Uncertainty Estimation in Supervised Learning

Andrey Malinin

25 August 2019

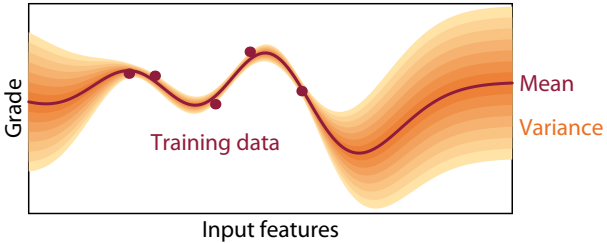
Joint work with...

(a) Prof. Mark Gales

(b) Bruno Mlodozeniec

Scenario

- Given a deployed model and a test input $\boldsymbol{x}^*$ we wish to:
  - Obtain a prediction
  - Obtain measure of model's uncertainty in prediction
  - Take action based estimate of uncertainty
- Example for Regression: Gaussian Process



Application of Uncertainty Estimation

- Threshold-based outlier detection $\rightarrow$
  - Misclassification Detection
  - Out-of-distribution input Detection
  - Adversarial Attack Detection
- Active Learning
- Reinforcement Learning uncertainty-driven exploration
- Other...

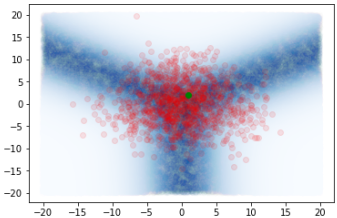
Overview of the Talk

1. Sources of Uncertainty in Predictions
2. Ensemble Approaches
3. Prior Networks
4. Assessment of Uncertainty Quality
5. Ensemble Distribution Distillation

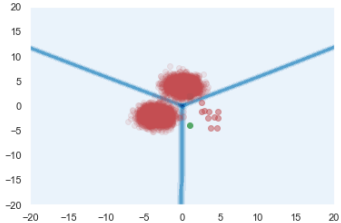
Overview of the Talk

1. **Sources of Uncertainty in Predictions**
2. Ensemble Approaches
3. Prior Networks
4. Assessment of Uncertainty Quality
5. Ensemble Distribution Distillation

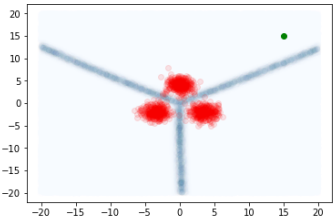
Sources of Uncertainty



(a) Data Uncertainty



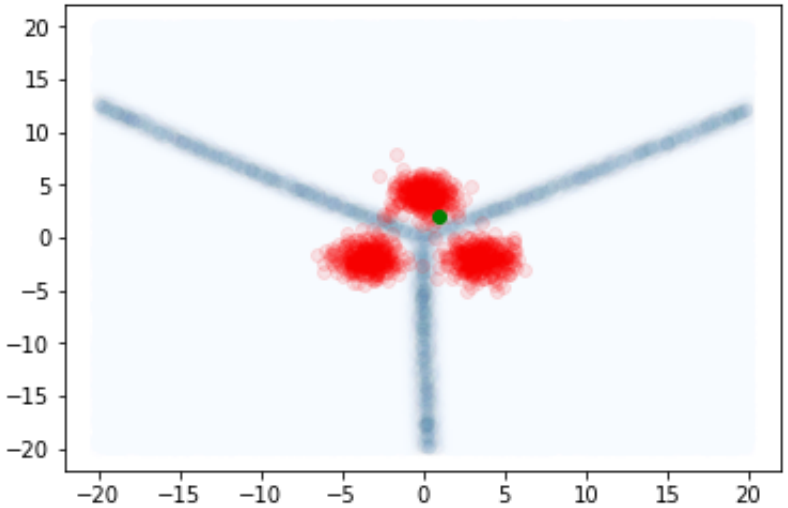
(b) Data Sparsity



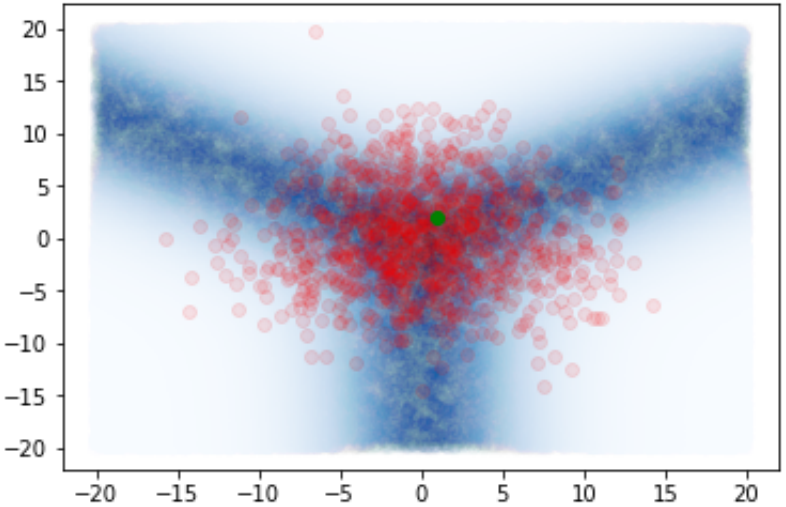
(c) Knowledge Uncertainty

- Epistemic (model) uncertainty often refers to both:
  - Data Sparsity and Knowledge Uncertainty
- **This talk refers only to knowledge uncertainty as defined above!**

Data (Aleatoric) Uncertainty

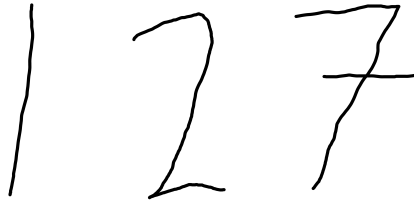


Data Uncertainty



Data Uncertainty

- Distinct Classes



- Overlapping Classes

Data Uncertainty

- Data Uncertainty $\rightarrow$ **Known-Unknown**
- Uncertainty due to properties of data
  - Class overlap (complexity of decision boundaries)
  - Homoscedastic and Heteroscedastic noise

Data Uncertainty

- Data Uncertainty is the *entropy* of the *true data distribution* $\rightarrow$

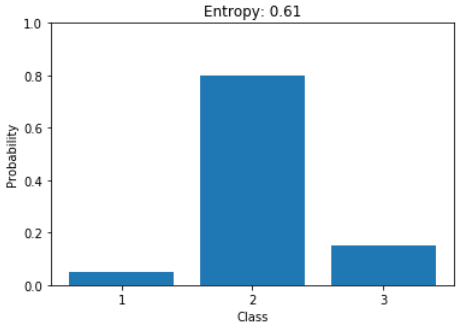
$$\mathcal{H}[\mathrm{P}_{\mathtt{tr}}(y|\boldsymbol{x}^*)] = -\sum_{c=1}^{K} \mathrm{P}_{\mathtt{tr}}(y=\omega_c|\boldsymbol{x}^*) \ln \mathrm{P}_{\mathtt{tr}}(y=\omega_c|\boldsymbol{x}^*)$$

- Captured by the entropy of a model's posterior over classes $\rightarrow$

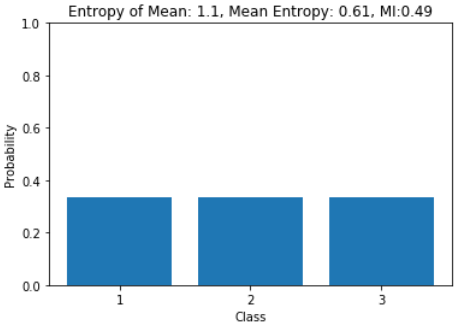
$$\mathcal{H}[\mathrm{P}(y|\boldsymbol{x}^*, \hat{\boldsymbol{\theta}})] = -\sum_{c=1}^{K} \mathrm{P}_{\mathtt{tr}}(y=\omega_c|\boldsymbol{x}^*, \hat{\boldsymbol{\theta}}) \ln \mathrm{P}_{\mathtt{tr}}(y=\omega_c|\boldsymbol{x}^*, \hat{\boldsymbol{\theta}})$$

