Uncertainty: Knowing What You Don't Know

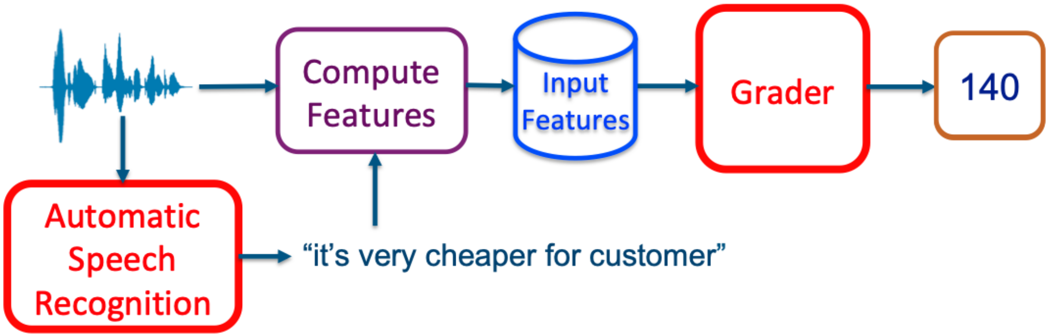
Mark Gales with Anton Ragni, Andrey Malinin
and ALTA Speech Group

24th October 2019

Computer says no: Irish vet fails oral English test needed to stay in Australia

Louise Kennedy, a native English speaker with two degrees, says flawed technology is to blame

Spoken Language Assessment Pipeline



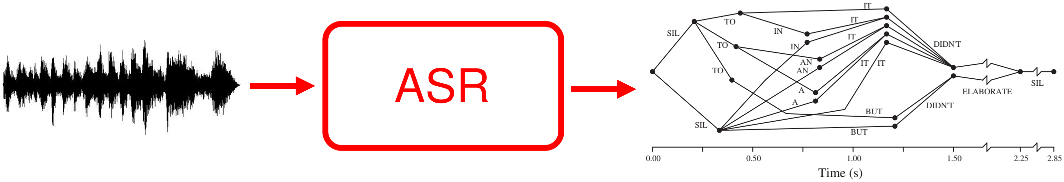
ASR Confidence Scores

ASR Confidence Scores [?]

- Useful to know whether ASR output is correct
  - confidence scores supply this information
  - three forms of error: substitutions, deletions and insertions

`manual`	AND	THESE	ARE		THE	FIMBLES
`asr`		THIS	ARE	TO	THE	FIMBLES
`error`	del	sub	—	ins	—	—
`conf`	—	0.4	0.8	0.3	0.9	0.9

Baseline Confidence Scores

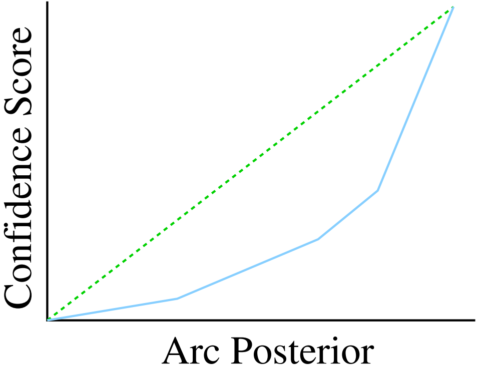


- Baseline confidence scores based on arc posteriors

$$\mathrm{p}(\boldsymbol{q}_{1:T},\boldsymbol{x}_{1:T}) = \mathrm{p_a}(\boldsymbol{x}_{1:T}|\boldsymbol{q}_{1:T})^{\frac{1}{\gamma}}\mathrm{P_l}(w_{1:L});\quad \mathrm{P}(a|\mathcal{L}) = \frac{\sum_{\boldsymbol{q}_{1:T}\in\mathcal{Q}_a}\mathrm{p}(\boldsymbol{q}_{1:T},\boldsymbol{x}_{1:T})}{p(\boldsymbol{x}_{1:T})}$$

