpreted as the result of a **new kind of algebraic operation** over the adjacency matrix. Let ϵ be an indeterminate and let $A(\epsilon)$ be the matrix whose (i, j) -th element is $\epsilon^{d_{ij}}$.

The matrix $A(\epsilon)^k$ gives information on **all** paths of k arcs which can be traveled in the graph G .

The ij element of the matrix $A(\epsilon)^k$ is

$$a_{ij}^{(k)}(\epsilon) = \sum_{h_1, \dots, h_{k-1}} \epsilon^{d_{ih_1} + \dots + d_{h_{k-1}j}}.$$

Hence, each path of length k joining the vertices $i, h_1, \dots, h_{k-1}, j$ contributes with $\epsilon^{d_{ih_1} + \dots + d_{h_{k-1}j}}$ at $a_{ij}^{(k)}(\epsilon)$. The coefficient of this term is the number of paths joining i with j and having total weight equal to the exponent.

Floyd-Warshall Algorithm and matrix algebra

The matrix $A(\epsilon)^k$ contains much more information than the matrix $D^{\odot k}$.

Floyd-Warshall Algorithm and matrix algebra

The matrix $A(\epsilon)^k$ contains much more information than the matrix $D^{\odot k}$. $D^{\odot k}$ is the **tropicalization** of $A^k(\epsilon)$, in the sense that

$$\lim_{\epsilon \rightarrow 0} \log_{\epsilon} A(\epsilon)^k = D^{\odot k},$$

where, by $\log_{\epsilon} A(\epsilon)^k$ we mean the matrix whose entries are the logarithms of the corresponding entries of $A(\epsilon)^k$.

Floyd-Warshall Algorithm and matrix algebra

The matrix $A(\epsilon)^k$ contains much more information than the matrix $D^{\odot k}$. $D^{\odot k}$ is the **tropicalization** of $A^k(\epsilon)$, in the sense that

$$\lim_{\epsilon \rightarrow 0} \log_{\epsilon} A(\epsilon)^k = D^{\odot k},$$

where, by $\log_{\epsilon} A(\epsilon)^k$ we mean the matrix whose entries are the logarithms of the corresponding entries of $A(\epsilon)^k$.

Note how the **tropicalization process** restrict the attention to the minimal exponent and forget its coefficient.

Floyd-Warshall Algorithm and matrix algebra

The matrix $A(\epsilon)^k$ contains much more information than the matrix $D^{\odot k}$. $D^{\odot k}$ is the **tropicalization** of $A^k(\epsilon)$, in the sense that

$$\lim_{\epsilon \rightarrow 0} \log_{\epsilon} A(\epsilon)^k = D^{\odot k},$$

where, by $\log_{\epsilon} A(\epsilon)^k$ we mean the matrix whose entries are the logarithms of the corresponding entries of $A(\epsilon)^k$.

Note how the **tropicalization process** restrict the attention to the minimal exponent and forget its coefficient.

The tropicalization process **piecewise linearize** polynomial functions, and extract simpler but useful information from them.

Floyd-Warshall Algorithm and matrix algebra

The matrix $A(\epsilon)^k$ contains much more information than the matrix $D^{\odot k}$. $D^{\odot k}$ is the **tropicalization** of $A^k(\epsilon)$, in the sense that

$$\lim_{\epsilon \rightarrow 0} \log_{\epsilon} A(\epsilon)^k = D^{\odot k},$$

where, by $\log_{\epsilon} A(\epsilon)^k$ we mean the matrix whose entries are the logarithms of the corresponding entries of $A(\epsilon)^k$.

Note how the **tropicalization process** restrict the attention to the minimal exponent and forget its coefficient.

The tropicalization process **piecewise linearize** polynomial functions, and extract simpler but useful information from them.

The right framework for performing tropicalization is **tropical algebra**, where $D^{\odot k}$ becomes the k -th power of D under the tropical product.

Tropical Algebra

Tropical algebra can be thought of as being deduced by classical algebra by a **suitable limiting process**.

Tropical Algebra

Tropical algebra can be thought of as being deduced by classical algebra by a **suitable limiting process**.