- Data Uncertainty is captured as a consequence of Maximum Likelihood Estimation

Reminder - Entropy

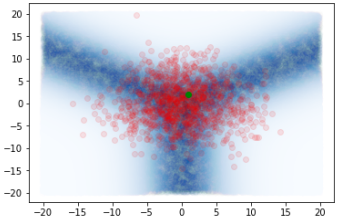


(a) Low Entropy

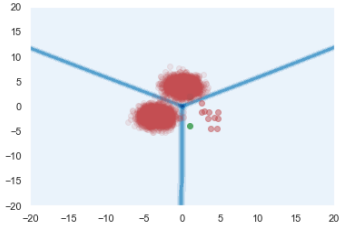


(b) High Entropy

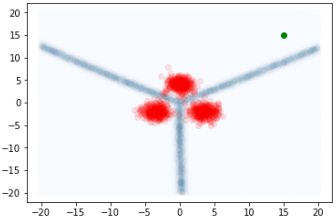
Sources of Uncertainty



(a) Data Uncertainty

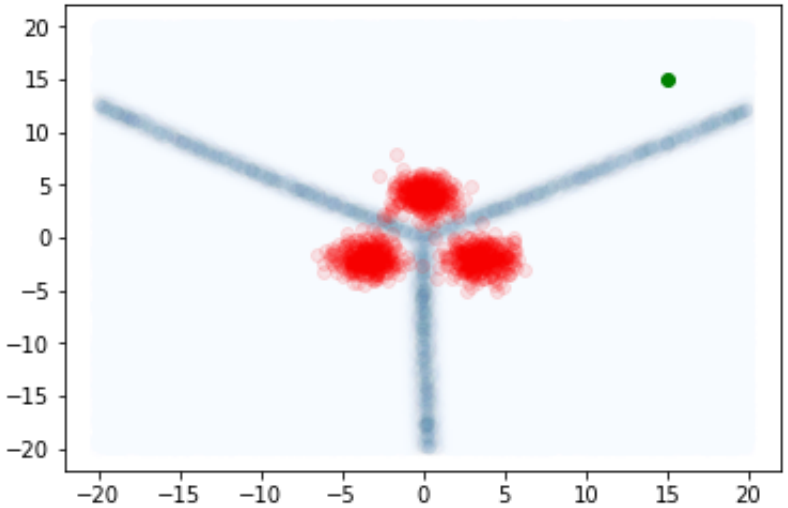


(b) Data Sparsity



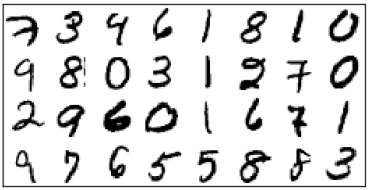
(c) Knowledge Uncertainty

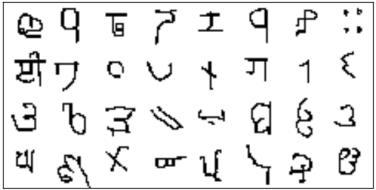
Knowledge Uncertainty



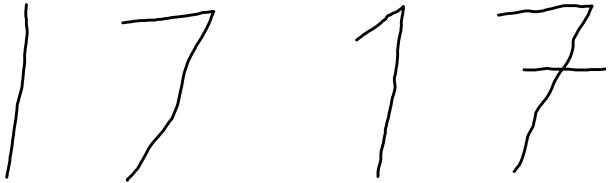
Knowledge Uncertainty - Out-of-Distribution

- Unseen classes





- Unseen variations of seen classes



Sources of Uncertainty

- Data Uncertainty → **Known-Unknown**
  - Class overlap (complexity of decision boundaries)
  - Homoscedastic and Heteroscedastic noise
- Knowledge Uncertainty → **Unknown-Unknown**
  - Test input in out-of-distribution region far from training data
- Appropriate **action** depends on **source** of uncertainty
  - Separating sources of uncertainty requires **Bayesian approaches**

Overview

1. Sources of Uncertainty in Predictions
2. **Ensemble Approaches**
3. Prior Networks
4. Assessment of Uncertainty Quality
5. Ensemble Distribution Distillation

Ensemble Approaches

- Uncertainty in $\boldsymbol{\theta}$ captured by model posterior $\mathrm{p}(\boldsymbol{\theta}|\mathcal{D}) \rightarrow$

$$\mathrm{p}(\boldsymbol{\theta}|\mathcal{D}) = \frac{\mathrm{p}(\mathcal{D}|\boldsymbol{\theta})\mathrm{p}(\boldsymbol{\theta})}{\mathrm{p}(\mathcal{D})}$$

- Bayesian inference of $\mathrm{P}(y|\boldsymbol{x}^*, \boldsymbol{\theta}) \rightarrow$

$$\mathrm{P}(y|\boldsymbol{x}^*, \mathcal{D}) = \mathbb{E}_{\mathrm{p}(\boldsymbol{\theta}|\mathcal{D})}\left[\mathrm{P}(y|\boldsymbol{x}^*, \boldsymbol{\theta})\right]$$

- Can consider an ensemble of models $\rightarrow$

$$\{\mathrm{P}(y|\boldsymbol{x}^*, \boldsymbol{\theta}^{(m)})\}_{m=1}^{M}, \;\; \boldsymbol{\theta}^{(m)} \sim \mathrm{p}(\boldsymbol{\theta}|\mathcal{D})$$

- Choose desired behaviour of ensemble via prior $\mathrm{p}(\boldsymbol{\theta})$

Total Uncertainty

- Consider the entropy of the predictive posterior $\mathrm{P}(y|\boldsymbol{x}^*, \mathcal{D})$

$$\begin{aligned}\mathcal{H}\big[\mathrm{P}(y|\boldsymbol{x}^*, \mathcal{D})\big] &= \mathcal{H}\big[\mathbb{E}_{p(\boldsymbol{\theta}|\mathcal{D})}[\mathrm{P}(y|\boldsymbol{x}^*, \boldsymbol{\theta})]\big] \\ &\approx \mathcal{H}\Big[\frac{1}{M}\sum_{m=1}^{M}\mathrm{P}(y|\boldsymbol{x}^*, \boldsymbol{\theta}^{(m)})\Big],\ \boldsymbol{\theta}^{(m)} \sim \mathrm{p}(\boldsymbol{\theta}|\mathcal{D})\end{aligned}$$

- Measure of Total Uncertainty
  - Combination of Data uncertainty and **Knowledge uncertainty**

Expected Data Uncertainty

- Lets consider an ensemble of models $\{\mathrm{P}(y|\boldsymbol{x}^*, \boldsymbol{\theta}^{(m)})\}_{m=1}^{M},\ \boldsymbol{\theta}^{(m)} \sim \mathrm{p}(\boldsymbol{\theta}|\mathcal{D})$
  - Each model $\mathrm{P}(y|\boldsymbol{x}^*, \boldsymbol{\theta}^{(m)})$ captures an different estimate of data uncertainty.
- Bayesian estimate of data uncertainty $\rightarrow$ Expected Data Uncertainty

$$\mathbb{E}_{p(\boldsymbol{\theta}|\mathcal{D})}\big[\mathcal{H}[\mathrm{P}(y|\boldsymbol{x}^*, \boldsymbol{\theta})]\big] \approx \frac{1}{M}\sum_{m=1}^{M} \mathcal{H}\big[\mathrm{P}(y|\boldsymbol{x}^*, \boldsymbol{\theta}^{(m)})\big],\ \boldsymbol{\theta}^{(m)} \sim \mathrm{p}(\boldsymbol{\theta}|\mathcal{D})$$

- Not the same as entropy of the predictive posterior $\mathrm{P}(y|\boldsymbol{x}^*, \mathcal{D})$

Model Uncertainty

- If the predictions from the models are consistent

$$\underbrace{\mathcal{H}\big[\mathbb{E}_{\mathrm{p}(\boldsymbol{\theta}|\mathcal{D})}[\mathrm{P}(y|\boldsymbol{x}^*,\boldsymbol{\theta})]\big]}_{\textit{Total Uncertainty}} - \underbrace{\mathbb{E}_{\mathrm{p}(\boldsymbol{\theta}|\mathcal{D})}\big[\mathcal{H}[\mathrm{P}(y|\boldsymbol{x}^*,\boldsymbol{\theta})]\big]}_{\textit{Expected Data Uncertainty}} = 0$$

- If the predictions from the models are diverse

$$\underbrace{\mathcal{H}\big[\mathbb{E}_{\mathrm{p}(\boldsymbol{\theta}|\mathcal{D})}[\mathrm{P}(y|\boldsymbol{x}^*,\boldsymbol{\theta})]\big]}_{\textit{Total Uncertainty}} - \underbrace{\mathbb{E}_{\mathrm{p}(\boldsymbol{\theta}|\mathcal{D})}\big[\mathcal{H}[\mathrm{P}(y|\boldsymbol{x}^*,\boldsymbol{\theta})]\big]}_{\textit{Expected Data Uncertainty}} > 0$$