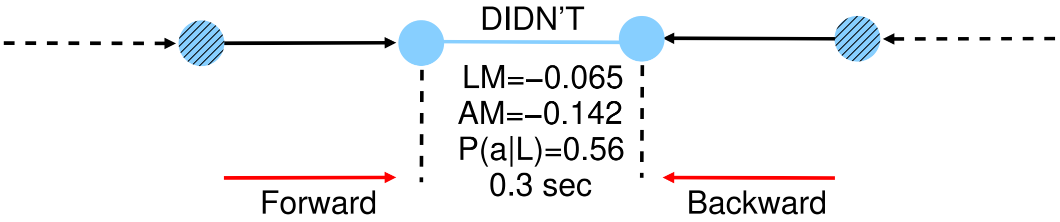
  - $\boldsymbol{q}_{1:T}$ T-length state sequence for word sequence $\boldsymbol{\omega}$
  - $\mathcal{Q}_a$ set of state sequences that pass through arc a
  - γ is usually the LM scale factor
  - does not alter 1-best (compared to scaling LM)

Confidence Score Calibration [?]



- Confidence scores often over-estimated
  - very simple normalisation approach

Neural Network Based Confidence Scores



- Use more general sequence model
  - for 1-best $w_{1:L} = w_1, \ldots, w_L$
  - use information associated with each arc $\mathrm{a}_{1:L}$

$$P(w_i|\mathrm{a}_{1:L}) = \mathcal{F}(\mathrm{a}_i, \overrightarrow{\mathrm{a}}_{1:i-1}, \overleftarrow{\mathrm{a}}_{i+1:L})$$

RNN-Based Confidence Scores [?, ?]

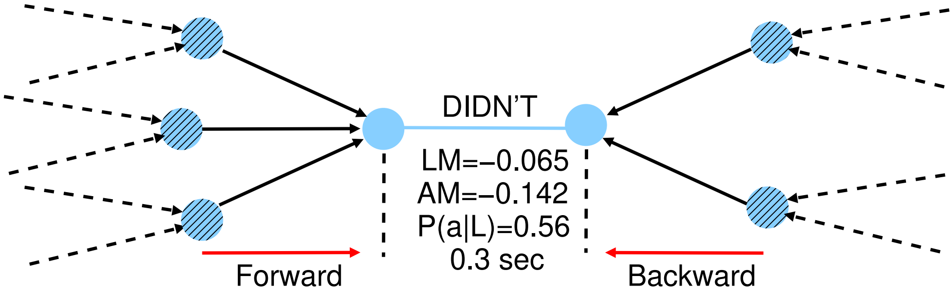
- Simple approach use recurrent neural networks

$$\overrightarrow{\boldsymbol{h}}_i = \mathcal{F}(\overrightarrow{\boldsymbol{h}}_{i-1}, \mathrm{a}_1); \quad \overleftarrow{\boldsymbol{h}}_i = \mathcal{F}(\overleftarrow{\boldsymbol{h}}_{i+1}, \mathrm{a}_1);$$
$$P(w_i|\mathrm{a}_{1:L}) = \mathcal{F}(\overrightarrow{\boldsymbol{h}}_i, \overleftarrow{\boldsymbol{h}}_i)$$

- Evaluation: Georgian (!) Conversation Telephone Speech
  - RNN-based on: posteriors, word ID and durations

System	NCE	AUC
Arc posteriors	-0.1978	0.9081
+ calibration	0.2755	0.9081
+ RNN	0.2911	0.9121

Lattice Neural Network Based Confidence Scores [?, ?]

