Density Ratio Estimators in Variational Bayesian Machine Learning

Lammy

Department of Mathematics and Statistics
UNSW

Statistics Honours, 2018

Outline

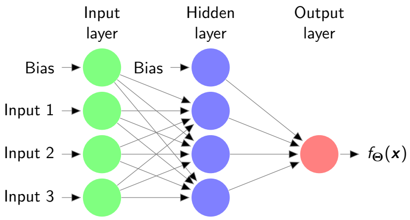
Outline

1. Background Info
   - Neural Networks
   - (Amortized) Variational Inference
   - Density Ratio Estimation
2. Activation Function Experiment
3. Theory Break
4. Experiments
   - Inference Experiment
   - Generation Experiment
5. Further Estimator Loss Function Analysis

Neural Networks

Overall Structure

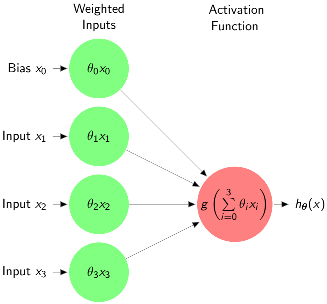
- Mathematical model based off human brain.
- Objective is to approximate a function f^* using mapping with parameters Θ: $\mathbf{f}_\Theta(\boldsymbol{x})$
- Universal Approximation Theorem states a neural network can approximate (almost) any function if it is complex enough.
- Consists of layers of nodes:



Neural Networks

Individual Node Structure

- Each node is a generalised linear model of preceding layer output.
- Weights θ are randomly initialised from normal or uniform distribution.
- Bias $x_0 = 1$ has role of intercept term in typical regression.



Neural Networks

Activation Functions

- Used to map node output to certain space.
- Every node except input nodes has an activation function.
- We are mostly concerned with activation function of output layer, which maps $\mathbb{R}$ to some space:
    - Linear (no) activation function $g(x) = x$ outputs in $\mathbb{R}$.
    - Rectified Linear Unit (ReLU) activation function $g(x) = \max\{0, x\}$ in $[0, \infty)$.
    - Sigmoid activation function $g(x) = (1 + \exp(-x))^{-1}$ in $(0, 1)$.

Neural Networks
Training

- Weights and biases trained such that (ideally convex) loss function is minimized e.g. Mean Squared Error: $\min_{\Theta} \frac{1}{2}||\boldsymbol{y} - \boldsymbol{f}_{\Theta}(\boldsymbol{x})||_2^2$.
- Back-propagation finds partial derivatives of loss function with respect to weights by propagating error backwards through network.
- Gradient descent uses these partial derivatives to optimize network.

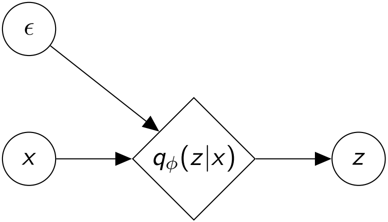
Outline

- Fundamental problem in Bayesian computation is to estimate posterior densities $p(z|x)$.

$$p(z|x) = \frac{p(z,x)}{p(x)} = \frac{p(z)p(x|z)}{\int_{\mathcal{Z}} p(z,x)dz}$$

- Problems arise when $\int_{\mathcal{Z}} p(z,x)dz$ is computationally intractable.
- Typical MCMC methods are slow with large datasets or high dimensional data.
- Variational Inference is a solution.

- Amortized variational inference approximates $p(z|x)$ with a different distribution $q_\phi(z|x)$.
- $q_\phi(z|x)$ is a neural network with parameters ϕ that takes in data x and random noise $\epsilon \sim \pi(\epsilon)$ and outputs samples $z \sim q_\phi(z|x)$.
- Typically $\pi(\epsilon) = \mathcal{N}(0, I_{n\times n})$.