- Difference of the two is a measure of model uncertainty

$$\underbrace{\mathcal{I}[y,\boldsymbol{\theta}|\boldsymbol{x}^*,\mathcal{D}]}_{\textit{Model Uncertainty}} = \underbrace{\mathcal{H}\big[\mathbb{E}_{\mathrm{p}(\boldsymbol{\theta}|\mathcal{D})}[\mathrm{P}(y|\boldsymbol{x}^*,\boldsymbol{\theta})]\big]}_{\textit{Total Uncertainty}} - \underbrace{\mathbb{E}_{\mathrm{p}(\boldsymbol{\theta}|\mathcal{D})}\big[\mathcal{H}[\mathrm{P}(y|\boldsymbol{x}^*,\boldsymbol{\theta})]\big]}_{\textit{Expected Data Uncertainty}}$$

Model Uncertainty

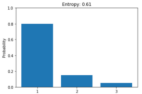
- Model uncertainty is captured by **Mutual Information** between y and $\boldsymbol{\theta}$

$$\begin{aligned}\mathcal{I}[y, \boldsymbol{\theta}|\boldsymbol{x}^*, \mathcal{D}] &= \mathcal{H}\left[\mathbb{E}_{\mathrm{p}(\boldsymbol{\theta}|\mathcal{D})}[\mathrm{P}(y|\boldsymbol{x}^*, \boldsymbol{\theta})]\right] - \mathbb{E}_{\mathrm{p}(\boldsymbol{\theta}|\mathcal{D})}\left[\mathcal{H}[\mathrm{P}(y|\boldsymbol{x}^*, \boldsymbol{\theta})]\right] \\ &= \mathrm{KL}\left[\mathrm{p}(y, \boldsymbol{\theta}|\boldsymbol{x}^*, \mathcal{D})||\mathrm{P}(y|\boldsymbol{x}^*, \mathcal{D})\mathrm{p}(\boldsymbol{\theta}|\mathcal{D})\right]\end{aligned}$$

- Mutual Information is a measure of independence between y and $\boldsymbol{\theta}$
  - Can we predict y given a knowledge of $\boldsymbol{\theta}$ and vice-versa?
- Given appropriate choice of $\mathrm{p}(\boldsymbol{\theta})$ model uncertainty captures knowledge uncertainty

Ensemble for certain in-domain input

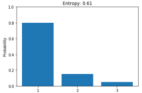
$$\mathrm{p}(\boldsymbol{\theta}) + \mathcal{D} \rightarrow \mathrm{p}(\boldsymbol{\theta}|\mathcal{D}) \longrightarrow$$

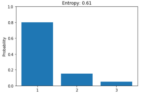
$\boldsymbol{\theta}^{(1)} \longrightarrow$ 

$\rightarrow$

$\mathrm{P}(y|\boldsymbol{x}^*, \boldsymbol{\theta}^{(1)})$

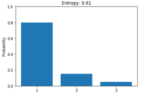
$\vdots$ $\vdots$

$\boldsymbol{\theta}^{(i)} \longrightarrow$ 

$\rightarrow$ 

$\mathrm{P}(y|\boldsymbol{x}^*, \boldsymbol{\theta}^{(i)})$ $\mathrm{P}(y|\boldsymbol{x}^*, \mathcal{D})$

$\vdots$ $\vdots$

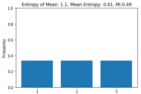
$\boldsymbol{\theta}^{(N)} \longrightarrow$ 

$\rightarrow$

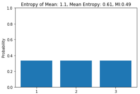
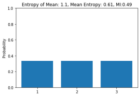
$\mathrm{P}(y|\boldsymbol{x}^*, \boldsymbol{\theta}^{(N)})$

Ensemble for uncertain in-domain input

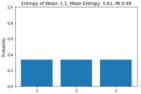
$$\mathrm{p}(\boldsymbol{\theta}) + \mathcal{D} \rightarrow \mathrm{p}(\boldsymbol{\theta}|\mathcal{D}) \longrightarrow$$

$\boldsymbol{\theta}^{(1)} \longrightarrow$  $\rightarrow$

$\mathrm{P}(y|\boldsymbol{x}^*, \boldsymbol{\theta}^{(1)})$

$\boldsymbol{\theta}^{(i)} \longrightarrow$  $\rightarrow$ 

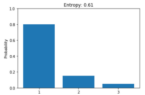
$\mathrm{P}(y|\boldsymbol{x}^*, \boldsymbol{\theta}^{(i)})$ $\mathrm{P}(y|\boldsymbol{x}^*, \mathcal{D})$

$\boldsymbol{\theta}^{(N)} \longrightarrow$  $\rightarrow$

$\mathrm{P}(y|\boldsymbol{x}^*, \boldsymbol{\theta}^{(N)})$

Ensemble for Out-of-Domain input

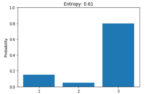
$$\mathrm{p}(\boldsymbol{\theta}) + \mathcal{D} \rightarrow \mathrm{p}(\boldsymbol{\theta}|\mathcal{D}) \longrightarrow$$

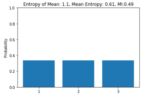
$\boldsymbol{\theta}^{(1)} \longrightarrow$ 

$\rightarrow$

$\mathrm{P}(y|\boldsymbol{x}^*, \boldsymbol{\theta}^{(1)})$

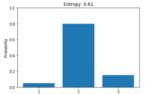
$\vdots$ $\quad$ $\vdots$

$\boldsymbol{\theta}^{(i)} \longrightarrow$ 

$\rightarrow$ 

$\mathrm{P}(y|\boldsymbol{x}^*, \boldsymbol{\theta}^{(i)})$ $\qquad$ $\mathrm{P}(y|\boldsymbol{x}^*, \mathcal{D})$

$\vdots$ $\quad$ $\vdots$

$\boldsymbol{\theta}^{(N)} \longrightarrow$ 

$\rightarrow$

$\mathrm{P}(y|\boldsymbol{x}^*, \boldsymbol{\theta}^{(N)})$

Approximate Inference

- Bayes' Rule is **intractable** for neural networks
  - Use variational approximation $\mathrm{q}(\boldsymbol{\theta}) \approx \mathrm{p}(\boldsymbol{\theta}|\mathcal{D})$
- Inference is **intractable** for neural networks
  - **Must** approximate integral using via ensemble approaches

Approximate Inference

- Variational Inference:
  - Bayes by Backprop [Blundell et al., 2015]
  - Probabalistic Backpropagation [Hernández-Lobato and Adams, 2015]
- Monte-Carlo Methods:
  - Monte-Carlo Dropout [Gal, 2016, Gal and Ghahramani, 2016]
  - Stochastic Gradient Langevin Dynamics [Welling and Teh, 2011]
  - Fast-Ensembling via Mode Connectivity [Garipov et al., 2018]
  - Stochastic Weight Averaging Gaussian (SWAG) [Maddox et al., 2019]
- Non-Bayesian Ensembles:
  - Bootstrap DQN [Osband et al., 2016]
  - Deep Ensembles [Lakshminarayanan et al., 2017]

Limitations

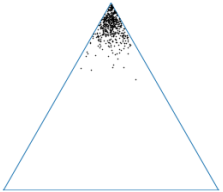
- Hard to guarantee diverse $\{\mathrm{P}(y|\boldsymbol{x}^*, \boldsymbol{\theta}^{(m)})\}_{m=1}^{M}$ for OOD $\boldsymbol{x}^*$
- Diversity of ensemble depends on:
  - Selection of prior
  - Nature of approximations
  - Architecture of network
  - Properties and size of data
- May be computationally expensive

Overview

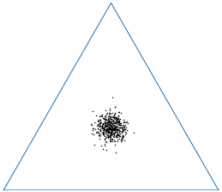
1. Sources of Uncertainty in Predictions
2. Ensemble Approaches
3. **Prior Networks**
4. Assessment of Uncertainty Quality
5. Ensemble Distribution Distillation

Distributions on a Simplex

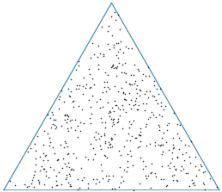
- Ensemble $\{\mathrm{P}(y|\boldsymbol{x}^*, \boldsymbol{\theta}^{(m)})\}_{m=1}^{M}$ can be visualized on a simplex