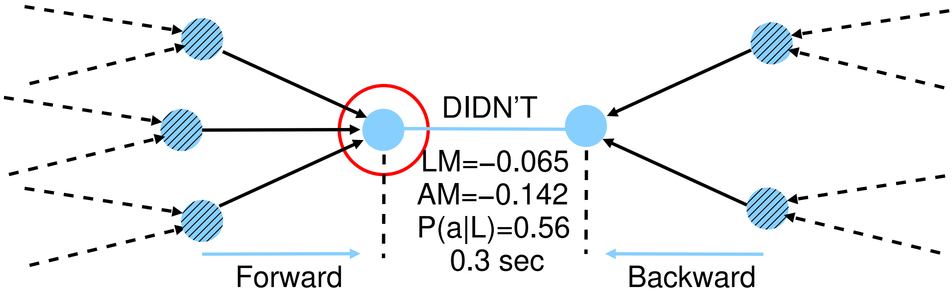


- Make use of complete lattice $\mathcal{L}$

$$P(w_i|\mathcal{L}) = \mathcal{F}(\mathrm{a}_i, \overrightarrow{\mathcal{Q}}_{\mathrm{a}_i}, \overleftarrow{\mathcal{Q}}_{\mathrm{a}_i})$$

  - $\overrightarrow{\mathcal{Q}}_{\mathrm{a}_i}$ set of arcs in forward direction to a_i
  - $\overleftarrow{\mathcal{Q}}_{\mathrm{a}_i}$ set of arcs in backward direction to a_i

Lattice Neural Network Based Confidence Scores

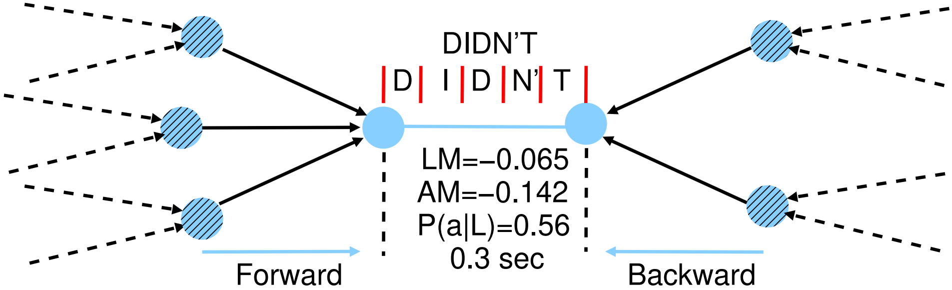


- Use attention to merge arcs

$$\overrightarrow{\boldsymbol{h}}_i = \texttt{attention}\left(\{\overrightarrow{\boldsymbol{h}}_j\}_{j\in\overrightarrow{\mathcal{N}}_{\mathrm{a}_i}}, \mathrm{a}_i\right); \quad \overleftarrow{\boldsymbol{h}}_i = \texttt{attention}\left(\{\overleftarrow{\boldsymbol{h}}_j\}_{j\in\overleftarrow{\mathcal{N}}_{\mathrm{a}_i}}, \mathrm{a}_i\right);$$

$$P(w_i|\mathrm{a}_{1:L}) = \mathcal{F}(\overrightarrow{\boldsymbol{h}}_i, \overleftarrow{\boldsymbol{h}}_i)$$

Grapheme Features



- Add grapheme ID and duration information

$$\boldsymbol{g}_i = \texttt{self}-\texttt{attention}(\{\boldsymbol{g}_i^{(1)}, \ldots, \boldsymbol{g}_i^{(N)}\}); \quad \boldsymbol{g}_i^{(j)} = \begin{bmatrix} \texttt{id}_i^{(j)} \\ d_i^{(j)} \end{bmatrix}$$

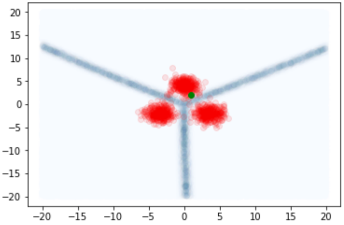
LatticeRNN Based Confidence Scores

- Evaluation on a CTS task
  - RNN-based on: posteriors, word ID and durations
  - latticeRNN acts on confusion networks