- Minimize (reverse) KL Divergence between the two distributions. Since $p(z|x)$ changes with different x, take expectation with respect to dataset $q^*(x)$:

$$q^*_\phi(z|x) = \underset{q(z|x)\in\mathcal{Q}}{\arg\min}\, \mathbb{E}_{q^*(x)}[KL(q_\phi(z|x)||p(z|x))].$$

- Reverse KL Divergence is the expected logarithmic difference between two distributions P and Q with respect to Q:

$$KL(q(z|x)||p(z|x)) = \mathbb{E}_{q(z|x)}\left[\log\left(\frac{q(z|x)}{p(z|x)}\right)\right]$$

.

(Amortized) Variational Inference

Network Training

- We don't know $p(z|x)$ so we apply Bayes' law to $p(z|x)$ and move out intractable $\log p(x)$ term.

$$\mathbb{E}_{q^*(x)}[KL(q_\phi(z|x)||p(z|x))]$$
$$= \mathbb{E}_{q^*(x)q_\phi(z|x)}[\log q(z) - \log p(x|z) - \log p(z) + \log p(x)]$$

$$\mathbb{E}_{q^*(x)}[KL(q_\phi(z|x)||p(z|x)) - \log p(x)]$$
$$= -\mathbb{E}_{q^*(x)q(z)}[\log p(x|z)] + \mathbb{E}_{q^*(x)} KL[q_\phi(z|x)||p(z)]$$

- Denote RHS as $NELBO(q)$, the **n**egative of the **e**vidence **l**ower **bo**und:

$$\min_\phi NELBO(q) = -\mathbb{E}_{q_\phi(z|x)q^*(x)}[\log p(x|z)] + \mathbb{E}_{q^*(x)}[KL(q_\phi(z|x)||p(z))].$$

(Amortized) Variational Inference

Prior-Contrastive

$$\min_{\phi} -\mathbb{E}_{q_\phi(z|x)q^*(x)}[\log p(x|z)] + \mathbb{E}_{q^*(x)}[KL(q_\phi(z|x)||p(z))].$$

- $q_\phi(z|x)$ is a neural network so extremely difficult to find explicit form, we therefore say that it is **implicit**.
- Use density ratio estimation to evaluate $\frac{q_\phi(z|x)}{p(z)}$ in $KL(q_\phi(z|x)||p(z))$.
- The prior $p(z)$ can therefore be implicit.
- We call this the "prior-contrastive" formulation.

(Amortized) Variational Inference

Joint-Contrastive

- If the likelihood $p(x|z)$ is implicit, then our optimization problem is

$$\min_{\phi} KL(q(z,x)||p(z,x)).$$

- Use density ratio estimation to evaluate $\frac{q(z,x)}{p(z,x)}$.
- For consistency, $NELBO(q) = \min_{\phi} KL(q(z,x)||p(z,x))$.
- We call this the "joint-contrastive" formulation.

Outline

1. Background Info
   - Neural Networks
   - (Amortized) Variational Inference
   - Density Ratio Estimation
2. Activation Function Experiment
3. Theory Break
4. Experiments
   - Inference Experiment
   - Generation Experiment
5. Further Estimator Loss Function Analysis

Density Ratio Estimation

Class Probability Estimation

We want to estimate $\frac{q(u)}{p(u)}$.

1. Define discriminator function that finds probability that a sample u came from $q(u)$: $D_\alpha(u) \simeq P(u \sim q(u))$, so that $\frac{q(u)}{p(u)} \simeq \frac{D_\alpha(u)}{1-D_\alpha(u)}$.
2. $D_\alpha(u)$ is neural network parametrised by α, sigmoid activation function used for output layer
3. Train discriminator with Bernoulli loss: $\min_\alpha -\mathbb{E}_{q(u)}[\log D_\alpha(u)] - \mathbb{E}_{p(u)}[\log(1 - D_\alpha(u))]$.
4. Optimal discriminator is $D^*_\alpha(u) = \frac{q(u)}{q(u)+p(u)}$.

