

MP, ML, AML Reconstruction of Phylogenetic Trees: A Status Report

Benny Chor

School of Computer Science
Tel-Aviv University

Phylogenetic Reconstruction

- Input: A set of n aligned sequences (genes, proteins) from n species,

Phylogenetic Reconstruction

- Input: A set of n aligned sequences (genes, proteins) from n species,
- Goal: Reconstruct the **tree** which **best explains** the evolutionary history of this gene/protein.

Phylogenetic Reconstruction

- Input: A set of n aligned sequences (genes, proteins) from n species,
- Goal: Reconstruct the **tree** which **best explains** the evolutionary history of this gene/protein.
- Tree reconstruction is still a challenge today.

Phylogenetic Reconstruction

- Input: A set of n aligned sequences (genes, proteins) from n species,
- Goal: Reconstruct the **tree** which **best explains** the evolutionary history of this gene/protein.
- Tree reconstruction is still a challenge today.
- Many concrete questions are still unresolved (*e.g.* mammalian evolutionary tree).

Phylogenetic Reconstruction

- Input: A set of n aligned sequences (genes, proteins) from n species,
- Goal: Reconstruct the **tree** which **best explains** the evolutionary history of this gene/protein.
- Tree reconstruction is still a challenge today.
- Many concrete questions are still unresolved (*e.g.* mammalian evolutionary tree).
- Most realistic formulations of the problem, which take errors into account, give rise to **hard computational problems**.

Popular Methods

- Distance based methods:

Popular Methods

- Distance based methods:
 - UPGMA

Popular Methods

- Distance based methods:
 - UPGMA
 - Neighbor Joining.

Popular Methods

- Distance based methods:
 - UPGMA
 - Neighbor Joining.
 - Buneman trees.

Popular Methods

- Distance based methods:
 - UPGMA
 - Neighbor Joining.
 - Buneman trees.
- Character Based Methods:

Popular Methods

- Distance based methods:
 - UPGMA
 - Neighbor Joining.
 - Buneman trees.
- Character Based Methods:
 - Maximum Parsimony.

Popular Methods

- Distance based methods:
 - UPGMA
 - Neighbor Joining.
 - Buneman trees.
- Character Based Methods:
 - Maximum Parsimony.
 - **Maximum Likelihood.**

Popular Methods

- Distance based methods:
 - UPGMA
 - Neighbor Joining.
 - Buneman trees.
- Character Based Methods:
 - Maximum Parsimony.
 - **Maximum Likelihood.**
- Additional Methods:

Popular Methods

- Distance based methods:
 - UPGMA
 - Neighbor Joining.
 - Buneman trees.
- Character Based Methods:
 - Maximum Parsimony.
 - **Maximum Likelihood.**
- Additional Methods:
 - Quartets Based.

Popular Methods

- Distance based methods:
 - UPGMA
 - Neighbor Joining.
 - Buneman trees.
- Character Based Methods:
 - Maximum Parsimony.
 - **Maximum Likelihood.**
- Additional Methods:
 - Quartets Based.
 - Disc Covering.

Talk Outline

- Maximum likelihood (ML).

Talk Outline

- Maximum likelihood (ML).
- The likelihood surface.

Talk Outline

- Maximum likelihood (ML).
- The likelihood surface.
- Existence of multiple maxima.

Talk Outline

- Maximum likelihood (ML).
- The likelihood surface.
- Existence of multiple maxima.
- Computation complexity: Maximum likelihood vs. maximum **parsimony** (MP).

Talk Outline

- Maximum likelihood (ML).
- The likelihood surface.
- Existence of multiple maxima.
- Computation complexity: Maximum likelihood vs. maximum **parsimony** (MP).
- **Ancestral** maximum likelihood (AML) and its computational complexity.

Maximum Likelihood

- **Input:** A set of n **observed sequences** and an underlying substitution **model**.

Maximum Likelihood

- **Input:** A set of n observed sequences and an underlying substitution model.
- **Desired Output:** The weighted tree T that maximizes the likelihood of the data.

