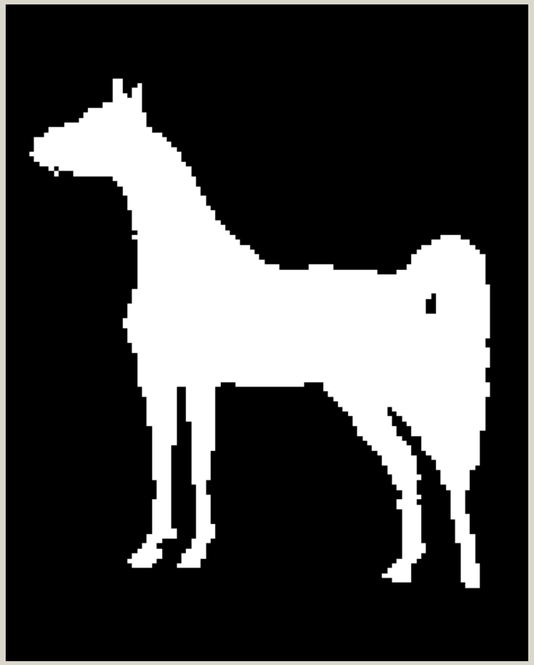
How Hard is Inference for Structured Prediction?

→



Aditya Kanteti

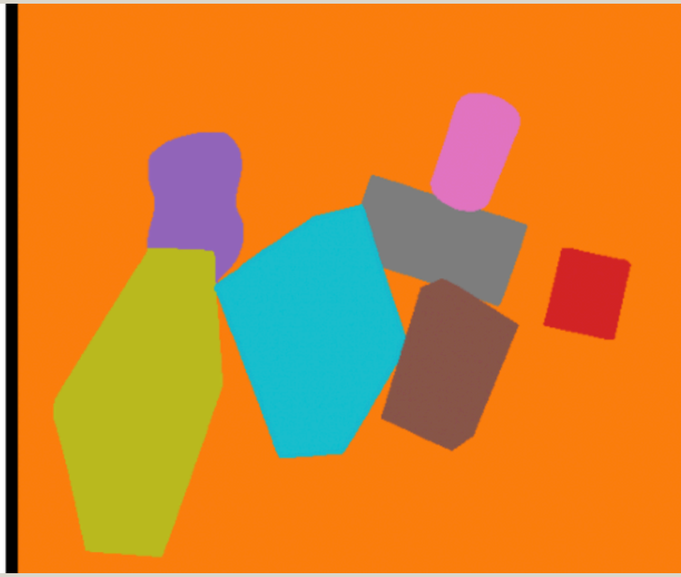
November 17th 2025

Introduction

Authors: Globerson, Roughgarden, Sontag, Yildirim
Venue: ICML 2015
Motivation: Understand why structured prediction inference seems easy in practice despite being NP-Hard

Structured prediction is a problem where you want to predict a large collection of interdependent variables.

The key point is that given realistic data, the focus should not be on worst-case complexity and instead on bringing down the hamming error



The problem: background/foreground prediction

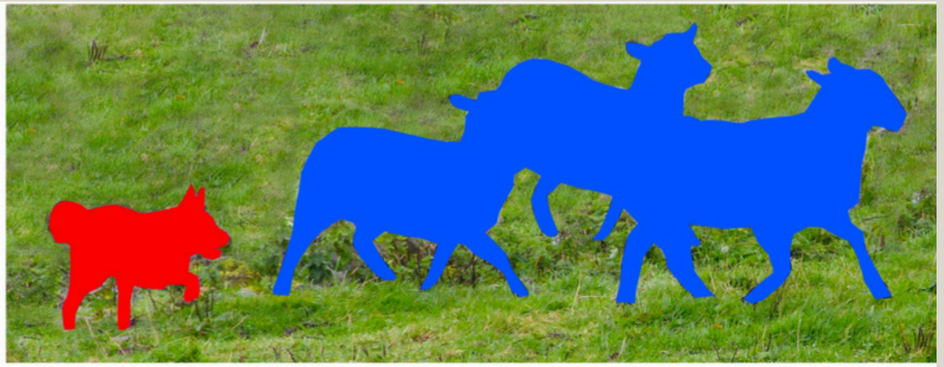
Perfect inference is intractable, yet heuristics work, so how good are they actually when using hamming error?