(a) Confident



(b) Data Uncertainty



(c) Knowledge Uncertainty

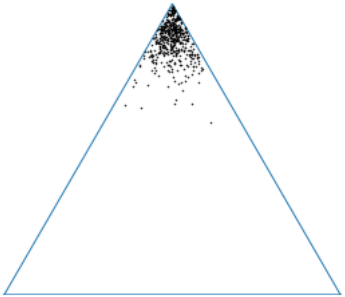
- Same as sampling from **implicit** Distribution over output Distributions

$$\mathrm{P}(y|\boldsymbol{x}^*, \boldsymbol{\theta}^{(m)}) \sim \mathrm{p}(\boldsymbol{\theta}|\mathcal{D}) \equiv \boldsymbol{\mu}^{(m)} \sim \mathrm{p}(\boldsymbol{\mu}|\boldsymbol{x}^*, \mathcal{D})$$

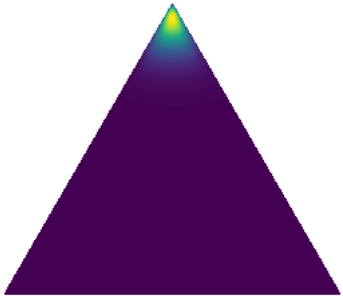
Distributions on a Simplex (cont)

- Expanding out $\boldsymbol{\mu}^{(m)} = \begin{bmatrix} \mathrm{P}(y = \omega_1) \\ \mathrm{P}(y = \omega_2) \\ \vdots \\ \mathrm{P}(y = \omega_K) \end{bmatrix}$, where each $\boldsymbol{\mu}^{(m)}$ is a point on a simplex.

Distribution over Distributions

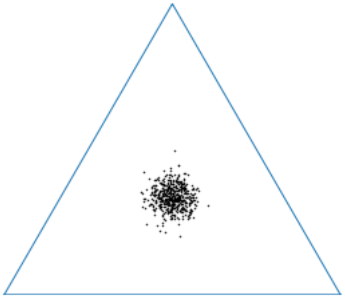


(a) $\{\boldsymbol{\mu}^{(m)}\}_{m=1}^{M}$

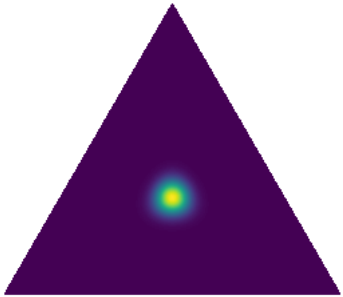


(b) $\mathrm{p}(\boldsymbol{\mu}|\boldsymbol{x}^*, \mathcal{D})$

Distribution over Distributions

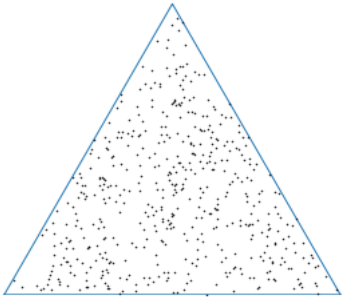


(a) $\{\boldsymbol{\mu}^{(m)}\}_{m=1}^{M}$

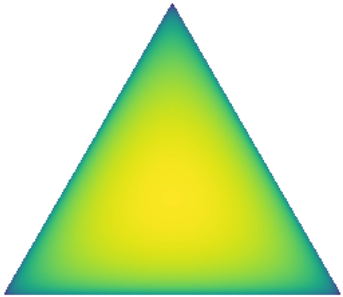


(b) $\mathrm{p}(\boldsymbol{\mu}|\boldsymbol{x}^*, \mathcal{D})$

Distribution over Distributions



(a) $\{\boldsymbol{\mu}^{(m)}\}_{m=1}^{M}$



(b) $\mathrm{p}(\boldsymbol{\mu}|\boldsymbol{x}^*, \mathcal{D})$

Prior Networks [Malinin and Gales, 2018]

- **Explicitly** model $\mathrm{p}(\boldsymbol{\mu}|\boldsymbol{x}^*, \mathcal{D})$ using a Prior Network $\mathrm{p}(\boldsymbol{\mu}|\boldsymbol{x}^*; \hat{\boldsymbol{\theta}})$

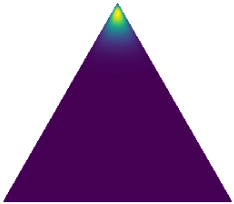
$$\mathrm{p}(\boldsymbol{\mu}|\boldsymbol{x}^*; \hat{\boldsymbol{\theta}}) \approx \mathrm{p}(\boldsymbol{\mu}|\boldsymbol{x}^*, \mathcal{D})$$

- Predictive posterior distribution is given by expected categorical

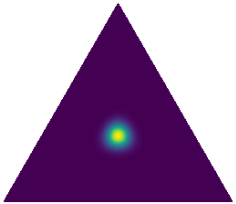
$$\mathrm{P}(y|\boldsymbol{x}^*; \hat{\boldsymbol{\theta}}) = \mathbb{E}_{\mathrm{p}(\boldsymbol{\mu}|\boldsymbol{x}^*; \hat{\boldsymbol{\theta}})}\left[\mathrm{p}(y|\boldsymbol{\mu})\right] = \hat{\boldsymbol{\mu}}$$

Prior Networks

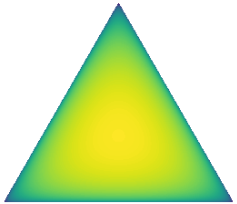
- Construct $\mathrm{p}(\boldsymbol{\mu}|\boldsymbol{x}^*, \hat{\boldsymbol{\theta}})$ to emulate ensemble



(a) Certain



(b) Data Uncertainty

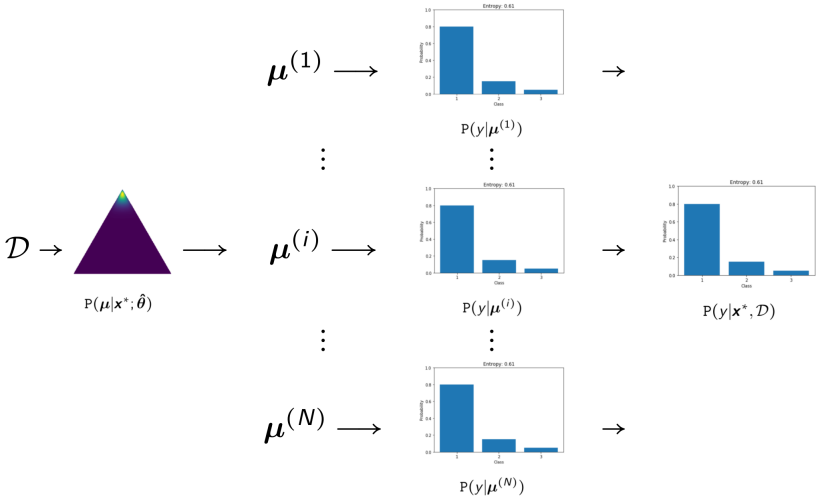


(c) Knowledge Uncertainty

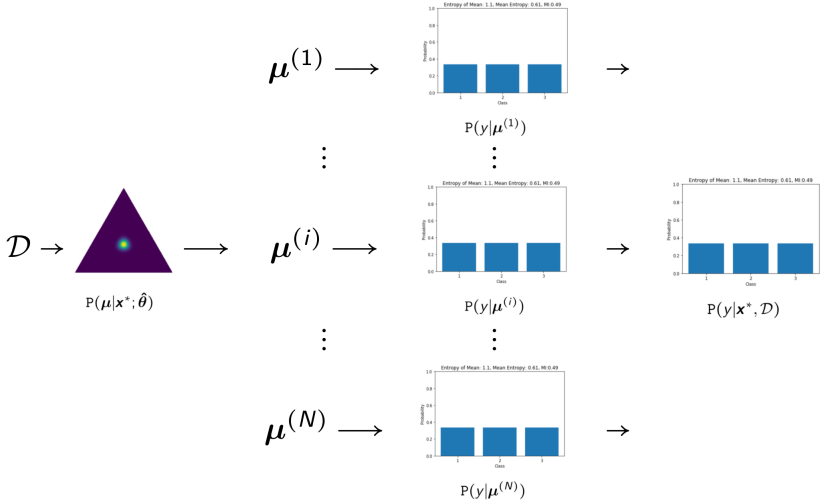
Prior Networks

- Behaviour of Ensemble distribution over distributions
  - Controlled via prior $\mathrm{p}(\boldsymbol{\theta})$ and inference scheme
- Behaviour of Prior Networks distribution over distributions
  - Controlled via loss function and training data $\mathcal{D}$

