Calibrated Structured Prediction

Volodymyr Kuleshov, Percy Liang

Department of Computer Science, Stanford University

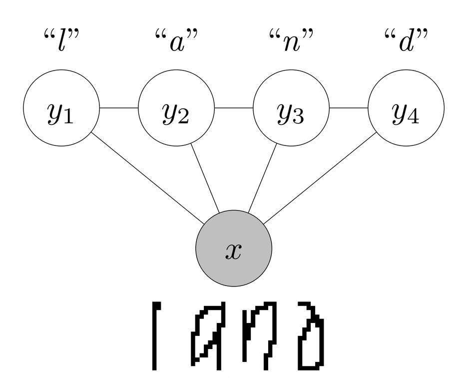
"It ain't what you don't know that gets you into trouble. It's what you know for sure that just ain't so."
– Mark Twain

Motivation

Assessing forecast confidence is often as important as achieving high accuracy, e.g.:

- How certain are we that this patent has cancer?
- Did we correctly understand the user's command?

This work studies *calibrated* confidence estimation for *structured* prediction problems.



(a) Structured prediction model

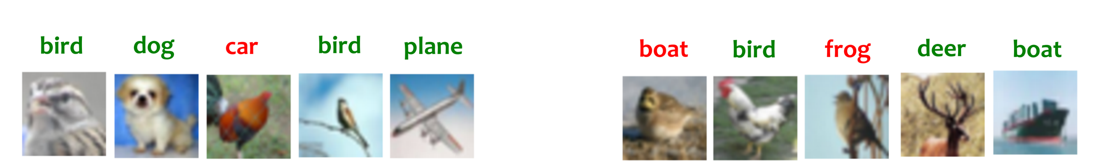
Event	Probability
$[y = \text{"land"}]$	0.8
$[y_1 = \text{"l"}]$	0.8
$[y_2 = \text{"a"}]$	0.9
$[y_3 = \text{"n"}]$	0.9
$[y_4 = \text{"d"}]$	0.8

(b) Forecaster output

Calibration

We assess confidence via *calibrated* probabilities: e.g., if forecaster $h(x)$ detects an object with 70% confidence, we see the object on 70% of these times.

$$\mathbb{P}[y = 1 \mid h(x) = p] = p \qquad \forall p \in [0,1]. \tag{1}$$



80% confidence predictions

60% confidence predictions

How to Ensure Calibration?

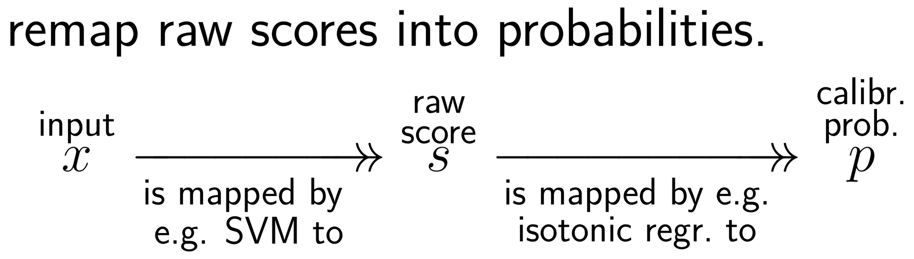
Suppose we have a binary classifier $h : \mathcal{X} \to [0,1]$. Calibration is implicitly optimized by ℓ_2 loss:

$$\mathbb{E}[(y - h(x))^2] \approx \underbrace{\mathbb{E}[(T(x) - h(x))^2]}_{\text{calibration error}} - \underbrace{\text{Var}[T(x)]}_{\text{sharpness}}$$

where $T(x) = \mathbb{E}[y \mid h(x)]$ is the true probability of $y = 1$ given a that x has forecast $h(x)$. Sharpness encourages useful predictions close to 0 or 1.

Recalibration

Popular methods like Platt scaling or isotonic regression remap raw scores into probabilities.