The goal is to formalize a model of the image segmentation process, analyze the expected Hamming error of an efficient algorithm, and prove its error is as small as theoretically possible

Structured prediction models like CRFs decompose to:

$$s(X, Y) = \sum_{v \in V} \phi_v(X, Y_v) + \sum_{uv \in E} \phi_{uv}(X, Y_u, Y_v).$$

(The exact MAP/marginal inference is NP-Hard)



Hardness vs practicality

Worst case analysis focuses on adversarially constructed images

For MAP or marginal inference on 2-D grids with pariwise interactions → NP-Hard

Real data has regularities like smooth boundaries, local consistency, and non-adversarial noise, which makes inference easier

The paper formalizes this intuition through a probabilistic generative model for observations

$$Y \in \{-1, +1\}^N$$

$$X_v = \begin{cases} Y_v, & \text{with probability } 1-q, \\ -Y_v, & \text{with probability } q. \end{cases}$$

$$X_{uv} = \begin{cases} Y_u Y_v, & \text{with probability } 1-p, \\ -Y_u Y_v, & \text{with probability } p. \end{cases}$$

The Generative Model

This is the main data-generative mechanism

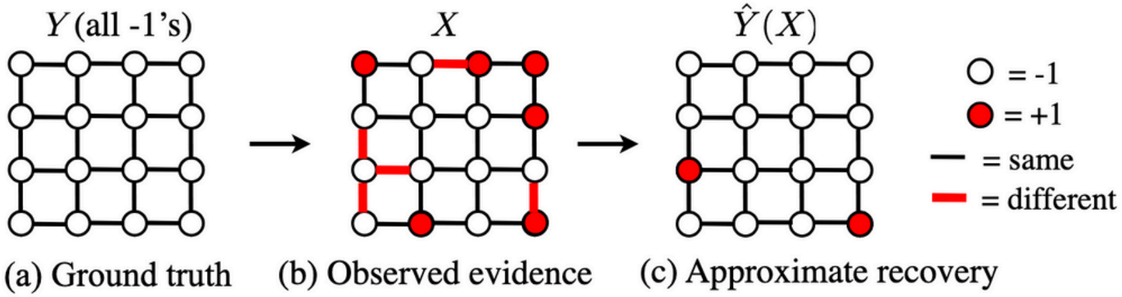
For each node we observe a noisy unary observation, which can flip the sign

$$X_v = \begin{cases} Y_v, & \text{with probability } 1-q, \\ -Y_v, & \text{with probability } q. \end{cases}$$

Edge observations represent whether neighbors agree

$$X_{uv} = \begin{cases} Y_u Y_v, & \text{with probability } 1-p, \\ -Y_u Y_v, & \text{with probability } p. \end{cases}$$

p is typically small, and q can potentially be high. This matches what we expect from real images



We're essentially stating that with small edge noise, and grid structure, inference can become tractable

Related to Beyond worst-case analysis

Hamming Error

Hamming error counts how many labels are mismatched, which is appropriate for pixels.

If an image has 10,000 pixels and we mislabel 17, then the segmentation is nearly perfect, regardless of whether the CRF score is exactly maximized

Hamming Error:

$$e(A) = \max_y \mathbb{E}_{X|Y=y} \left[\frac{1}{2} \|A(X) - y\|_1 \right]$$

We don’t need the exact MAP solution

This is the entire premise of the paper, if exact inference is hard, low error labeling may be easy, and Hamming error is the metric we should use

The two stage algorithm

So what algorithm finds an optimal solution effectively, using what we know about the general data?