Maximum Likelihood

- **Input:** A set of n **observed sequences** and an underlying substitution **model**.
- **Desired Output:** The weighted tree T that maximizes the **likelihood** of the data.
- **Likelihood** of a data: The conditional probability of producing the data, given the model parameters.

Maximum Likelihood

- **Input:** A set of n **observed sequences** and an underlying substitution **model**.
- **Desired Output:** The weighted tree T that maximizes the **likelihood** of the data.
- **Likelihood** of a data: The conditional probability of producing the data, given the model parameters.
- Likelihood is a common optimization criteria in numerous settings, including phylogenetic (Felsenstein 1981).

2–State Substitution Model

species	observed data
1	XXXXXXXXXXYY Y XX Y XY YX XY X
2	XXXXXXXXXXYY Y YY X YX YX YX X
3	XXXXXXXXXXYY Y YY X XY XY XY X
4	XXXXXXXXXXYY Y YY X XY XY YX Y

- Just **two** characters states, **X** and **Y**.

2-State Substitution Model

species	observed data
1	XXXXXXXXYY Y XX Y XY YX XY X
2	XXXXXXXXYY Y YYX YX YX YX X
3	XXXXXXXXYY Y YYX XY XY XY X
4	XXXXXXXXYY Y YYX XY XY YX Y

- Just **two** characters states, **X** and **Y**.
- Transitions between states are symmetric.

2–State Substitution Model

species	observed data
1	XXXXXXXXXXYY Y XX Y XY YX XY X
2	XXXXXXXXXXYY Y YY X YX YX YX X
3	XXXXXXXXXXYY Y YY X XY XY XY X
4	XXXXXXXXXXYY Y YY X XY XY YX Y

- Just **two** characters states, **X** and **Y**.
- Transitions between states are symmetric.
- Equal rates across sites.

2–State Substitution Model

species	observed data
1	XXXXXXXXXXYY Y XX Y XY YX XY X
2	XXXXXXXXXXYY Y YY X YX YX YX X
3	XXXXXXXXXXYY Y YY X XY XY XY X
4	XXXXXXXXXXYY Y YY X XY XY YX Y

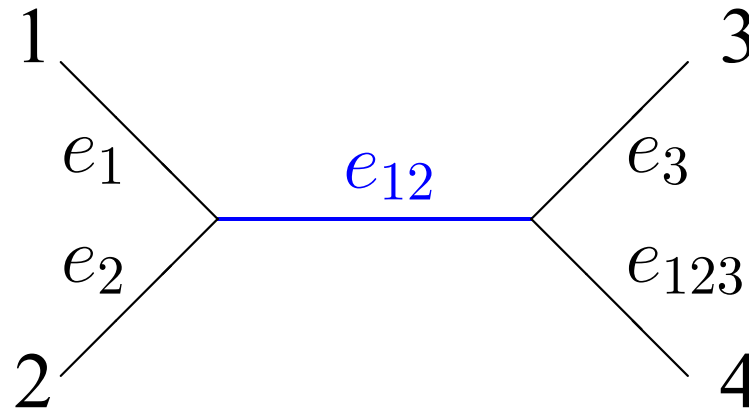
- Just **two** characters states, **X** and **Y**.
- Transitions between states are symmetric.
- Equal rates across sites.
- Every column induces a **pattern**.

2–State Substitution Model

species	observed data
1	XXXXXXXXXXYY Y XX Y XY YX XY X
2	XXXXXXXXXXYY Y YY X YX YX YX X
3	XXXXXXXXXXYY Y YY X XY XY XY X
4	XXXXXXXXXXYY Y YY X XY XY YX Y

- Just **two** characters states, **X** and **Y**.
- Transitions between states are symmetric.
- Equal rates across sites.
- Every column induces a **pattern**.
- **Remark:** A simple model, yet very powerful.