Prior-Contrastive Application:

$$\min_{\alpha} -\mathbb{E}_{q^*(x)q_\phi(z|x)}[\log D_\alpha(z,x)] - \mathbb{E}_{q^*(x)p_\theta(z)}[\log(1 - D_\alpha(z,x))]$$

$$\min_{\phi} -\mathbb{E}_{q^*(x)q_\phi(z|x)}[\log p(x|z)] + \mathbb{E}_{q^*(x)q_\phi(z|x)}\left[\log \frac{D_\alpha(z,x)}{1 - D_\alpha(z,x)}\right]$$

Joint-Contrastive Application:

$$\min_{\alpha} -\mathbb{E}_{q^*(x)q_\phi(z|x)}[\log D_\alpha(z,x)] - \mathbb{E}_{p(z)p(x|z)}[\log(1 - D_\alpha(z,x))]$$

$$\min_{\phi} \mathbb{E}_{q^*(x)q_\phi(z|x)} \log \frac{D_\alpha(z,x)}{1 - D_\alpha(z,x)}$$

Program alternates between several optimisation steps of discriminator and one optimisation step of posterior.

Theorem

If f is a convex function with derivative f' and convex conjugate f^*, and $\mathscr{R}$ is a class of functions with codomains equal to the domain of f', then we have the lower bound for the f-divergence between distributions $p(u)$ and $q(u)$:

$$D_f[p(u)||q(u)] \geq \sup_{r \in \mathscr{R}} \{\mathbb{E}_{q(u)}[f'(r(u))] - \mathbb{E}_{p(u)}[f^*(f'(r(u)))]\},$$

with equality when $r(u) = q(u)/p(u)$.

For the reverse KL divergence, $f(u) = u \log u$ so we have

$$KL[q(u)||p(u)] \geq \sup_{r \in \mathscr{R}} \{\mathbb{E}_{q(u)}[1 + \log r(u)] - \mathbb{E}_{p(u)}[r(u)]\}$$

- Let our ratio estimator be a neural network parametrised by α: $r_\alpha(u) \simeq \frac{q(u)}{p(u)}$.
- Maximise the lower bound w.r.t. α until equality, which is when $r_\alpha(u) = \frac{q(u)}{p(u)}$. The optimisation problem for this is

$$\min_{\alpha} -\mathbb{E}_{q(u)}[\log r_\alpha(u)] + \mathbb{E}_{p(u)}[r_\alpha(u)].$$

- Obviously our optimal ratio estimator is $r_\alpha^*(u) = \frac{q(u)}{p(u)}$.

Prior-Contrastive Application:

$$\min_{\alpha} -\mathbb{E}_{q^*(x)q_\phi(z|x)}[\log r_\alpha(z,x)] + \mathbb{E}_{q^*(x)p(z)}[r_\alpha(z,x)]$$

$$\min_{\phi} -\mathbb{E}_{q^*(x)q_\phi(z|x)}\left[\log p(x|z)\right] + E_{q^*(x)q_\phi(z|x)}[\log r_\alpha(z,x)]$$

Joint-Contrastive Application:

$$\min_{\alpha} -\mathbb{E}_{q^*(x)q_\phi(z|x)}[\log r_\alpha(z,x)] + \mathbb{E}_{p(z)p(x|z)}[r_\alpha(z,x)]$$

$$\min_{\phi} \mathbb{E}_{q^*(x)q_\phi(z|x)}[\log r_\alpha(z,x)]$$

$$p(z_1, z_2) \sim \mathcal{N}(0, \sigma^2 I_{2\times 2})$$
$$p(x|\boldsymbol{z}) \sim EXP(3 + \max(0, z_1)^3 + \max(0, z_2)^3)$$

- Posterior is flexible and bimodal.
- Use Gaussian KDE to find 'true' KL divergence for $q_\phi(z|x = 0, 5, 8, 12, 50)$.

Activation Function Experiment

Failures

- Divergence Minimisation regularly experienced 'failures'
- Estimator loss initialised at 41.4465 and remained constant over optimisation steps.
- Analysis of estimator output showed that it was outputting negative number which was mapped to 0 by ReLU.
- Recall ratio estimator loss of $-\mathbb{E}_q[\log r_\alpha(z,x) + \mathbb{E}_p[r_\alpha(z,x)]$.
- We added constant term of $c = 10^{-18}$ to log input.
- $-\log 10^{-18} = 41.4465$
- Partial derivative of loss function w.r.t weights is 0 as changing weight values slightly still results in negative output before ReLU.