Edge observations are reliable ($p \ll 1$)

Node observations are noisy (q can be large)

The algorithm shared is a simple 2 stage decoding algorithm that achieves the optimal possible Hamming error - matching what the intractable marginal MAP would achieve

$$\hat{Y} = \arg\max_{Y \in \{\pm 1\}^N} \sum_{uv \in E} X_{uv}\, Y_u Y_v.$$

The first stage gives the shape of the segmentation (recovering the connected geometry)

The second stage picks the correct sign

$$\text{If } \sum_v X_v \hat{Y}_v < 0, \quad \text{output } -\hat{Y}; \quad \text{else output } \hat{Y}.$$

Why does Stage 1 work?

Ignore all unary/node observations, instead compute a Max-Agreement problem (solve the labeling that agrees with the max number of edges

We reduce the problem to Maximum-Weight Perfect Matching which has a polynomial time solution $O(n^3)$ or $O(n^{2/3})$

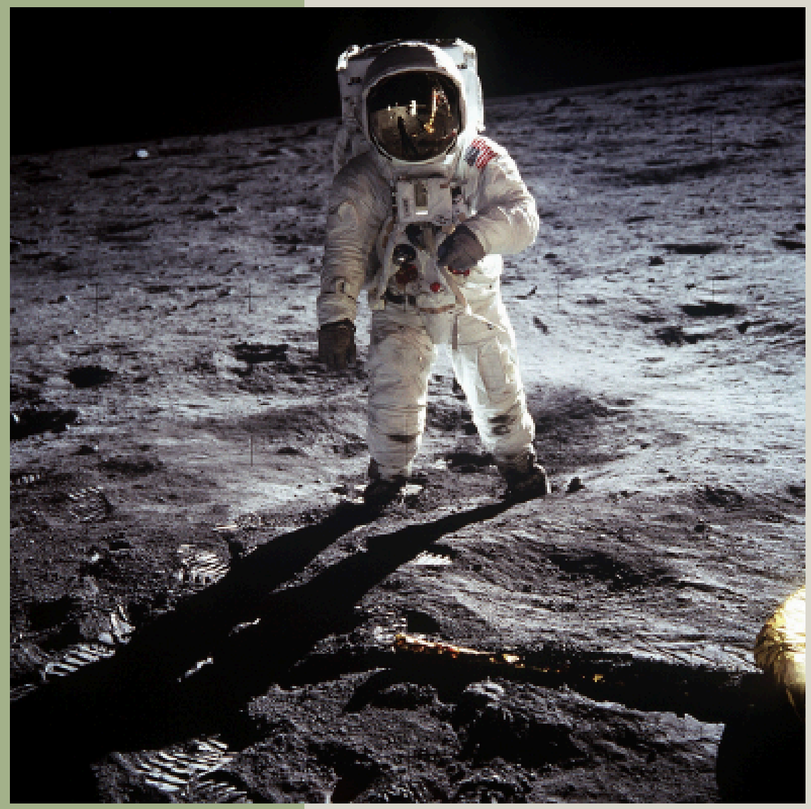
$$\hat{Y} = \arg \max_{Y \in \{\pm 1\}^N} \sum_{uv \in E} X_{uv}\, Y_u Y_v.$$

$$\mathbb{E}[\text{Hamming error after Stage 1}] = O(p^2 N).$$

Why does Stage 1 work? - Continued

Intuitively the edge observations encode local smoothness, and indicate whether two nieghboring pixes should match or differ

In the real world, noise on edges is small, so we’re essentially solving a denoising problem where we trust local pairwise relations



If a pixel was incorrectly labeled, the edge observations would need to be wrong enough to support the mistake.

How does Stage 2 work?

After stage 1, the algorithm has nearly perfect segmentation, but because it's related to pairwise constraints (that only measure relative differences) we don't know the absolute labels.