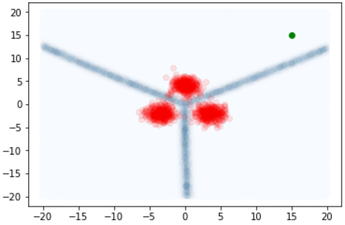
System	NCE	AUC
RNN	0.2911	0.9121
lattice-RNN	0.2934	0.9178
+ grapheme	0.3004	0.9231

Prediction Uncertainty

Sources of Uncertainty

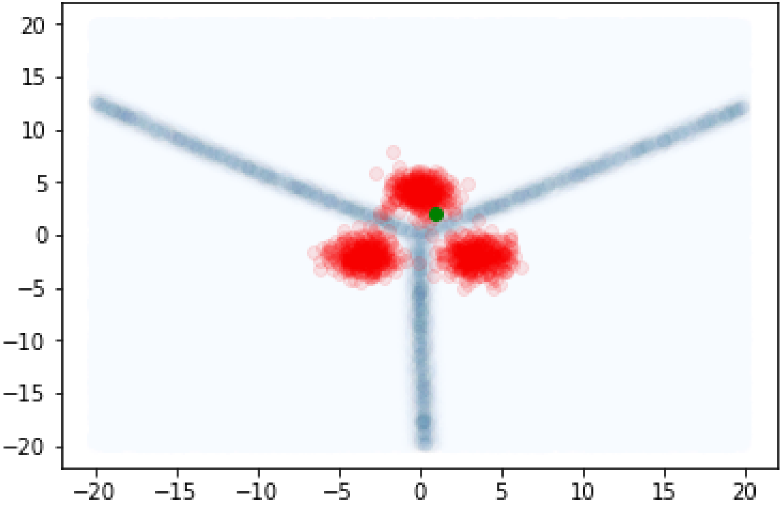


(a) Data Uncertainty



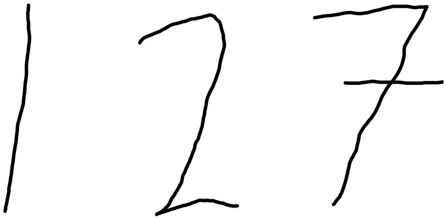
(b) Knowledge Uncertainty

Data (Aleatoric) Uncertainty

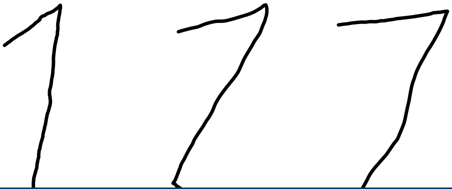


Data Uncertainty

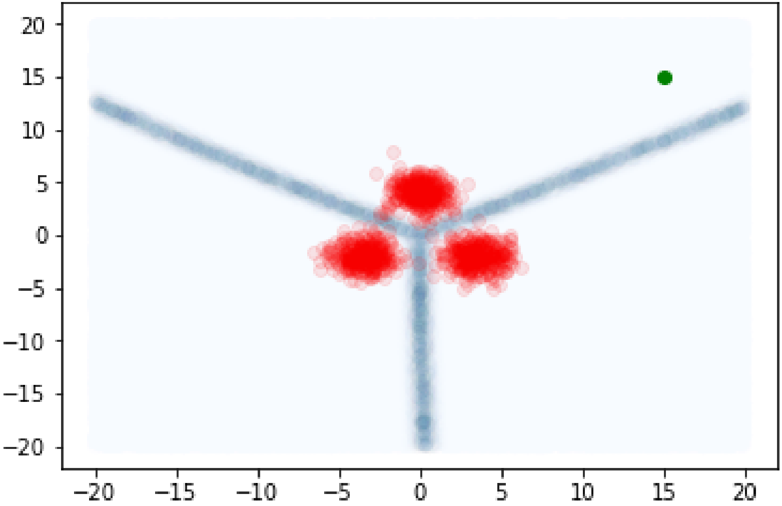
- Distinct Classes



- Overlapping Classes

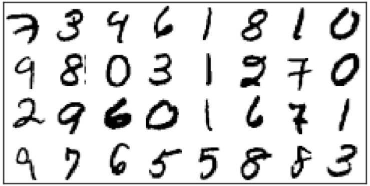


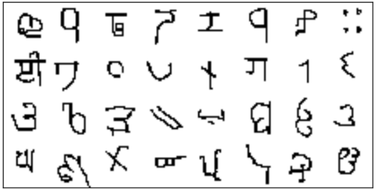
Knowledge (Distributional/Epistemic) Uncertainty



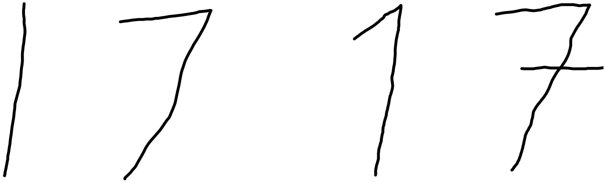
Knowledge Uncertainty

- Unseen classes





- Unseen variations of seen classes

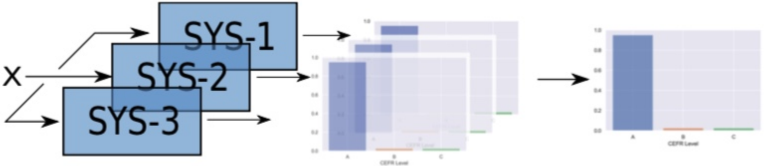


Ensemble Approaches

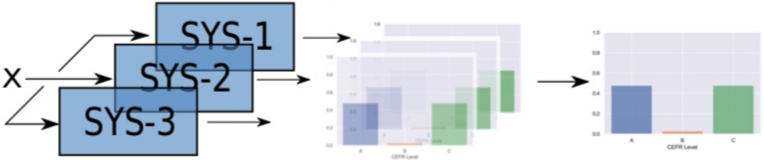
- Given training data $\mathcal{D}$

$$
\begin{aligned}