Activation Function Experiment

Problems with ReLU

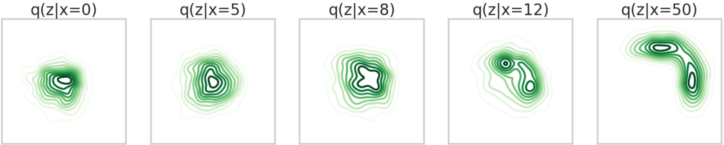
- 'Failures' caused from ReLU outputting in $[0, \infty)$ despite $\frac{q(u)}{p(u)} \in (0, \infty)$.
- If $q(u) < p(u)$, $\frac{q(u)}{p(u)} \in (0, 1)$, and if $q(u) > p(u)$, $\frac{q(u)}{p(u)} \in (1, \infty)$.
- Linearity of ReLU activation results in inconsistent training, as small training steps should be taken if $q(u) < p(u)$, but large training steps required for $q(u) > p(u)$.

Activation Function Experiment

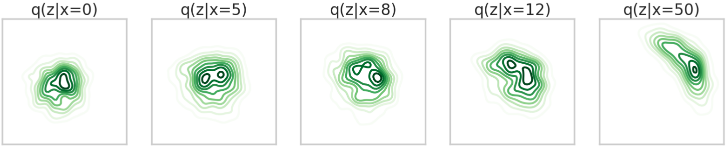
Parameters

- First contribution of thesis: we propose exponential activation function $g(x) = e^x$ for ratio estimator.
- This maps $\mathbb{R}^-$ to $(0, 1)$, and $\mathbb{R}^+$ to $(1, \infty)$.
- Training is consistent and neural network cannot output 0.
- Compare ReLU vs exp activation function for divergence minimisation.
- Low training rate, high iterations to ensure smooth convergence.

Algorithm	Mean KL Divergence	Standard Deviation
PC Divergence Minimisation - ReLU	1.3807	0.0391
PC Divergence Minimisation - Exp	1.3265	0.0045
JC Divergence Minimisation - ReLU	1.6954	0.4337
JC Divergence Minimisation - Exp	1.3397	0.0066