Subtleties in the Structured Setting

Suppose we have a CRF $p_\theta(y|x) : \mathcal{Y} \times \mathcal{X} \to [0,1]$:

- The set $\mathcal{Y}$ of labels y_i may be huge.
- Complexity of inference becomes an issue (e.g. evaluating calibration error may be hard)

Generalizing Calibration

Events of interest. Users specify a set of $\mathcal{I}(x)$ of events $E \subseteq \mathcal{Y}$ whose $\mathbb{P}$ they want to estimate, e.g.:

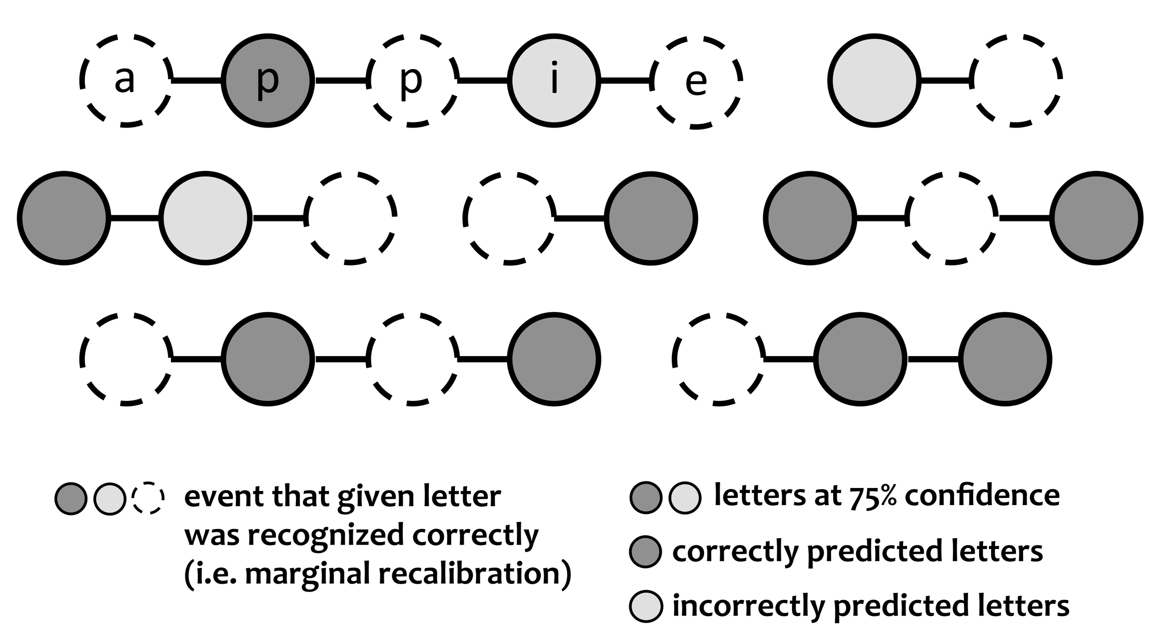
- *MAP calibration:* $\mathcal{I}(x) = \{\text{MAP}(x)\}$.
- *Marginal calibration:* $\mathcal{I}(x) = \{y : y_j = \text{MAP}(x)_j\}$.

The OCR example illustrates the notion of events.

Event Pooling. We say that a forecaster $F : \mathcal{X} \times 2^{\mathcal{Y}} \to [0,1]$ is perfectly calibrated if

$$\mathbb{P}(y \in E \mid F(x, E) = p) = p, \tag{2}$$

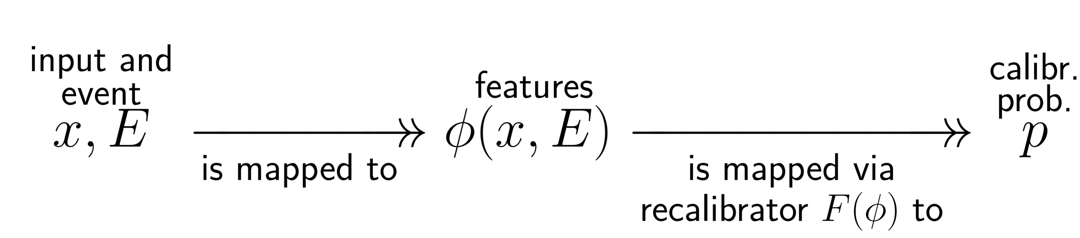
where $\mathbb{P}$ is extended to (x, y, E), and E is drawn uniformly from $\mathcal{I}(x)$, e.g.:



Of the 75% confidence marginals, 75% are correct; note that the first letter in each word is not calibrated.

Recalibration Framework for CRFs

Idea: Reduce to binary calibration of $\mathbb{I}[E \in \mathcal{I}(x)]$ at x based on domain-general features $\phi(x, E)$.