What this essentially means is we're unsure if the labelings of Y are actually Y or -Y

$$\text{Score}(Y) = \sum_{v \in V} X_v Y_v.$$

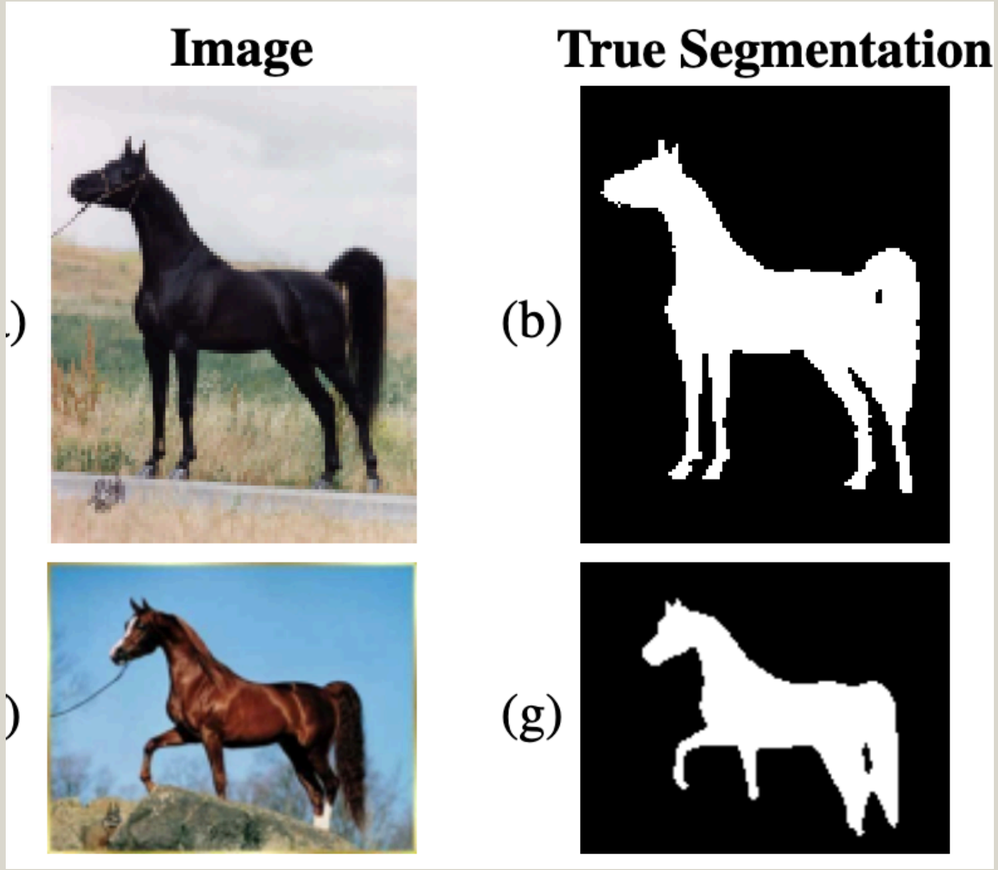
Decision rule:

$$\text{If } \sum_{v} X_v \hat{Y}_v < 0, \quad \text{output } -\hat{Y}; \quad \text{else output } \hat{Y}.$$

How does Stage 2 work? - continued

Our solution is to use unary/node observations to decide between two labelings.

Even if the node noise is large (even close to 0.5) we aggregate evidence and with majority vote end up with the correct answer with high probability



True Optimal vs Current 2 step algorithm

Even if we perform exact marginal inference, the total number of mistakes for an optimal predictor is not zero. It is $O(p^2 N)$. Meaning our algorithm is as good as any.

The reason why is because the optimal solution must deal with the inherent uncertainty introduced by the generative model. This model can include ambiguous data and noise, which limits the accuracy

A pixel becomes impossible to ddetermine correctly when multiple noisy edges around it are wrong at the same time.

probability of single edge wrong → p
probability with 2 edges wrong → p^2
For N pixels → $p^2 N$