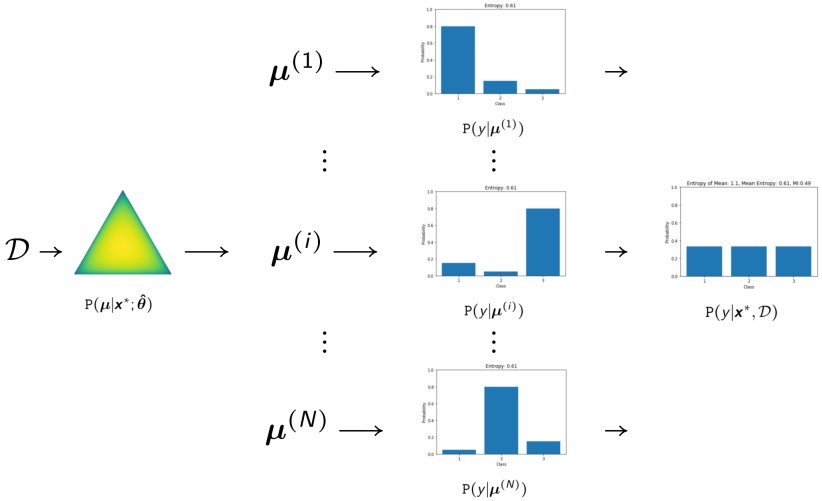
Distributions over Distributions via Prior Networks



Distributions over Distributions via Prior Networks



Distributions over Distributions via Prior Networks



Uncertainty Measures for Prior Networks

- Ensemble Mutual Information:

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

- Mutual Information between y and $\boldsymbol{\mu}$

$$\underbrace{\mathcal{I}[y, \boldsymbol{\mu} | \boldsymbol{x}^*; \hat{\boldsymbol{\theta}}]}_{\textit{Knowledge Uncertainty}} = \underbrace{\mathcal{H}\big[\mathbb{E}_{\mathrm{p}(\boldsymbol{\mu}|\boldsymbol{x}^*; \hat{\boldsymbol{\theta}})}[\mathrm{P}(y|\boldsymbol{\mu})]\big]}_{\textit{Total Uncertainty}} - \underbrace{\mathbb{E}_{\mathrm{p}(\boldsymbol{\mu}|\boldsymbol{x}^*; \hat{\boldsymbol{\theta}})}\big[\mathcal{H}[\mathrm{P}(y|\boldsymbol{\mu})]\big]}_{\textit{Expected Data Uncertainty}}$$

Prior Network Construction

- A Prior Network parametrizes the **Dirichlet Distribution**

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

- Dirichlet Distribution $\rightarrow$ Distribution over simplex
  - Conjugate prior to categorical distribution
  - Convenient properties $\rightarrow$ analytically tractable

Reminder - Dirichlet Distribution

- Dirichlet is a distribution over categorical distributions

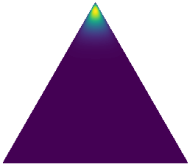
$$\texttt{Dir}(\boldsymbol{\mu}|\boldsymbol{\alpha}) = \frac{\Gamma(\alpha_0)}{\prod_{c=1}^{K}\Gamma(\alpha_c)}\prod_{c=1}^{K}\pi_c^{\alpha_c-1}; \quad \alpha_0 = \sum_{c=1}^{K}\alpha_c$$

  - Parameterised by concentration parameters: $\boldsymbol{\alpha}$, $\alpha_c > 0$
  - Expected label posteriors given by

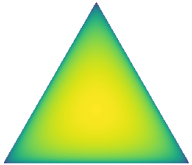
$$\hat{\mathrm{P}}(y=\omega_c) = \hat{\mu}_c = \frac{\alpha_c}{\sum_{k=1}^{K}\alpha_k}$$

Prior Network Construction

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



(a) In-Domain Target



(b) OOD Target

Target Concentration Parameters

- To train the prior network we need a target distribution $\mathrm{p}(\boldsymbol{\mu}|\boldsymbol{\beta})$ for $\boldsymbol{x}^{(i)}$
  - We **want** training data $\{\boldsymbol{\beta}^{(i)}, \boldsymbol{x}^{(i)}\}_{i=1}^{N}$
  - ... but have training data $\{y^{(i)}, \boldsymbol{x}^{(i)}\}_{i=1}^{N}$, where $y^{(i)} \in \{\omega_1, \ldots, \omega_K\}$
- Solution $\rightarrow$ specify concentration parameters $\boldsymbol{\beta}^{(c)}$ as a function of target class y
  - Need $\boldsymbol{\beta}^{(c)}$ to reflect "confidence" in sample
  - Need distribution to yield correct class
  - $\beta_k > 0 \ \forall k$

Target Concentration Parameters

- Consider setting $\boldsymbol{\beta}^{(c)}$ as follows $\rightarrow$

$$\beta_k^{(c)} = \begin{cases} \beta + 1 & \text{if } c = k \\ 1 & \text{if } c \neq k \end{cases}$$

- If β is large $\rightarrow$
  - Specifies a sharp Dirichlet at corner of simplex corresponding to target class.
- If β is zero $\rightarrow$
  - Specifies a flat Dirichlet distribution.

KL-divergence Losses

- We can consider two loss functions - *Forward KL-Divergence* $\rightarrow$

$$\mathcal{L}^{KL}(y, \boldsymbol{x}, \boldsymbol{\theta}; \beta) = \sum_{c=1}^{K} \mathcal{I}(y = \omega_c) \cdot \mathrm{KL}[\mathrm{Dir}(\boldsymbol{\mu}|\boldsymbol{\beta}^{(c)})||\mathrm{p}(\boldsymbol{\mu}|\boldsymbol{x}; \boldsymbol{\theta})]$$

- ... or *reverse KL-Divergence* $\rightarrow$

$$\mathcal{L}^{RKL}(y, \boldsymbol{x}, \boldsymbol{\theta}; \beta) = \sum_{c=1}^{K} \mathcal{I}(y = \omega_c) \cdot \mathrm{KL}[\mathrm{p}(\boldsymbol{\mu}|\boldsymbol{x}; \boldsymbol{\theta})||\mathrm{Dir}(\boldsymbol{\mu}|\boldsymbol{\beta}^{(c)})]$$

Reminder - Kullback-Leibler Divergence

- Standard "measure" between distributions

$$\mathrm{KL}[\mathrm{P}_{\mathrm{tr}}(y|\boldsymbol{x})||\mathrm{P}(y|\boldsymbol{x};\boldsymbol{\theta})] = \sum_{c=1}^{K} \mathrm{P}_{\mathrm{tr}}(y=\omega_c|\boldsymbol{x}) \ln\left(\frac{\mathrm{P}_{\mathrm{tr}}(y=\omega_c|\boldsymbol{x})}{\mathrm{P}(y=\omega_c|\boldsymbol{x};\boldsymbol{\theta})}\right)$$

- Variational optimization often yields reverse KL for training

$$\mathrm{KL}[\mathrm{P}(y|\boldsymbol{x};\boldsymbol{\theta})||\mathrm{P}_{\mathrm{tr}}(y|\boldsymbol{x})] = \sum_{c=1}^{K} \mathrm{P}(y=\omega_c|\boldsymbol{x};\boldsymbol{\theta}) \ln\left(\frac{\mathrm{P}(y=\omega_c|\boldsymbol{x};\boldsymbol{\theta})}{\mathrm{P}_{\mathrm{tr}}(y=\omega_c|\boldsymbol{x})}\right)$$

Forward KL-divergence Loss

- Consider expectation of *forward* KL-div loss wrt. empirical distribution $\hat{\mathrm{p}}(\boldsymbol{x}, y) \rightarrow$

$$\mathcal{L}^{KL}(\boldsymbol{\theta}; \beta) = \mathbb{E}_{\hat{\mathrm{p}}_{\mathtt{tr}}(\boldsymbol{x},y)}\Big[\sum_{c=1}^{K} \mathcal{I}(y = \omega_c) \cdot \mathtt{KL}[\mathtt{Dir}(\boldsymbol{\mu}|\boldsymbol{\beta}^{(c)})||\mathrm{p}(\boldsymbol{\mu}|\boldsymbol{x}; \boldsymbol{\theta})]\Big]$$

$$= \mathbb{E}_{\hat{\mathrm{p}}_{\mathtt{tr}}(\boldsymbol{x})}\Big[\mathtt{KL}[\sum_{c=1}^{K} \hat{\mathrm{P}}_{\mathtt{tr}}(y = \omega_c|\boldsymbol{x})\mathtt{Dir}(\boldsymbol{\mu}|\boldsymbol{\beta}^{(c)})||\mathrm{p}(\boldsymbol{\mu}|\boldsymbol{x}; \boldsymbol{\theta})]\Big]$$