Neyman 2-State Substitution Model



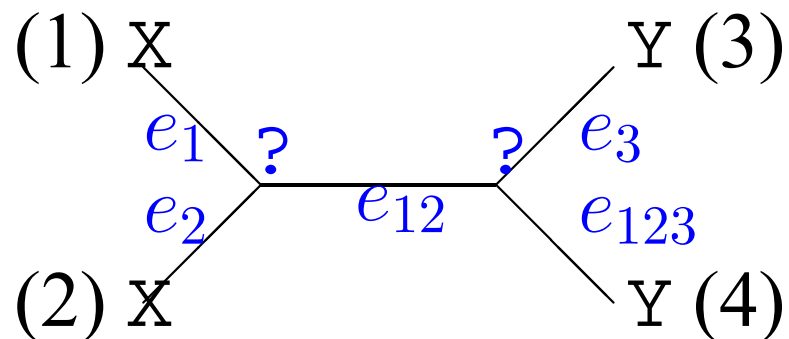
For each edge e of a tree T , the edge weight p_e represents the probability of having **different states** at the two ends of e .

A **Very** Simple Example

Four species ($n = 4$), just **one** site ($c = 1$)

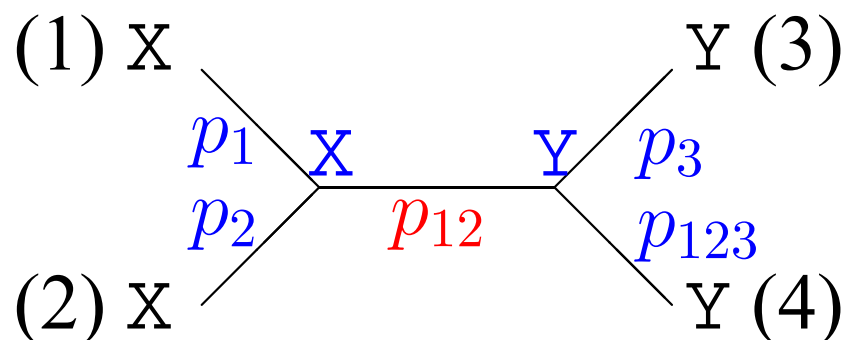
species	observed data
1	X
2	X
3	Y
4	Y

Analyze the *natural* tree (12)(34)



Computing the Likelihood

Each unknown state (?) can assume **one of two** possibilities, **X** or **Y**. For example, the assignment



contributes $(1 - p_1) \cdot (1 - p_2) \cdot p_{12} \cdot (1 - p_3) \cdot (1 - p_{123})$.

The likelihood is the sum of this

+ **three** similar expressions. . .

Computing the Likelihood (2)

$$L(\text{ data } | T, \text{ edge parameters}) \\ \triangleq \sum_{\text{internal assignments}} \prod_{\text{edges}} p^{d_e} (1 - p)^{\ell - d_e} .$$

Each d_e is number of unequal sites along edge e . It depends on the internal assignment a , and input pattern t at two ends of the edge.

Computing the Likelihood (2)

$$L(\text{data} \mid T, \text{edge parameters}) \\ \triangleq \sum_{\text{internal assignments}} \prod_{\text{edges}} p^{d_e} (1 - p)^{\ell - d_e} .$$

Each d_e is number of unequal sites along edge e . It depends on the internal assignment a , and input pattern t at two ends of the edge.

- A well defined objective function to maximize.
- Termed **average likelihood** by Penny and Steel.
- Widely used in practice.

Three Likelihood Versions

- **Big Likelihood**: Given the sequence data, find a tree and edge weights that maximize $L(\text{data}|\text{tree \& edge weights})$.

Three Likelihood Versions

- **Big Likelihood**: Given the sequence data, find a tree and edge weights that maximize $L(\text{data}|\text{tree \& edge weights})$.
- **Small Likelihood**: Given observed data & a tree, but not the edge weights, find the edge weights that maximize the likelihood.

Three Likelihood Versions

- **Big Likelihood**: Given the sequence data, find a tree and edge weights that maximize $L(\text{data}|\text{tree \& edge weights})$.
- **Small Likelihood**: Given observed data & a tree, but not the edge weights, find the edge weights that maximize the likelihood.
- **Tiny Likelihood**: Given observed data & a tree & edge weights, find the likelihood.