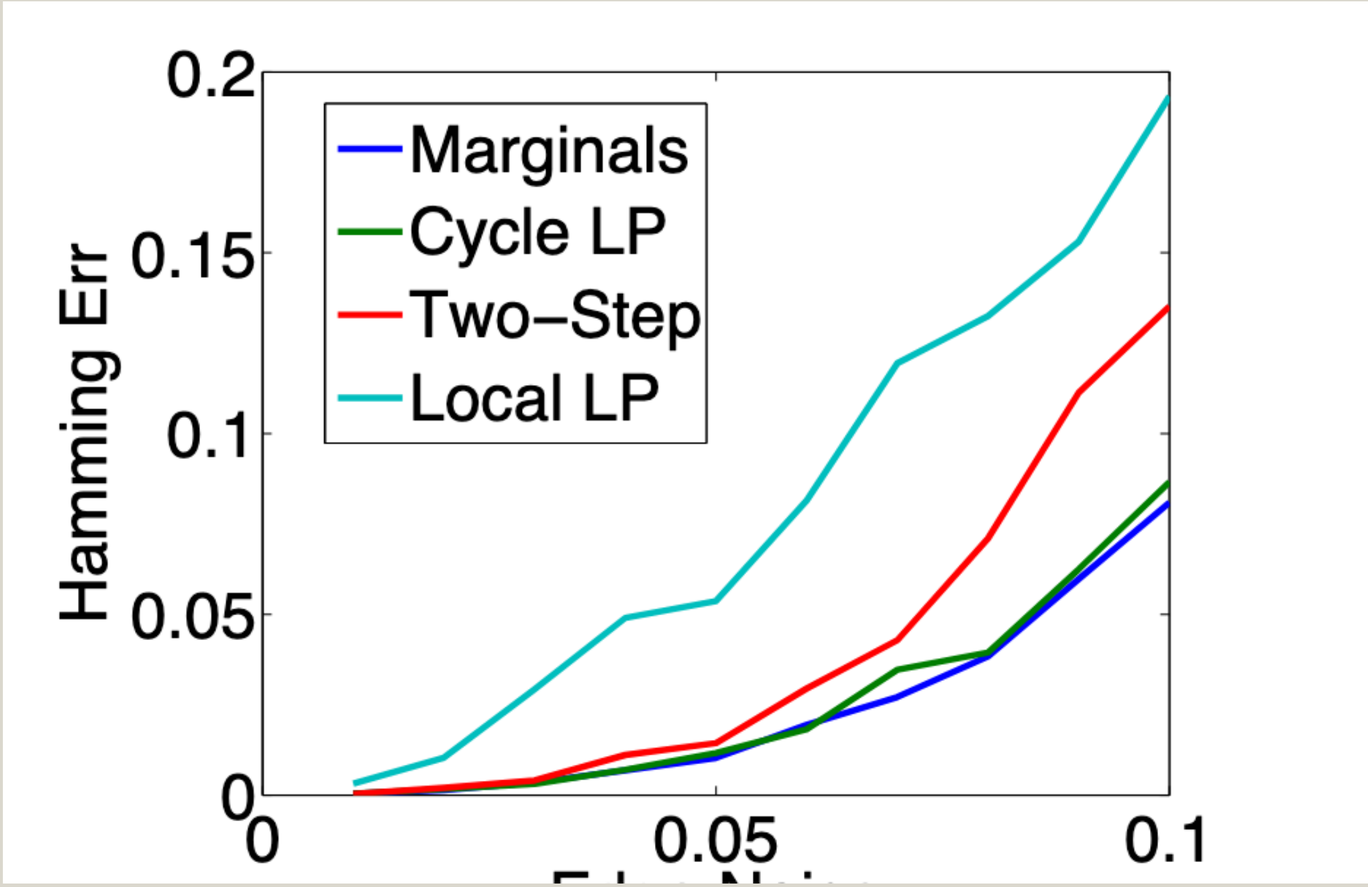
True Optimal vs Current 2 step algorithm - Continued

The authors valdiated this theory with synthetic 20x20 grids and tested them.

p = 0.4

They tested multiple inference strategies:
1) Exact marginals
2) LP relaxations
3) Cycle LP relaxations

When edge noise is low, the 2 step algorithm is virtually identical to the exact marginal inference



Thank you!

Aditya Kanteti