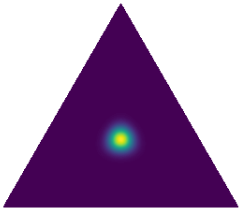
- Target distribution becomes a *mixture of Dirichlets* $\rightarrow$

$$\mathrm{p}(\boldsymbol{\mu}) = \sum_{c=1}^{K} \hat{\mathrm{P}}_{\mathtt{tr}}(y = \omega_c|\boldsymbol{x})\mathtt{Dir}(\boldsymbol{\mu}|\boldsymbol{\beta}^{(c)})$$

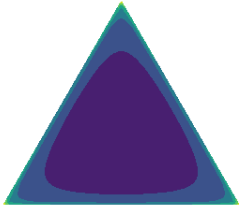
- Cannot be modelled using a single Dirichlet!

Forward KL-divergence Loss

- Forward KL-divergence is zero avoiding $\rightarrow$
  - Model $\mathrm{p}(\boldsymbol{\mu}|\boldsymbol{x};\boldsymbol{\theta})$ will try to cover **each mode** of $\sum_{c=1}^{K}\hat{\mathrm{P}}_{\mathtt{tr}}(y=\omega_c|\boldsymbol{x})\mathtt{Dir}(\boldsymbol{\mu}|\boldsymbol{\beta}^{(c)})$
- Leads to **undesired behaviour**!



(a) Want



(b) Get

Reverse KL-divergence Loss [Malinin and Gales, 2019]

- Consider expectation of *reverse* KL-div loss wrt. empirical distribution $\hat{\mathrm{p}}(\boldsymbol{x}, y) \rightarrow$

$$\begin{aligned}
\mathcal{L}^{RKL}(\boldsymbol{\theta}; \beta) &= \mathbb{E}_{\hat{\mathrm{p}}_{\mathtt{tr}}(\boldsymbol{x})}\Big[\sum_{c=1}^{K} \hat{\mathrm{P}}_{\mathtt{tr}}(y = \omega_c|\boldsymbol{x})\mathrm{KL}\big[\mathrm{p}(\boldsymbol{\mu}|\boldsymbol{x};\boldsymbol{\theta})||\mathtt{Dir}(\boldsymbol{\mu}|\boldsymbol{\beta}^{(c)})\big]\Big] \\
&= \mathbb{E}_{\hat{\mathrm{p}}_{\mathtt{tr}}(\boldsymbol{x})}\Big[\mathbb{E}_{\mathrm{p}(\boldsymbol{\mu}|\boldsymbol{x};\boldsymbol{\theta})}\big[\ln \mathrm{p}(\boldsymbol{\mu}|\boldsymbol{x};\boldsymbol{\theta}) - \ln \prod_{c=1}^{K} \mathtt{Dir}(\boldsymbol{\mu}|\boldsymbol{\beta}^{(c)})^{\hat{\mathrm{P}}_{\mathtt{tr}}(y=\omega_c|\boldsymbol{x})}\big]\Big] \\
&= \mathbb{E}_{\hat{\mathrm{p}}_{\mathtt{tr}}(\boldsymbol{x})}\Big[\mathrm{KL}\big[\mathrm{p}(\boldsymbol{\mu}|\boldsymbol{x};\boldsymbol{\theta})||\mathtt{Dir}(\boldsymbol{\mu}|\sum_{c=1}^{K} \hat{\mathrm{P}}_{\mathtt{tr}}(y = \omega_c|\boldsymbol{x}) \cdot \boldsymbol{\beta}^{(c)})\big]\Big] + C
\end{aligned}$$

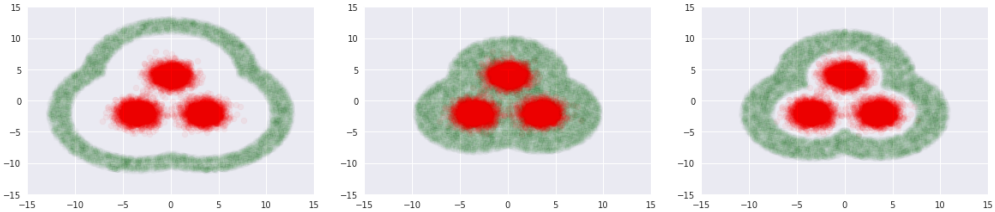
- Expectation induces product of target Dirichlet distributions
  - target becomes a uni-modal Dirichlet distribution focused on appropriate location!

Prior Network Construction

- Reverse KL $\rightarrow$ appropriate loss function.
  - $\beta = 1e2$ (or some other large value...) in-domain
  - $\beta = 0$ out-of-domain.
- But how to obtain out-of-domain training data $\mathcal{D}_{OOD} = \hat{\mathrm{p}}_{out}(\boldsymbol{x})$?
- Can consider 3 approaches $\rightarrow$
  - Use a different dataset, eg: CIFAR10 vs CIFAR100
  - Synthesize using generative model (VAE/GAN)
  - Generate using adversarial attacks

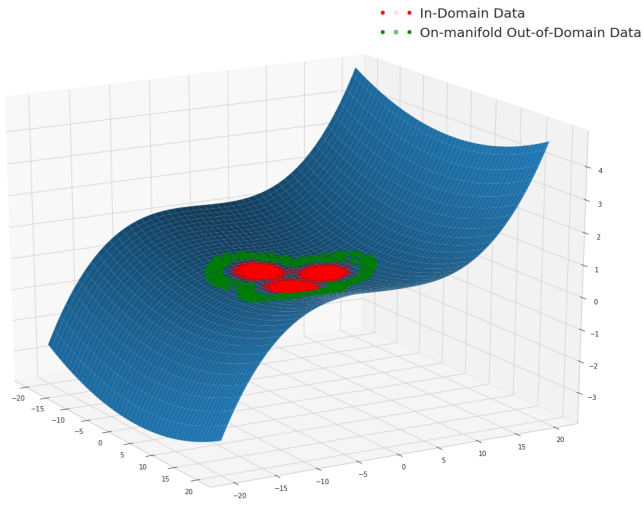
Prior Network Construction

- Out-of-domain (OOD) training data must be on *boundary* on in-domain region →
  - Too loose → Some OOD might be considered in-domain
  - Too tight → Some in-domain might be considered OOD
- OOD data should also lie on the same manifold

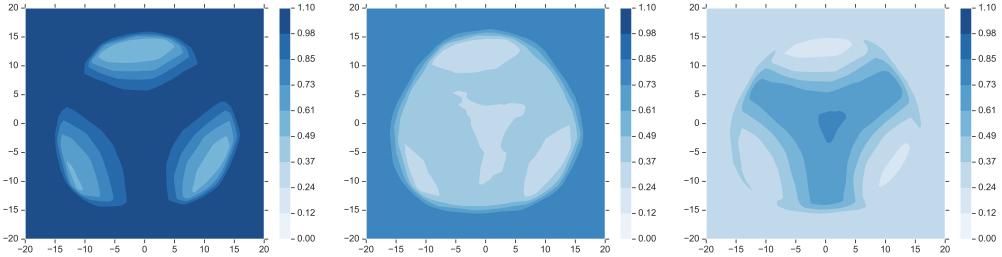


(a) Too Loose (b) Too Tight (c) Good

Prior Network Construction

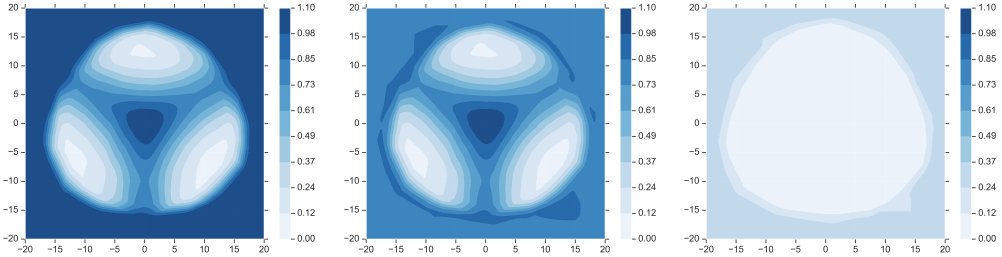


Prior Networks trained with *forward* KL-divergence loss on Artificial Data



(a) Total Uncertainty (b) Data Uncertainty (c) Knowledge Uncertainty

Prior Networks trained with *reverse* KL-divergence loss on Artificial Data



(a) Total Uncertainty (b) Data Uncertainty (c) Knowledge Uncertainty

Overview

1. Sources of Uncertainty in Predictions
2. Ensemble Approaches
3. Prior Networks
4. **Assessment of Uncertainty Quality**
5. Ensemble Distribution Distillation