Three Likelihood Versions

- **Big Likelihood**: Given the sequence data, find a tree and edge weights that maximize $L(\text{data}|\text{tree \& edge weights})$.
- **Small Likelihood**: Given observed data & a tree, but not the edge weights, find the edge weights that maximize the likelihood.
- **Tiny Likelihood**: Given observed data & a tree & edge weights, find the likelihood.
- Tiny likelihood can be efficiently computed using dynamic programming (Felsenstein, 1981).

Hill Climbing / Small Likelihood

- Typical approach to small likelihood, used in practice:

Hill Climbing / Small Likelihood

- Typical approach to small likelihood, used in practice:
- Start at some initial point with edge weights \mathbf{p} .

Hill Climbing / Small Likelihood

- Typical approach to small likelihood, used in practice:
- Start at some initial point with edge weights \mathbf{p} .
- Apply **hill climbing** to the likelihood function, till reaching a **maximum**.

The Likelihood Surface

- For hill climbing to be guaranteed to find the maximum, there must be a **single** *local and global* **maximum** in the parameter space.

The Likelihood Surface

- For hill climbing to be guaranteed to find the maximum, there must be a **single local and global maximum** in the parameter space.
- Fukami and Tateno (89), Tillier (94): For any tree, the ML point will be **unique**.

The Likelihood Surface

- For hill climbing to be guaranteed to find the maximum, there must be a **single local and global maximum** in the parameter space.
- Fukami and Tateno (89), Tillier (94): For any tree, the ML point will be **unique**.
- Steel (94): Proofs are erroneous - A simple but pathological **counter example** (multiple maxima on the **wrong tree**).

The Likelihood Surface

- For hill climbing to be guaranteed to find the maximum, there must be a **single local and global maximum** in the parameter space.
- Fukami and Tateno (89), Tillier (94): For any tree, the ML point will be **unique**.
- Steel (94): Proofs are erroneous - A simple but pathological **counter example** (multiple maxima on the **wrong tree**).
- (94–present): Hill climbing techniques still used. Steel's counter example is considered too “biologically unrealistic” to warrant concern.

The Likelihood Surface (cont.)

- Rogers and Swofford (99): Simulation Study

The Likelihood Surface (cont.)

- Rogers and Swofford (99): Simulation Study
 - Data is simulated on a tree.

The Likelihood Surface (cont.)

- Rogers and Swofford (99): Simulation Study
 - Data is simulated on a tree.
 - Multiple optima are rare...

The Likelihood Surface (cont.)

- Rogers and Swofford (99): Simulation Study
 - Data is simulated on a tree.
 - Multiple optima are rare...
 - ...especially on the *correct* tree.

The Likelihood Surface (cont.)

- Rogers and Swofford (99): Simulation Study
 - Data is simulated on a tree.
 - Multiple optima are rare...
 - ...especially on the *correct* tree.
- Goal here: Investigate the problem **analytically** (joint work with Hendy, Holland, Penny).

Maximizing Likelihood on Trees

Tools used

- Hadamard conjugation (Hendy and Penny 93).

Maximizing Likelihood on Trees

Tools used

- Hadamard conjugation (Hendy and Penny 93).
- Splits and sequence spectra (change of variables)

Maximizing Likelihood on Trees

Tools used

- Hadamard conjugation (Hendy and Penny 93).
- Splits and sequence spectra (change of variables)
- Constrained optimization.

Maximizing Likelihood on Trees

Tools used

- Hadamard conjugation (Hendy and Penny 93).
- Splits and sequence spectra (change of variables)
- Constrained optimization.
- Systems of **polynomial equations**.

Maximizing Likelihood on Trees

Tools used

- Hadamard conjugation (Hendy and Penny 93).
- Splits and sequence spectra (change of variables)
- Constrained optimization.
- Systems of **polynomial equations**.
- Analytical solution: very hard in general, even for **four taxa**.