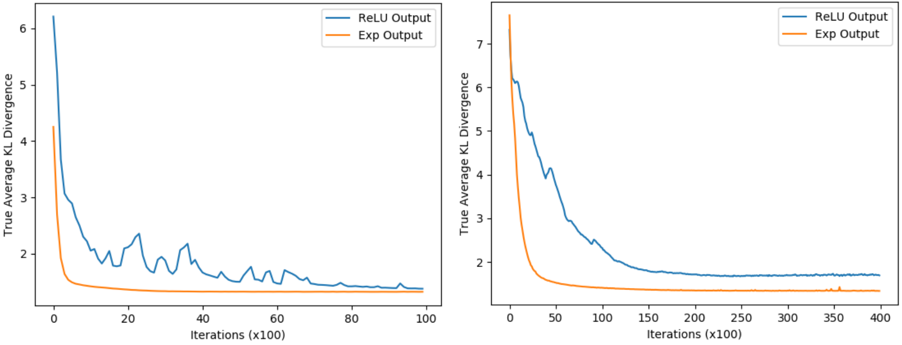
(a) Average KL Divergence of 1.3288



(b) Average KL Divergence of 1.3963

Inference Experiment - Activation Function

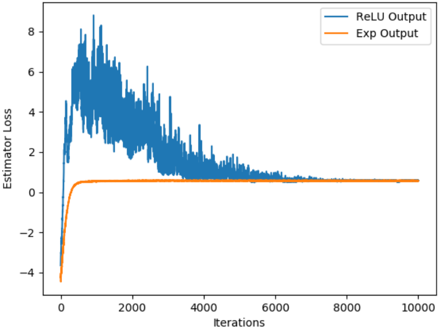
KL Divergence Plots



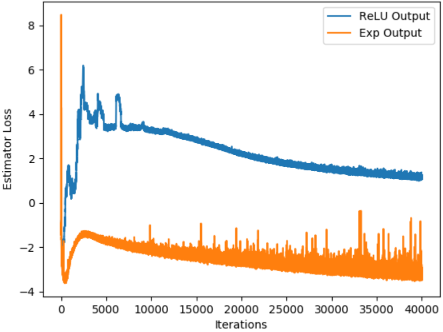
(a) Prior-Contrastive (b) Joint-Contrastive

Inference Experiment - Activation Function

Estimator Losses



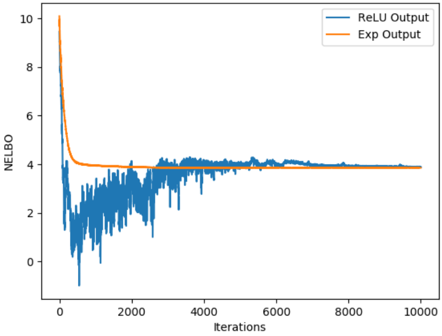
(a) Prior-Contrastive



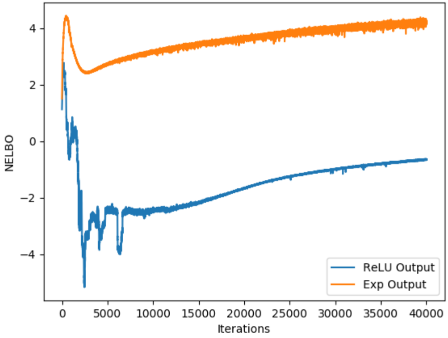
(b) Joint-Contrastive

Inference Experiment - Activation Function

NELBOs



(a) Prior-Contrastive



(b) Joint-Contrastive

Theory Break

Alternative Derivation of Class Probability Estimation

- Recall theorem behind divergence minimisation:

$$D_f[p(u)||q(u)] \geq \sup_{r \in \mathscr{R}} \{\mathbb{E}_{q(u)}[f'(r(u))] - \mathbb{E}_{p(u)}[f^*(f'(r(u)))]\},$$

- If we let $f(u) = u \log u - (u+1)\log(u+1)$ and $D(u) = \frac{r(u)}{r(u)+1}$, we have the lower bound

$$2JS[p(u)||q(u)] - \log 4 \geq \sup_{D} \{\mathbb{E}_{q(u)}[\log D(u)] + \mathbb{E}_{p(u)}[\log(1 - D(u))]\}$$

- This is the same estimator loss as in class probability estimation:

$$\min_{\alpha} -\mathbb{E}_{q(u)}[\log D_\alpha(u)] - \mathbb{E}_{p(u)}[\log(1 - D_\alpha(u))]$$

- We call $2JS[p(u)||q(u)] - \log 4$ the ‘GAN’ divergence

- $D(u) = \frac{r(u)}{r(u)+1}$ is bijective transformation of estimated density ratio.
- Also propose $T(u) = \log r(u)$.
- 2 f-divergences being compared: $KL[q(u)||p(u)]$ and $2JS[p(u)||q(u)] - \log 4$.
- 2 problem contexts (PC, JC)
  $\times$ 2 f-divergences (Reverse KL, GAN)
  $\times$ 3 estimator parametrisations
  $(D_\alpha(u) \simeq \frac{q(u)}{q(u)+p(u)}, r_\alpha(u) \simeq \frac{q(u)}{p(u)}, T_\alpha(u) \simeq \log \frac{q(u)}{p(u)})$
  $=$ 12 experiments.

Outline

Inference Experiment

Comparing Optimal Estimators

- Same inference problem as before.
- Aim of this experiment is to verify that choice of estimator does not matter as long as it reaches equality.
- Low training rate with high estimator to posterior optimisation ratio (100:1).
- High posterior iterations.