Assessment of Uncertainty Quality

- Quality of uncertainty estimates of commonly assessed via
  - Log-likelihood of test data $\mathcal{D}_{tst} = \{\boldsymbol{x}^*_{(i)}, y^*_{(i)}\}$
  - Calibration (Expected Calibration Error)
- Test Log-likelihood $\rightarrow$

$$NLL = \frac{1}{N}\sum_{i=1}^{N} \ln \mathrm{P}(y^*_{(i)}|\boldsymbol{x}^*_{(i)}, \mathcal{D})$$

- Calibration $\rightarrow$ Does confidences correspond to long-run accuracy?
- Informative quality statistics, but weakly related to application

Assessment of Uncertainty Quality

- Uncertainty should be assessed in the context of an **application**
- Threshold-based outlier detection $\rightarrow$
  - Misclassification Detection [Hendrycks and Gimpel, 2016]
  - Out-of-distribution input Detection
  - Adversarial Attack Detection [Malinin and Gales, 2019]
- Active Learning [Gal, 2016]
- Reinforcement Learning uncertainty-driven exploration [Osband et al., 2016]
- Other...

Misclassification and Out-of-Distribution detection

- Threshold-based detection $\rightarrow$

$$\mathcal{I}_T(\boldsymbol{x}) = \begin{cases} 1, & \mathcal{H}(\boldsymbol{x}) > T \\ 0, & \mathcal{H}(\boldsymbol{x}) \leq T \end{cases}$$

- If $\mathcal{I}_T(\boldsymbol{x}) = 1 \rightarrow$ outlier
- If $\mathcal{I}_T(\boldsymbol{x}) = 0 \rightarrow$ normal
- Evaluate performance using
  - Area under Precision-Recall Curve (AUPR) $\rightarrow$ Misclassification Detection
  - Area under ROC curve (AUROC) $\rightarrow$ Out-of-distribution Detection

Misclassification and Out-of-Distribution detection

- Area under Precision Recall Curve
  - Good for mis-balanced datasets
  - Good performance $\rightarrow$ 100%
  - Random performance $\rightarrow$ Classifier % Error
- Area under ROC Curve
  - Good for balanced datasets
  - Good performance $\rightarrow$ 100 %
  - Random performance $\rightarrow$ 50 %

CIFAR10 Experiments

- Compare Prior Networks (PN) to DeepEnsembles (10 models)
- Prior Networks trained on:
  - CIFAR-10 (ID)
  - CIFAR-100 (OOD)
- OOD test data:
  - SVHN
  - LSUN
  - TinyImageNet (TIM)

Classification Error Rate

Dataset	DNN	PN-KL	PN-RKL	Ensemble
CIFAR-10	8.0 ±0.4	14.7 ±0.4	7.5 ±0.3	**6.6** ± NA
CIFAR-100	30.4 ±0.6	-	28.1 ±0.2	**26.9** ± NA
TinyImageNet	41.7 ±0.4	-	40.3 ±0.4	**36.9** ± NA

Misclassification Detection

Model	Total Uncertainty	Knowledge Uncertainty	% Error
Ensemble	48.0	38.6	**6.2**
PN-RKL	40.5	35.0	7.5

Table: CIFAR-10 misclassification detection results (mean % AUPR across 10 rand. inits).

Out-of-Distribution Detection

OOD test Data	Model	Total Uncertainty	Knowledge Uncertainty
SVHN	Ensemble	92.0	89.5
	PN-RKL	98.2	**98.3**
LSUN	Ensemble	94.3	93.2
	PN-RKL	95.7	**95.8**
TinyImageNet	Ensemble	91.5	90.3
	PN-RKL	95.7	**95.8**

Table: CIFAR-10 out-of-domain detection results (mean % AUROC across 10 rand. inits).

Overview

1. Sources of Uncertainty in Predictions
2. Ensemble Approaches
3. Prior Networks
4. Assessment of Uncertainty Quality
5. **Ensemble Distribution Distillation (EnD^2)**

Ensemble Distillation (EnD)

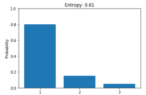
$$\mathrm{p}(\boldsymbol{\theta}) + \mathcal{D} \rightarrow \mathrm{p}(\boldsymbol{\theta}|\mathcal{D}) \longrightarrow \boldsymbol{\theta}^{(1)} \longrightarrow \mathrm{P}(y|\boldsymbol{x}^*, \boldsymbol{\theta}^{(1)}) \rightarrow$$

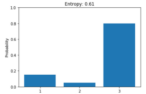
$$\vdots \qquad \vdots$$

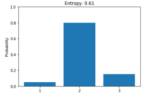
$$\boldsymbol{\theta}^{(i)} \longrightarrow \mathrm{P}(y|\boldsymbol{x}^*, \boldsymbol{\theta}^{(i)}) \rightarrow \mathrm{P}(y|\boldsymbol{x}^*, \mathcal{D})$$

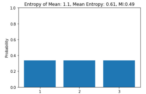
$$\vdots \qquad \vdots$$

$$\boldsymbol{\theta}^{(N)} \longrightarrow \mathrm{P}(y|\boldsymbol{x}^*, \boldsymbol{\theta}^{(N)}) \rightarrow$$









Ensemble Distillation (EnD) [Hinton et al., 2015, Korattikara et al., 2015]

- Ensembles are computationally expensive
  - Distill an ensemble into a **single** model

$$\{\mathrm{P}(y|\boldsymbol{x}, \boldsymbol{\theta}^{(m)})\}_{m=1}^{M} \rightarrow \mathrm{P}(y|\boldsymbol{x}, \hat{\boldsymbol{\theta}})$$

- Minimize KL-divergence to mean of ensemble:

$$\mathcal{L}(\hat{\boldsymbol{\theta}}, \mathcal{D}) = \mathbb{E}_{\mathrm{p}(\boldsymbol{x})}\Big[\mathrm{KL}\big[\mathbb{E}_{\mathrm{p}(\boldsymbol{\theta}|\mathcal{D})}[\mathrm{P}(y|\boldsymbol{x}, \boldsymbol{\theta})]||\mathrm{P}(y|\boldsymbol{x}, \hat{\boldsymbol{\theta}})\big]\Big]$$

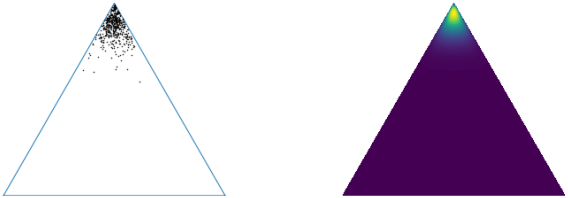
- Computational Performance gain
- Robustness to Adversarial Attack (Defensive Distillation)

Ensemble Distillation (EnD)

- EnD $\rightarrow$ model captures only *mean* of ensemble
- Diversity of ensemble is lost $\rightarrow$
  - Cannot separate measures of uncertainty
- Solution $\rightarrow$ Ensemble Distribution Distillation

Ensemble Distribution Distillation (End^2) [Malinin et al., 2019]

- Distill an ensemble into a **single** Prior Network



$$\left\{\mathrm{P}(y|\boldsymbol{x},\boldsymbol{\theta}^{(m)})\right\}_{m=1}^{M} \quad \rightarrow \quad \mathrm{p}(\boldsymbol{\mu}|\boldsymbol{x};\hat{\boldsymbol{\theta}})$$

Ensemble Distribution Distillation (End^2)