Maximizing Likelihood on Trees

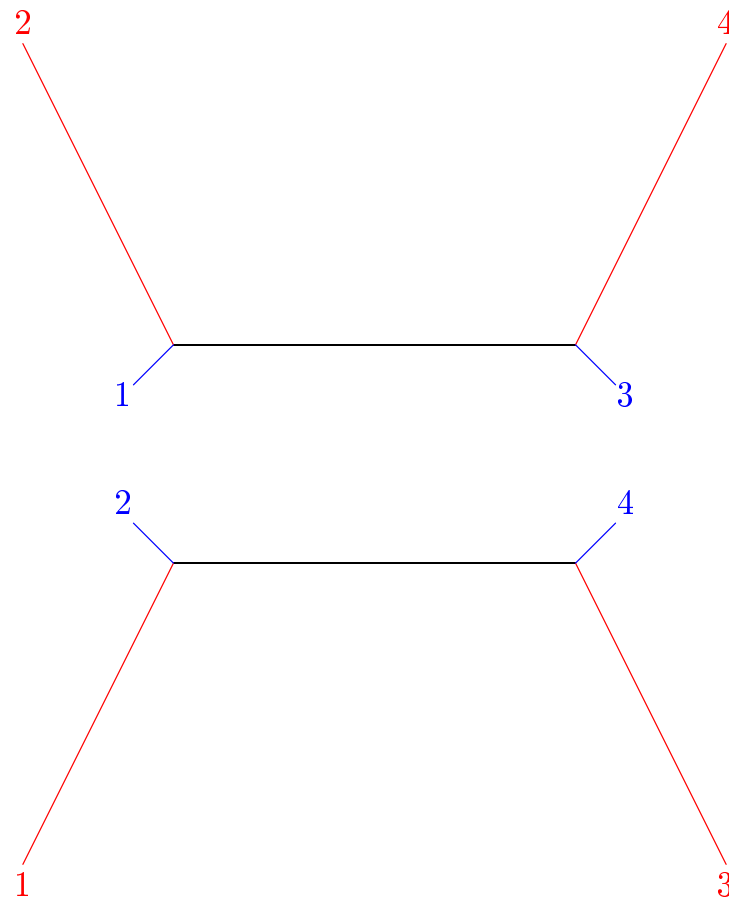
Tools used

- Hadamard conjugation (Hendy and Penny 93).
- Splits and sequence spectra (change of variables)
- Constrained optimization.
- Systems of **polynomial equations**.
- Analytical solution: very hard in general, even for **four taxa**.
- Employing computer algebra and algebraic geometry tools.

Example: Conservative Data, Two **Very Different** ML Trees

species	observed data
1	XXXXXXXXXXYY Y XX Y XY YX XY X
2	XXXXXXXXXXYY Y YY X YX YX YX X
3	XXXXXXXXXXYY Y YY X XY XY XY X
4	XXXXXXXXXXYY Y YY X XY XY YX Y

Example: Conservative Data, Two **Very Different** ML Trees



Small Likelihood & Multiple Maxima

- **Small Likelihood** (reminder): Given **observed data & a tree**, but **not the edge weights**, find the edge weights that maximize the likelihood.

Small Likelihood & Multiple Maxima

- **Small Likelihood** (reminder): Given **observed data & a tree**, but **not the edge weights**, find the edge weights that maximize the likelihood.
- Multiple ML points for **general case** imply small likelihood **cannot be solved by hill climbing**.

Small Likelihood & Multiple Maxima

- **Small Likelihood** (reminder): Given **observed data & a tree**, but **not the edge weights**, find the edge weights that maximize the likelihood.
- Multiple ML points for **general case** imply small likelihood **cannot be solved by hill climbing**.
- Not clear if small likelihood has efficient (worst case) solutions.

Maximum Parsimony (MP)

- **Big Parsimony:** Given the sequence data, find a tree and assignment of sequences to internal nodes that minimizes the number of changes across all edges.