Inference Experiment

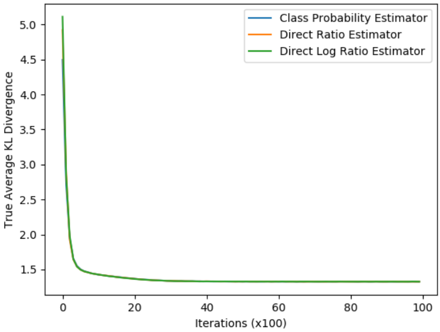
Comparing Optimal Estimators

Algorithm	Mean KL Divergence	Standard Deviation
PC Reverse KL - $D_\alpha(z,x)$	1.3271	0.0041
PC Reverse KL - $r_\alpha(z,x)$	1.3265	0.0045
PC Reverse KL - $T_\alpha(z,x)$	1.3262	0.0041
PC CPE - $D_\alpha(z,x)$	1.3267	0.0041
PC GAN - $r_\alpha(z,x)$	1.3263	0.0035
PC GAN - $T_\alpha(z,x)$	1.3258	0.0039
JC Reverse KL - $D_\alpha(z,x)$	1.3416	0.0068
JC Reverse KL - $r_\alpha(z,x)$	1.3397	0.0066
JC Reverse KL - $T_\alpha(z,x)$	1.3446	0.0108
JC GAN - $D_\alpha(z,x)$	1.3648	0.0242
JC GAN - $r_\alpha(z,x)$	1.3657	0.0302
JC GAN - $T_\alpha(z,x)$	1.3670	0.0387

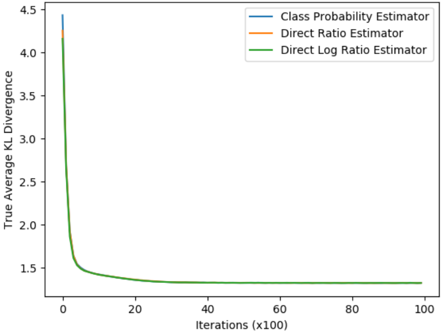
- PC posteriors fully converged, reverse KL converged faster for JC.