\mathrm{P}(y|\boldsymbol{x}^*, \boldsymbol{\theta}) &= \int \mathrm{P}(y|\boldsymbol{x}^*, \boldsymbol{\theta})\mathrm{p}(\boldsymbol{\theta}|\mathcal{D})d\boldsymbol{\theta} \\
&\approx \frac{1}{M}\sum_{i=1}^{M} \mathrm{P}(y|\boldsymbol{x}^*, \boldsymbol{\theta}^{(i)}); \quad \boldsymbol{\theta}^{(i)} \sim \mathrm{p}(\boldsymbol{\theta}|\mathcal{D})
\end{aligned}
$$

Ensemble Approaches



(a) Confident



(b) Uncertain on decision boundary

Entropy for Uncertainty

- Simple reminder of Entropy

$$\mathcal{H}[\mathrm{P}(y|\boldsymbol{x}^*,\boldsymbol{\theta})] = -\sum_{c=1}^{K} \mathrm{P}(y=\omega_c|\boldsymbol{x}^*,\boldsymbol{\theta}) \log\left(\mathrm{P}(y=\omega_c|\boldsymbol{x}^*,\boldsymbol{\theta})\right)$$

- General attributes
  - high entropy: "flat" distribution, low confidence
  - low entropy: "peaky" distribution, high confidence

- Doesn't give information about source of uncertainty!

Ensemble Consistency [?, ?]