Maximum Parsimony (MP)

- **Big Parsimony:** Given the sequence data, find a tree and assignment of sequences to internal nodes that minimizes the number of changes across all edges.
- **Small Parsimony:** Given the sequence data and a tree, find internal assignment(s) that minimizes total number of changes.

Maximum Parsimony (MP)

- **Big Parsimony**: Given the sequence data, find a tree and assignment of sequences to internal nodes that minimizes the number of changes across all edges.
- **Small Parsimony**: Given the sequence data and a tree, find internal assignment(s) that minimizes total number of changes.
- MP considered by practitioners easier than ML. Indeed **small parsimony** has efficient algorithms (Fitch 1971, Sankoff and Cedergren 1983).

Complexity of Reconstruction

- Both MP and ML have well-defined objective functions
 - ⇒ Reconstruction is a **computational problem**.

Complexity of Reconstruction

- Both MP and ML have well-defined objective functions
 - ⇒ Reconstruction is a **computational problem**.
- Number of trees over n leaves is **exponential** in n
 - ⇒ Cannot **exhaustively search** all trees.

Complexity: Small MP vs. ML

- Small parsimony is in P.

Complexity: Small MP vs. ML

- Small parsimony is in P.
- Small likelihood – **unknown**.

Complexity: Big MP vs. ML

Is ML Computationally Intractable?

- **Big MP** known for almost 20 years to be **computationally intractable** [Day *et al.*, 1986, reduction from **vertex cover**].
- No such result has been found for **Big ML** to date (2004).
- Tuffley and Steel (1997): Relations between likelihood and parsimony.
- Addario-Berry *et al.* (2003): **Big Ancestral ML** is hard.
- Still, no cigar (and **not even close**).

Ancestral ML (AML)

- A tree reconstruction method that is “in between” **ML** and **MP**.

Ancestral ML (AML)

- A tree reconstruction method that is “in between” **ML** and **MP**.
- The goal is to simultaneously find edge weights and **assignment of sequences to internal nodes** so that the likelihood of the data, given the tree parameters, is maximized.

Ancestral ML (AML)

- A tree reconstruction method that is “in between” **ML** and **MP**.
- The goal is to simultaneously find edge weights and **assignment of sequences to internal nodes** so that the likelihood of the data, given the tree parameters, is maximized.
- AML is widely used in evolutionary studies.

Ancestral ML (AML)

- A tree reconstruction method that is “in between” **ML** and **MP**.
- The goal is to simultaneously find edge weights and **assignment of sequences to internal nodes** so that the likelihood of the data, given the tree parameters, is maximized.
- AML is widely used in evolutionary studies.
- Also termed **joint reconstruction of ancestral sequences**.

Ancestral ML (AML)

- A tree reconstruction method that is “in between” **ML** and **MP**.
- The goal is to simultaneously find edge weights and **assignment of sequences to internal nodes** so that the likelihood of the data, given the tree parameters, is maximized.
- AML is widely used in evolutionary studies.
- Also termed **joint reconstruction of ancestral sequences**.
- AML computes the likelihood contribution resulting from **best assignment** to internal nodes, while “regular ML” sums up over **all assignments**.

Two AML Versions

- **Big AML**: Given the sequence data, find a **tree**, assignment to **internal nodes**, and **edge weights** that maximize the likelihood of the data.

Two AML Versions

- **Big AML**: Given the sequence data, find a **tree**, assignment to **internal nodes**, and **edge weights** that maximize the likelihood of the data.
- **Small AML**: Given **observed data**, a **tree** and **edge weights**, but **not** the internal assignment, find the assignment that maximize the likelihood.

Two AML Versions

- **Big AML**: Given the sequence data, find a **tree**, assignment to **internal nodes**, and **edge weights** that maximize the likelihood of the data.
- **Small AML**: Given **observed data**, a **tree** and **edge weights**, but **not** the internal assignment, find the assignment that maximize the likelihood.
- PPSG 2000: A poly time, dynamic programming algorithm for **small AML**.