Starting with calibration set $\mathcal{S}$:

- Construct the events dataset $\mathcal{D} = \{(\phi(x, E), \mathbb{I}[y \in E]) : (x, y) \in \mathcal{S}, E \in \mathcal{I}(x)\}$.
- Train the forecaster F (e.g., k-NN) on $\mathcal{D}$.

Experimental Setup

- *Multi-class image classification* on CIFAR-10 using SVM with features learned via k-means.
- *Optical character recognition* via chain CRF on 3-12 letter words.
- *Scene understanding*: predicting superpixel labels with graph CRF on VOC Pascal dataset.

Experiment Highlights

- Domain-independent features are effective for recalibrating structured predictors
- Structured predictors can be recalibrated with little computational overhead; MAP-based features are effective for marginal recalibration.
- In multi-class setting, framework improves over existing 1-vs-all recalibration methods.

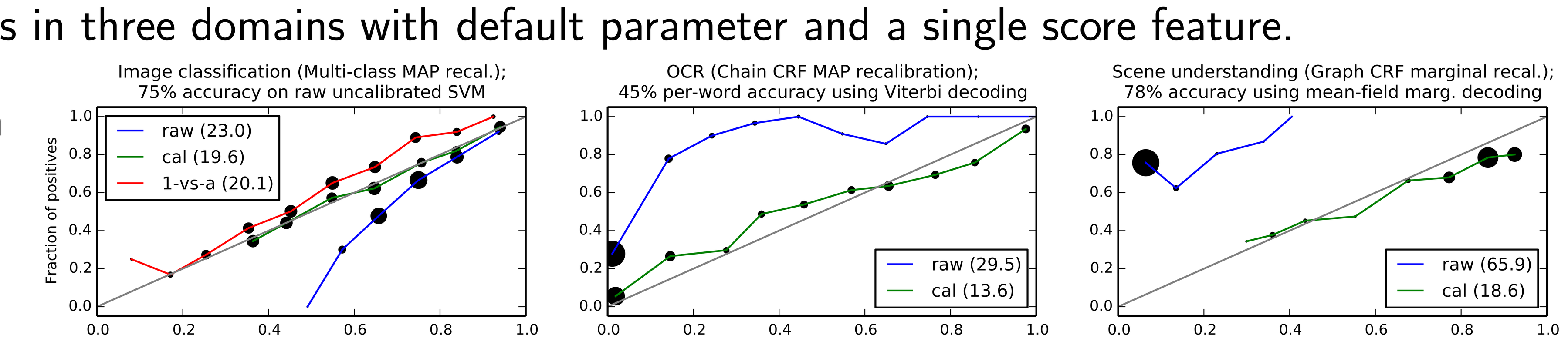
Features

Type	*Features for MAP recalibration on y^{MAP}*	*Features for Marginal recalibration on y_j^{MAP}*
none	ϕ_1^{no} : Regular SVM scores	ϕ_2^{no} : Regular SVM scores
MAP	ϕ_1^{mp} : Number of labels $\lvert y^{MAP}\rvert$	ϕ_4^{mp} : % positions j' labeled y_j^{MAP}
	ϕ_2^{mp} : Is y^{MAP} in user-defined set $\mathcal{G}$?	ϕ_5^{mp} : % neighbors j' labeled y_j^{MAP}
	ϕ_3^{mp} : Scores $p_\theta(y^{MAP}\|x)$	ϕ_6^{mp} : Is y_j^{MAP} in user-defined set $\mathcal{G}$?
		ϕ_7^{mp} : Pseudomarginals $p_\theta(y_j^{MAP}\|y_{-j}^{MAP}, x)$
Marg.	ϕ_1^{mg} : Label scores $p_\theta(y_j^{MAP}\|x)$	ϕ_2^{mg} : Label scores $p_\theta(y_j^{MAP}\|x)$
		ϕ_3^{mg} : Concordance of MAP/marginal decoding

Out-of-the-Box Performance

We obtain calibrated scores in three domains with default parameter and a single score feature.

- In the multi-class domain (left), we do better than the existing 1-vs-all approach.



Feature Analysis

Main observations:

- We can always achieve calibration; features determine sharpness.
- Simple features do almost as well as computationally complex ones.
- Features act synergistically to help each other.
- Recalibration benefits from "global" features to simple graphical models.