- Mutual Information

$$\underbrace{\mathcal{I}[y,\boldsymbol{\theta}|\boldsymbol{x}^\star,\mathcal{D}]}_{\texttt{Knowledge Uncertainty}} = \underbrace{\mathcal{H}[\mathbb{E}_{\mathrm{p}(\boldsymbol{\theta}|\mathcal{D})}[\mathrm{p}(y|\boldsymbol{x}^\star,\boldsymbol{\theta})]]}_{\texttt{Total Uncertainty}} - \underbrace{\mathbb{E}_{\mathrm{p}(\boldsymbol{\theta}|\mathcal{D})}[\mathcal{H}[\mathrm{p}(y|\boldsymbol{x}^\star,\boldsymbol{\theta})]]}_{\texttt{Expected Data Uncertainty}}$$

- Total Variance

$$\underbrace{\mathbb{V}[y,\boldsymbol{\theta}|\boldsymbol{x}^\star,\mathcal{D}]}_{\texttt{Total Variance}} = \underbrace{\mathbb{V}_{\mathrm{p}(\boldsymbol{\theta}|\mathcal{D})}\Big[\mathbb{E}_{\mathrm{p}(y|\boldsymbol{x}^\star,\boldsymbol{\theta})}[y]\Big]}_{\texttt{Mean Variance}} + \underbrace{\mathbb{E}_{\mathrm{p}(\boldsymbol{\theta}|\mathcal{D})}\Big[\mathbb{V}_{\mathrm{p}(y|\boldsymbol{x}^\star,\boldsymbol{\theta})}[y]\Big]}_{\texttt{Expected Data Variance}}$$

- Expected (Pairwise) KL-Divergence

$$\mathrm{KL}[y,\boldsymbol{\theta}|\boldsymbol{x}^\star,\mathcal{D}] = \mathbb{E}_{\mathrm{p}(\boldsymbol{\theta}|\mathcal{D}),\mathrm{p}(\tilde{\boldsymbol{\theta}}|\mathcal{D})}\Big[\mathrm{KL}[\mathrm{p}(y|\boldsymbol{x}^\star,\boldsymbol{\theta})||\mathrm{p}(y|\boldsymbol{x}^\star,\tilde{\boldsymbol{\theta}})]\Big]$$

Ensemble "Generation" [?, ?, ?]

- Deep learning approaches often use 10,000,000+ parameters

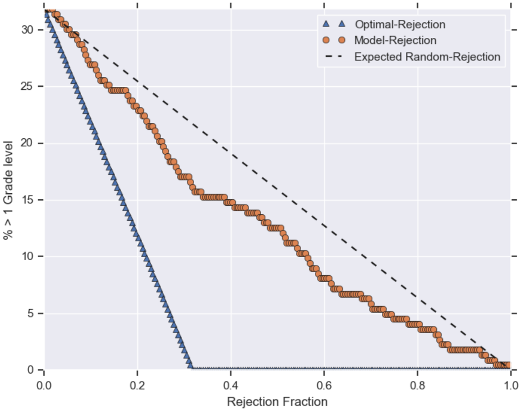
Modelling $p(\boldsymbol{\theta}|\mathcal{D})$ challenging

  - use variational approximations
  - Monte-Carlo methods
  - non-Bayesian approaches e.g. random network initialisation

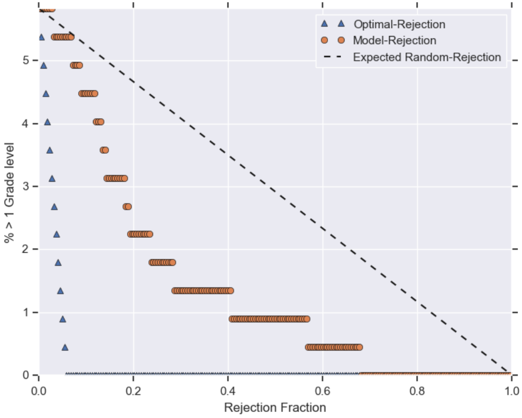
Computer says no: Irish vet fails oral English test needed to stay in Australia

Louise Kennedy, a native English speaker with two degrees, says flawed technology is to blame

Grader Uncertainty: Ensemble-Based



Within 0.5 CEFR-Level



Within 1.0 CEFR-Level

Prior Networks

Ensemble Modelling [?, ?]