Two AML Versions

- **Big AML**: Given the sequence data, find a **tree**, assignment to **internal nodes**, and **edge weights** that maximize the likelihood of the data.
- **Small AML**: Given **observed data**, a **tree** and **edge weights**, but **not** the internal assignment, find the assignment that maximize the likelihood.
- PPSG 2000: A poly time, dynamic programming algorithm for **small AML**.
- Remark: Version where tree is given but no edge weights or assignment is still open.

Two AML Versions

- **Big AML**: Given the sequence data, find a **tree**, assignment to **internal nodes**, and **edge weights** that maximize the likelihood of the data.
- **Small AML**: Given **observed data**, a **tree** and **edge weights**, but **not** the internal assignment, find the assignment that maximize the likelihood.
- PPSG 2000: A poly time, dynamic programming algorithm for **small AML**.
- Remark: Version where tree is given but no edge weights or assignment is still open.
- ACHLPW 2003: **Big AML** is NP-hard.

Open Problems as of 2004

- Hardness proof for **big AML** as a stepping stone for **big ML**?

Open Problems as of 2004

- Hardness proof for **big AML** as a stepping stone for **big ML**?
- Is **small ML** in poly-time?

Our Major Question

Is ML Computationally Intractable?

- **MP** known for almost 20 years to be **computationally intractable** [Day *et al.*, 1986, translation from **vertex cover**].
- No such result has been found for **ML** to date.
- Tuffley and Steel (1997): Relations between likelihood and parsimony.
- Addario-Berry *et al.* (2003): **Ancestral** ML is hard.
- Still, no cigar (and **not even close**).

Is ML Computationally Intractable?

- Still, no cigar (and **not even close**).
- Particularly frustrating in light of intuition among practitioners that **ML** is **harder** than **MP**.
- Maybe some slick and efficient **ML** algorithm **lurks out there**, waiting to be discovered?

Is ML Computationally Intractable?

- Still, no cigar (and **not even close**).
- Particularly frustrating in light of intuition among practitioners that **ML** is **harder** than **MP**.
- Maybe some slick and efficient **ML** algorithm **lurks out there**, waiting to be discovered?

CT2005:

ML is computationally hard (**NP complete**)

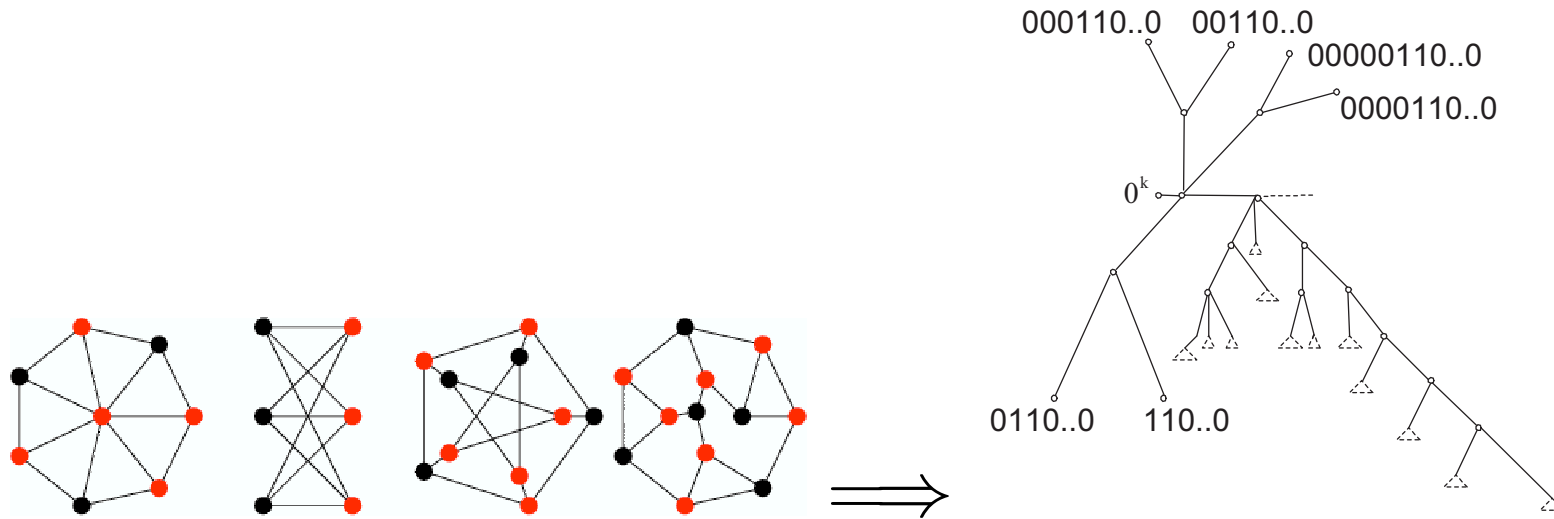
⇒ No such algorithm exists (**unless P=NP**).