It transforms **algebraic geometric objects**, i.e. polynomials and algebraic varieties, into **combinatorial** ones for which it is easier to read some of the important properties of the original object.

Tropical Algebra

Tropical algebra can be thought of as being deduced by classical algebra by a **suitable limiting process**.

It transforms **algebraic geometric objects**, i.e. polynomials and algebraic varieties, into **combinatorial** ones for which it is easier to read some of the important properties of the original object.

Recent applications of tropical algebraic geometry to enumerative problems are due to **Mikhalkin**.

Tropical Algebra

Tropical algebra can be thought of as being deduced by classical algebra by a **suitable limiting process**.

It transforms **algebraic geometric objects**, i.e. polynomials and algebraic varieties, into **combinatorial** ones for which it is easier to read some of the important properties of the original object.

Recent applications of tropical algebraic geometry to enumerative problems are due to **Mikhalkin**.

Basic for tropical algebra is the **tropical semiring**

$$\mathbb{R}_{max} = (\mathbb{R} \cup -\infty, \oplus, \odot)$$

where the operations are defined in the following way:

Tropical Algebra

Tropical algebra can be thought of as being deduced by classical algebra by a **suitable limiting process**.

It transforms **algebraic geometric objects**, i.e. polynomials and algebraic varieties, into **combinatorial** ones for which it is easier to read some of the important properties of the original object.

Recent applications of tropical algebraic geometry to enumerative problems are due to **Mikhalkin**.

Basic for tropical algebra is the **tropical semiring**

$$\mathbb{R}_{max} = (\mathbb{R} \cup -\infty, \oplus, \odot)$$

where the operations are defined in the following way:

- the **tropical sum** \oplus is **max**;

Tropical Algebra

Tropical algebra can be thought of as being deduced by classical algebra by a **suitable limiting process**.

It transforms **algebraic geometric objects**, i.e. polynomials and algebraic varieties, into **combinatorial** ones for which it is easier to read some of the important properties of the original object.

Recent applications of tropical algebraic geometry to enumerative problems are due to **Mikhalkin**.

Basic for tropical algebra is the **tropical semiring**

$$\mathbb{R}_{max} = (\mathbb{R} \cup -\infty, \oplus, \odot)$$

where the operations are defined in the following way:

- the **tropical sum** \oplus is **max**;
- the **tropical product** \odot is **usual sum**.

Tropical Algebra

Tropical algebra can be thought of as being deduced by classical algebra by a **suitable limiting process**.

It transforms **algebraic geometric objects**, i.e. polynomials and algebraic varieties, into **combinatorial** ones for which it is easier to read some of the important properties of the original object.

Recent applications of tropical algebraic geometry to enumerative problems are due to **Mikhalkin**.

Basic for tropical algebra is the **tropical semiring**

$$\mathbb{R}_{max} = (\mathbb{R} \cup -\infty, \oplus, \odot)$$

where the operations are defined in the following way:

- the **tropical sum** \oplus is **max**;
- the **tropical product** \odot is **usual sum**.

The **neutral element** for \oplus is $-\infty$ and that for \odot is 0.

Tropical Algebra

Tropical algebra can be thought of as being deduced by classical algebra by a **suitable limiting process**.

It transforms **algebraic geometric objects**, i.e. polynomials and algebraic varieties, into **combinatorial** ones for which it is easier to read some of the important properties of the original object.

Recent applications of tropical algebraic geometry to enumerative problems are due to **Mikhalkin**.

Basic for tropical algebra is the **tropical semiring**

$$\mathbb{R}_{max} = (\mathbb{R} \cup -\infty, \oplus, \odot)$$

where the operations are defined in the following way:

- the **tropical sum** \oplus is **max**;
- the **tropical product** \odot is **usual sum**.

The **neutral element** for \oplus is $-\infty$ and that for \odot is 0.

The tropical semiring **is not a ring** since the sum does not have opposite.

Tropical linear algebra

There exists a completely analogous tropical semiring

$$\mathbb{R}_{min} = (\mathbb{R} \cup \infty, \oplus, \odot)$$

where \oplus is **min**.