- Ensembles compute/memory intensive (scales linearly)
  - challenging to guarantee performance for outliers

Ensemble Modelling [?, ?]

- Ensembles compute/memory intensive (scales linearly)
  - challenging to guarantee performance for outliers

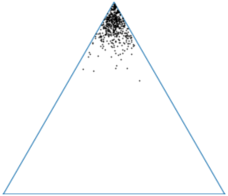
- Possible to compress ensemble to a single model:
  - Ensemble Distillation: standard compression approach

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \left\{ \mathrm{KL}\left(\frac{1}{M}\sum_{i=1}^{M} \mathrm{P}(y|\boldsymbol{x}^*, \boldsymbol{\theta}^{(i)}) || \mathrm{P}(y|\boldsymbol{x}^*, \boldsymbol{\theta}) \right) \right\}$$

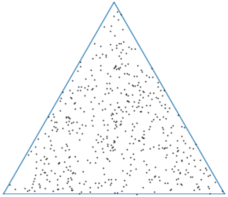
    - models average distribution - loses diversity of ensemble
  - Ensemble Distribution Distillation: model ensemble diversity
    - maintains diversity of the ensemble

Distributions on a Simplex

- Ensemble $\{\mathrm{P}(y|\boldsymbol{x}^*, \boldsymbol{\theta}^{(i)})\}_{i=1}^{M}$ can be visualised on a simplex



(a) In domain $\boldsymbol{x}^*$

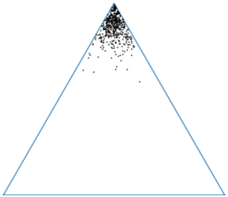


(b) Out-of-domain $\boldsymbol{x}^*$

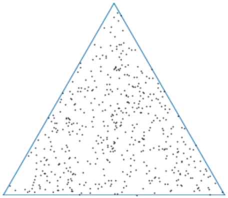
- ensemble samples from a distribution over distributions

Distributions on a Simplex

- Ensemble $\{\mathrm{P}(y|\boldsymbol{x}^*, \boldsymbol{\theta}^{(i)})\}_{i=1}^{M}$ can be visualised on a simplex



(a) In domain $\boldsymbol{x}^*$



(b) Out-of-domain $\boldsymbol{x}^*$

  - ensemble samples from a distribution over distributions
- Only need to model desired distribution
  - should allow explicit control over diversity

Prior Networks [?]

- A Prior Network predicts parameters of Dirichlet Distribution

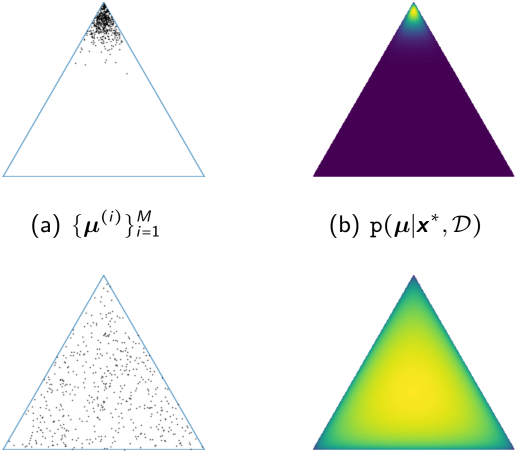
$$\mathtt{p}(\boldsymbol{\mu}|\boldsymbol{x}^*;\hat{\boldsymbol{\theta}}) = \mathtt{Dir}(\boldsymbol{\mu}|\boldsymbol{\alpha}), \quad \boldsymbol{\alpha} = \boldsymbol{f}(\boldsymbol{x}^*;\hat{\boldsymbol{\theta}})$$

where

$$\boldsymbol{\mu} = \begin{bmatrix} \mathtt{P}(y = \omega_1|\boldsymbol{x}^*) \\ \mathtt{P}(y = \omega_2|\boldsymbol{x}^*) \\ \vdots \\ \mathtt{P}(y = \omega_K|\boldsymbol{x}^*) \end{bmatrix}$$