Intractability Proof: The Big Picture

Efficiently translate **vertex cover (VC)** to **ML**.

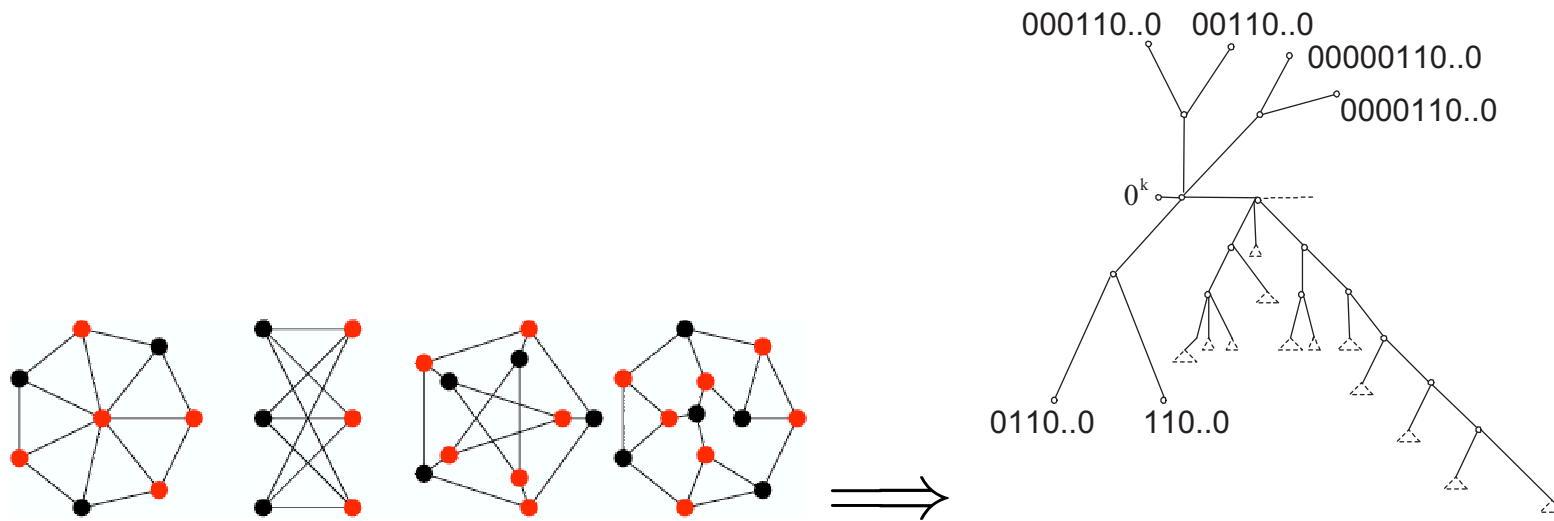
Intractability Proof: The Big Picture

Efficiently translate **vertex cover (VC)** to **ML**.



Intractability Proof: The Big Picture

Efficiently translate **vertex cover (VC)** to **ML**.



“Translation” means

- **Small** cover \implies **Large** likelihood.
- **Large** cover \implies **Small** likelihood.