Tropical linear algebra

There exists a completely analogous tropical semiring

$$\mathbb{R}_{min} = (\mathbb{R} \cup \infty, \oplus, \odot)$$

where \oplus is **min**.

Changing sign gives an isomorphism between \mathbb{R}_{min} and \mathbb{R}_{max} .

Tropical linear algebra

There exists a completely analogous tropical semiring

$$\mathbb{R}_{min} = (\mathbb{R} \cup \infty, \oplus, \odot)$$

where \oplus is **min**.

Changing sign gives an isomorphism between \mathbb{R}_{min} and \mathbb{R}_{max} .

Usual vector and matrix operations have **tropical analogues**.

Tropical linear algebra

There exists a completely analogous tropical semiring

$$\mathbb{R}_{min} = (\mathbb{R} \cup \infty, \oplus, \odot)$$

where \oplus is **min**.

Changing sign gives an isomorphism between \mathbb{R}_{min} and \mathbb{R}_{max} .

Usual vector and matrix operations have **tropical analogues**. In \mathbb{R}_{min} the iterative formula for $d_{i,j}^{(k)}$ is equivalent to

Tropical linear algebra

There exists a completely analogous tropical semiring

$$\mathbb{R}_{min} = (\mathbb{R} \cup \infty, \oplus, \odot)$$

where \oplus is **min**.

Changing sign gives an isomorphism between \mathbb{R}_{min} and \mathbb{R}_{max} .

Usual vector and matrix operations have **tropical analogues**. In \mathbb{R}_{min} the iterative formula for $d_{i,j}^{(k)}$ is equivalent to

$$D^{\odot k} = D^{\odot(k-1)} \odot D.$$

Tropical polynomials

In the tropical semiring one can define **tropical polynomials**. Their interpretation is in term of **piecewise linear functions**.

Tropical polynomials

In the tropical semiring one can define **tropical polynomials**. Their interpretation is in term of **piecewise linear functions**. For example the polynomial

$$a \odot x^{\odot 2} \oplus b \odot x \odot y \oplus c \odot y^{\odot 2} \oplus d \odot x \oplus e \odot y \oplus f \quad (3)$$

Tropical polynomials

In the tropical semiring one can define **tropical polynomials**. Their interpretation is in term of **piecewise linear functions**. For example the polynomial

$$a \odot x^{\odot 2} \oplus b \odot x \odot y \oplus c \odot y^{\odot 2} \oplus d \odot x \oplus e \odot y \oplus f \quad (3)$$

corresponds to the piecewise linear function

$$\max\{2x + a, x + y + b, 2y + c, x + d, y + e, f\}.$$

Tropical polynomials

In the tropical semiring one can define **tropical polynomials**. Their interpretation is in term of **piecewise linear functions**. For example the polynomial

$$a \odot x^{\odot 2} \oplus b \odot x \odot y \oplus c \odot y^{\odot 2} \oplus d \odot x \oplus e \odot y \oplus f \quad (3)$$

corresponds to the piecewise linear function

$$\max\{2x + a, x + y + b, 2y + c, x + d, y + e, f\}.$$

The **corner locus** of a tropical polynomial is the set of points where the corresponding picewise linear function is not differentiable.

Tropical polynomials

In the tropical semiring one can define **tropical polynomials**. Their interpretation is in term of **piecewise linear functions**. For example the polynomial

$$a \odot x^{\odot 2} \oplus b \odot x \odot y \oplus c \odot y^{\odot 2} \oplus d \odot x \oplus e \odot y \oplus f \quad (3)$$

corresponds to the piecewise linear function

$$\max\{2x + a, x + y + b, 2y + c, x + d, y + e, f\}.$$

The **corner locus** of a tropical polynomial is the set of points where the corresponding picewise linear function is not differentiable. The corner locus is the tropical analogue of the zero locus of the polynomial and is called the **tropicalization** of the zero locus.