- Parameterize a Dirichlet distribution using Neural Network:

$$\mathrm{p}(\boldsymbol{\mu}|\boldsymbol{x};\boldsymbol{\theta}) = \mathtt{Dir}(\boldsymbol{\mu};\boldsymbol{\alpha}), \quad \boldsymbol{\alpha} = \boldsymbol{f}(\boldsymbol{x};\boldsymbol{\theta}), \quad \alpha_c \geq 0$$

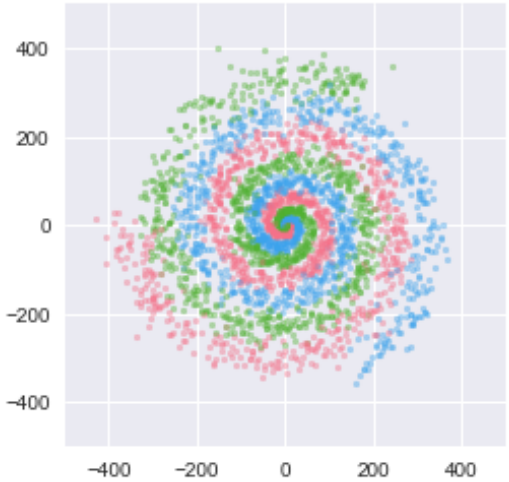
- Training data are ensemble predictions for every input:

$$\mathcal{D} = \left\{ \left\{ p(y|\boldsymbol{x}^{(i)};\boldsymbol{\theta}^{(j)}), \boldsymbol{x}^{(i)} \right\}_{j=1}^{N} \right\}_{i=1}^{M} \sim \hat{\mathrm{p}}(\boldsymbol{\mu}, \boldsymbol{x})$$

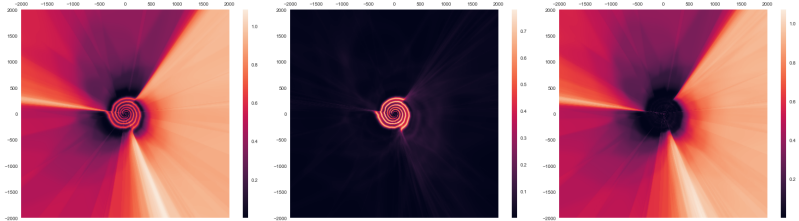
- Train via Maximum Likelihood:

$$\mathcal{L}(\boldsymbol{\theta}, \mathcal{D}) = -\mathbb{E}_{\hat{\mathrm{p}}(\boldsymbol{x})}\Big[\mathbb{E}_{\hat{\mathrm{p}}(\boldsymbol{\mu}|\boldsymbol{x})}[\ln \mathrm{p}(\boldsymbol{\mu}|\boldsymbol{x};\boldsymbol{\theta})]\Big]$$

Ensemble Distribution Distillation: Uncertainty



Ensemble Distribution Distillation: Uncertainty



(a) Ensm - Total (b) Ensm - Data (c) Ensm - Knowledge

(d) EnD^2 - Total (e) EnD^2 - Data (f) EnD^2 - Knowledge

Ensemble Distribution Distillation: Image Classification

Dataset	Individual	Ensemble	EnD	EnD^2
CIFAR-10	8.0	**6.2**	6.7	6.9
CIFAR-100	30.4	**26.3**	28.3	28.2
TinyImageNet	41.8	**36.6**	38.3	37.6

Table: Classification Performance (% Error).

Ensemble Distribution Distillation: Misclassification Detection

Model	CIFAR-10	
	AUPR	Error (%)
Individual	48.0	8.0
EnD	44.5	6.7
EnD^2	46.5	6.9
Ensemble	43.9	6.2

Table: Misclassification detection performance (% AUPR) for CIFAR-10.

Ensemble Distribution Distillation: OOD Detection

Test OOD Dataset	Model	CIFAR-10		CIFAR-100	
		Total Unc.	Knowledge Unc.	Total Unc.	Knowledge Unc.
LSUN	Individual	91.3	-	75.6	-
	EnD	89.0	-	76.5	-
	EnD^2	94.4	**95.3**	83.5	86.9
	Ensemble	94.5	94.4	82.4	**88.4**
TIM	Individual	88.9	-	70.5	-
	EnD	86.9	-	70.0	-
	EnD^2	91.3	**91.8**	76.4	79.3
	Ensemble	**91.8**	91.4	76.6	**81.7**

Table: OOD detection performance (% AUC-ROC) for CIFAR-10 and CIFAR-100 models.

Conclusions

- Sources of Uncertainty
  - Data Uncertainty $\rightarrow$ Class overlap
  - Knowledge Uncertainty $\rightarrow$ Out-of-distribution inputs
- Ensembles $\rightarrow$ Implicit distribution over output distributions
  - Theoretically motivated separation of uncertainty sources
  - Control behavior via prior $\mathrm{P}(\boldsymbol{\theta})$
- Prior Networks $\rightarrow$ Explicit distribution over output distributions
  - Emulates ensembles
  - Control behavior via data $\mathcal{D}$
- Uncertainty Assessment
  - Assess uncertainty on **applications!**
- Ensemble Distribution Distillation $\rightarrow$ Distill ensemble into a Prior Network
  - Hybrid PN/Ensemble approach
  - + Classification / Misc. Detection Performance

Thank You!

Any questions?

References I

[Blundell et al., 2015] Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D. (2015).
Weight uncertainty in neural networks.
arXiv preprint arXiv:1505.05424.

[Gal, 2016] Gal, Y. (2016).
Uncertainty in Deep Learning.
PhD thesis, University of Cambridge.

[Gal and Ghahramani, 2016] Gal, Y. and Ghahramani, Z. (2016).
Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning.
In *Proc. 33rd International Conference on Machine Learning (ICML-16)*.

References II

[Garipov et al., 2018] Garipov, T., Izmailov, P., Podoprikhin, D., Vetrov, D. P., and Wilson, A. G. (2018).
Loss surfaces, mode connectivity, and fast ensembling of dnns.
In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems 31*, pages 8789–8798. Curran Associates, Inc.

[Hendrycks and Gimpel, 2016] Hendrycks, D. and Gimpel, K. (2016).
A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks.
http://arxiv.org/abs/1610.02136.
arXiv:1610.02136.

References III

[Hernández-Lobato and Adams, 2015] Hernández-Lobato, J. M. and Adams, R. (2015).
Probabilistic backpropagation for scalable learning of bayesian neural networks.
In *International Conference on Machine Learning*, pages 1861–1869.

[Hinton et al., 2015] Hinton, G., Vinyals, O., and Dean, J. (2015).
Distilling the knowledge in a neural network.
In *NIPS Deep Learning and Representation Learning Workshop*.

References IV

[Korattikara et al., 2015] Korattikara, A., Rathod, V., Murphy, K. P., and Welling, M. (2015).
Bayesian dark knowledge.
In Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M., and Garnett, R., editors, *Advances in Neural Information Processing Systems 28*, pages 3438–3446. Curran Associates, Inc.

[Lakshminarayanan et al., 2017] Lakshminarayanan, B., Pritzel, A., and Blundell, C. (2017).
Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles.
In *Proc. Conference on Neural Information Processing Systems (NIPS)*.

References V

[Maddox et al., 2019] Maddox, W., Garipov, T., Izmailov, P., Vetrov, D. P., and Wilson, A. G. (2019).
A simple baseline for bayesian uncertainty in deep learning.
CoRR, abs/1902.02476.

[Malinin and Gales, 2018] Malinin, A. and Gales, M. (2018).
Predictive uncertainty estimation via prior networks.
In *Advances in Neural Information Processing Systems*, pages 7047–7058.

[Malinin and Gales, 2019] Malinin, A. and Gales, M. (2019).
Reverse kl-divergence training of prior networks: Improved uncertainty and adversarial robustness.
arXiv preprint arXiv:1905.13472.

References VI

[Malinin et al., 2019] Malinin, A., Mlodozeniec, B., and Gales, M. (2019).
Ensemble distribution distillation.
arXiv preprint arXiv:1905.00076.

[Osband et al., 2016] Osband, I., Blundell, C., Pritzel, A., and Van Roy, B. (2016).
Deep exploration via bootstrapped dqn.
In Lee, D. D., Sugiyama, M., Luxburg, U. V., Guyon, I., and Garnett, R., editors, *Advances in Neural Information Processing Systems 29*, pages 4026–4034. Curran Associates, Inc.

[Welling and Teh, 2011] Welling, M. and Teh, Y. W. (2011).
Bayesian Learning via Stochastic Gradient Langevin Dynamics.
In *Proc. International Conference on Machine Learning (ICML)*.