- Dirichlet Distribution → Distribution over simplex
  - Conjugate prior to categorical distribution
  - Convenient properties → analytically tractable

Distribution over Distributions



(a) $\{\boldsymbol{\mu}^{(i)}\}_{i=1}^{M}$ (b) $\mathrm{p}(\boldsymbol{\mu}|\boldsymbol{x}^*, \mathcal{D})$

(c) $\{\boldsymbol{\mu}^{(i)}\}_{i=1}^{M}$ (d) $\mathrm{p}(\boldsymbol{\mu}|\boldsymbol{x}^*, \mathcal{D})$

Prior Network Construction

$$\mathcal{L}(\boldsymbol{\theta},\mathcal{D}) = \underbrace{\mathcal{L}_{in}(\boldsymbol{\theta},\mathcal{D}_{\texttt{trn}})}_{In\ Domain\ Loss} + \gamma\cdot\underbrace{\mathcal{L}_{out}(\boldsymbol{\theta},\mathcal{D}_{\texttt{out}})}_{OOD\ Loss}$$

(a) In-Domain Target (b) OOD Target

- Explicitly train the form of the Dirichlet distributions
  - but requires selection/generation of out-of-distribution data

Target Dirichlet Parameters [?]

- Train network to predict appropriate distribution:
  - map $y^{(i)} \to \boldsymbol{\beta}^{(i)}$: should yield correct class - minimise

$$\mathcal{L}(\boldsymbol{\theta}; \mathcal{D}) = \sum_{i=1}^{N} \mathrm{KL}\left(\mathrm{p}(\boldsymbol{\mu}|\boldsymbol{\beta}^{(i)})||\mathrm{p}(\boldsymbol{\mu}|\boldsymbol{x}^{(i)}; \boldsymbol{\theta})\right)$$

- Consider setting $\boldsymbol{\beta}^{(i)}$ as follows $\to$

$$\beta_k^{(i)} = \begin{cases} \beta + 1 & \mathit{if} \;\; y^{(i)} = \omega_k \\ 1 & \mathit{if} \;\; y^{(i)} \neq \omega_k \end{cases}$$

  - if β is large $\to$: high confidence
  - if β is low $\to$: low confidence
  - If β is zero $\to$: flat (uniform) distribution

- Reverse-KL yields better results (see paper for reasons)

Out-of-Distribution Detection: Image Tasks

- Use CIFAR-100 for out-of-distribution (OOD) training data
  - evaluate performance in detecting OOD test samples
  - metric AUC (average 10 randomly initialised models $\pm 2\sigma$)

Model	CIFAR-10		
	SVHN	LSUN	TinyImageNet
Ensemble	89.5 $\pm$ NA	93.2 $\pm$ NA	90.3 $\pm$ NA
Prior Network	**98.2** $\pm$1.1	**95.7** $\pm$0.9	**95.7** $\pm$0.7

Conclusions

Uncertainty: Knowing What You Don't Know

- Uncertainty important for deploying machine learning
  - systems tend to be overly confident

Uncertainty: Knowing What You Don't Know

- Uncertainty important for deploying machine learning
  - systems tend to be overly confident
- Knowing the cause of uncertainty useful
  - allows different actions to be taken to address uncertainty
  - applications: active learning, uncertainty for RL, ...

Uncertainty: Knowing What You Don't Know

- Uncertainty important for deploying machine learning
  - systems tend to be overly confident
- Knowing the cause of uncertainty useful
  - allows different actions to be taken to address uncertainty
  - applications: active learning, uncertainty for RL, ...
- It's hard!
  - humans aren't too good at it either