Tropical polynomials

In the tropical semiring one can define **tropical polynomials**. Their interpretation is in term of **piecewise linear functions**. For example the polynomial

$$a \odot x^{\odot 2} \oplus b \odot x \odot y \oplus c \odot y^{\odot 2} \oplus d \odot x \oplus e \odot y \oplus f \quad (3)$$

corresponds to the piecewise linear function

$$\max\{2x + a, x + y + b, 2y + c, x + d, y + e, f\}.$$

The **corner locus** of a tropical polynomial is the set of points where the corresponding picewise linear function is not differentiable. The corner locus is the tropical analogue of the zero locus of the polynomial and is called the **tropicalization** of the zero locus. With some care one can also **tropicalize** any variety and get a polyhedral complex of the same dimension.

Tropical polynomials

In the tropical semiring one can define **tropical polynomials**. Their interpretation is in term of **piecewise linear functions**. For example the polynomial

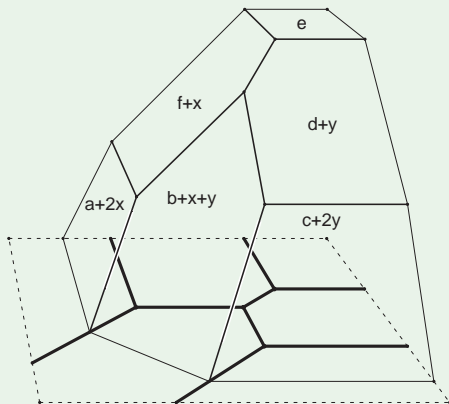
$$a \odot x^{\odot 2} \oplus b \odot x \odot y \oplus c \odot y^{\odot 2} \oplus d \odot x \oplus e \odot y \oplus f \quad (3)$$

corresponds to the piecewise linear function

$$\max\{2x + a, x + y + b, 2y + c, x + d, y + e, f\}.$$

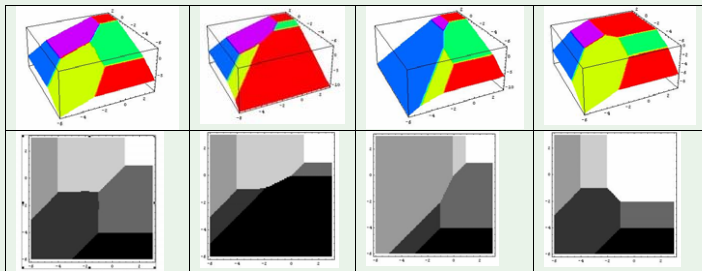
The **corner locus** of a tropical polynomial is the set of points where the corresponding picewise linear function is not differentiable. The corner locus is the tropical analogue of the zero locus of the polynomial and is called the **tropicalization** of the zero locus. With some care one can also **tropicalize** any variety and get a polyhedral complex of the same dimension. The corner locus of a tropical polynomial of degree d in n variables is called **tropical hypersurface of degree d** .

Tropical conics

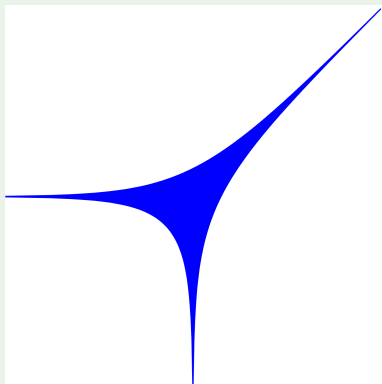


Proper tropical conics

There are four types of **proper** tropical conics.



Amoebas

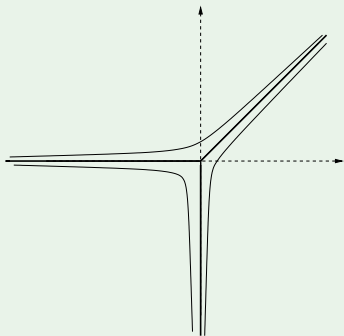


Amoeba of a line



Amoeba of cubic

A tropical curve as a limit of an amoeba



Phylogenetic

Phylogenetic studies **evolution models** according to Darwin's theory.

Phylogenetic

Phylogenetic studies **evolution models** according to Darwin's theory. The probabilistic models used in phylogenetic are **natural generalizations** of Hidden Markov Models.

Phylogenetic

Phylogenetic studies **evolution models** according to Darwin's theory. The probabilistic models used in phylogenetic are **natural generalizations** of Hidden Markov Models. They give rise to **new algebraic varieties**.

Phylogenetic

Phylogenetic studies **evolution models** according to Darwin's theory. The probabilistic models used in phylogenetic are **natural generalizations** of Hidden Markov Models. They give rise to **new algebraic varieties**.

The basic mathematical structure in phylogenetic is that of a **phylogeny**, i. e. a tree (connected graph with no circuits) with labeled leaves.

Phylogenetic

Phylogenetic studies **evolution models** according to Darwin's theory. The probabilistic models used in phylogenetic are **natural generalizations** of Hidden Markov Models. They give rise to **new algebraic varieties**.

The basic mathematical structure in phylogenetic is that of a **phylogeny**, i. e. a tree (connected graph with no circuits) with labeled leaves.

For biological reasons it is important to introduce a notion of **weight** for arcs in a phylogeny for measuring dissimilarities of adjacent species.

Phylogenetic

Phylogenetic studies **evolution models** according to Darwin's theory. The probabilistic models used in phylogenetic are **natural generalizations** of Hidden Markov Models. They give rise to **new algebraic varieties**.

The basic mathematical structure in phylogenetic is that of a **phylogeny**, i. e. a tree (connected graph with no circuits) with labeled leaves.

For biological reasons it is important to introduce a notion of **weight** for arcs in a phylogeny for measuring dissimilarities of adjacent species.

Fundamental problem of phylogenetic

Given a set of observed data about a certain set of living species (usually biochemical data), find the **best** phylogeny which explains the data

Phylogenetic

Phylogenetic studies **evolution models** according to Darwin's theory. The probabilistic models used in phylogenetic are **natural generalizations** of Hidden Markov Models. They give rise to **new algebraic varieties**.

The basic mathematical structure in phylogenetic is that of a **phylogeny**, i. e. a tree (connected graph with no circuits) with labeled leaves.

For biological reasons it is important to introduce a notion of **weight** for arcs in a phylogeny for measuring dissimilarities of adjacent species.

Fundamental problem of phylogenetic

Given a set of observed data about a certain set of living species (usually biochemical data), find the **best** phylogeny which explains the data

The space of phylogenies with weighted arcs is a **metric space** whose combinatorial structure is of the utmost importance for applications.

Applications of tropical geometry to phylogenetic

The space of phylogenies with weighted arcs is the tropicalization of a classical algebraic variety, the **grassmannian of lines in projective space**.

Applications of tropical geometry to phylogenetic

The space of phylogenies with weighted arcs is the tropicalization of a classical algebraic variety, the **grassmannian of lines in projective space**.

The geometric structure of the space of all phylogenies provides important help in guiding the **resolution of ambiguities** in phylogeny reconstruction.

Applications of tropical geometry to phylogenetic

The space of phylogenies with weighted arcs is the tropicalization of a classical algebraic variety, the **grassmannian of lines in projective space**.

The geometric structure of the space of all phylogenies provides important help in guiding the **resolution of ambiguities** in phylogeny reconstruction.

An important need of modern phylogenetic is to improve the accuracy of phylogeny reconstruction, by generalizing the concept of dissimilarities to **more than two species**.

Applications of tropical geometry to phylogenetic

The space of phylogenies with weighted arcs is the tropicalization of a classical algebraic variety, the **grassmannian of lines in projective space**.

The geometric structure of the space of all phylogenies provides important help in guiding the **resolution of ambiguities** in phylogeny reconstruction.

An important need of modern phylogenetic is to improve the accuracy of phylogeny reconstruction, by generalizing the concept of dissimilarities to **more than two species**. **Tropical grassmannian of k spaces** seems to provide a suitable framework for this generalization.

Biology asks for deep and new mathematics.

Biology asks for deep and new mathematics.

Will be possible that one of the next winners of a Field Medal will be a biologist?