Inference Experiment

Prior-Contrastive KL Divergence Plots



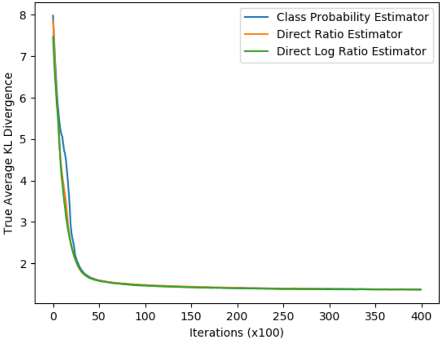
(a) GAN Divergence



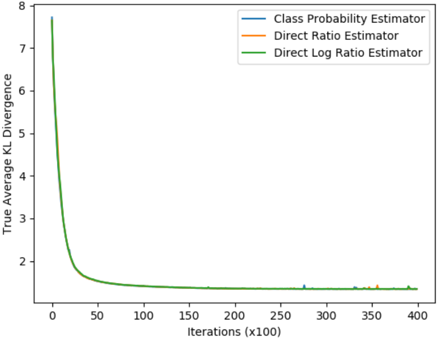
(b) Reverse KL Divergence

Inference Experiment

Joint-Contrastive KL Divergence Plots



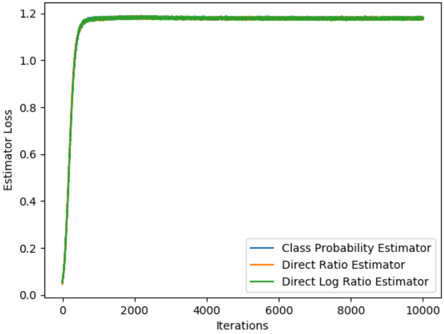
(a) GAN Divergence



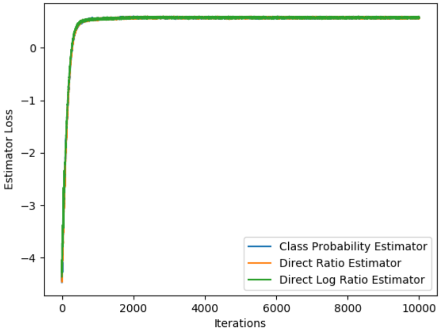
(b) Reverse KL Divergence

Inference Experiment

Prior-Contrastive Estimator Loss Plots



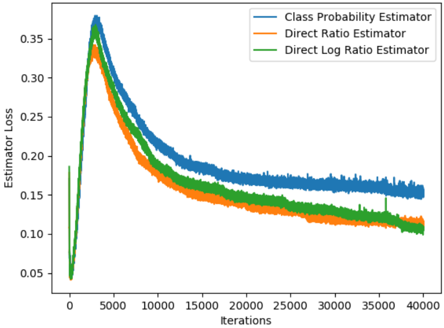
(a) GAN Divergence



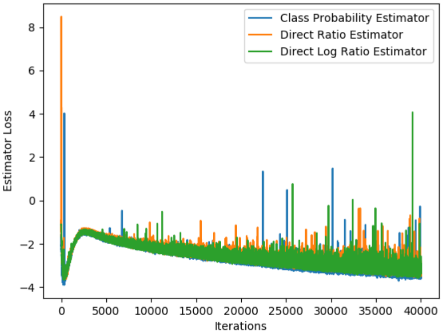
(b) Reverse KL Divergence

Inference Experiment

Joint-Contrastive Estimator Loss Plots



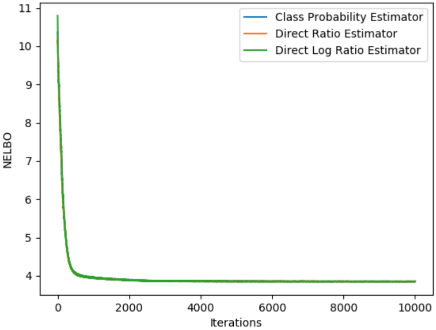
(a) GAN Divergence



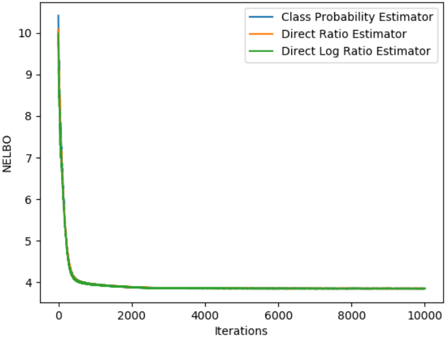
(b) Reverse KL Divergence

Inference Experiment

Prior-Contrastive NELBO Plots



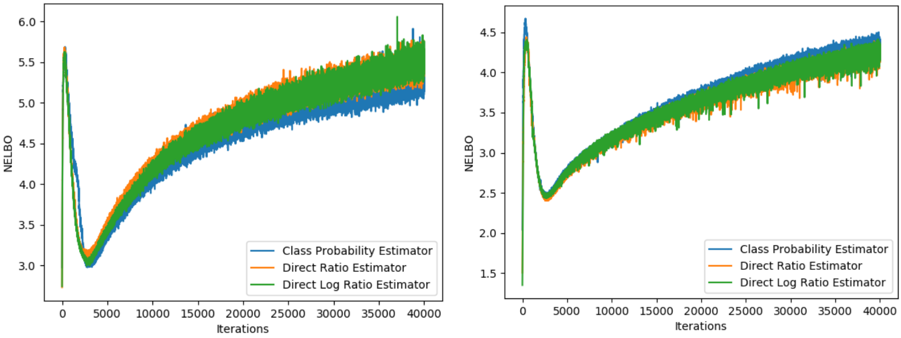
(a) GAN Divergence



(b) Reverse KL Divergence

Inference Experiment

Joint-Contrastive NELBO Plots



(a) GAN Divergence

(b) Reverse KL Divergence

Inference Experiment
Comparing Undertrained Estimators

- Aim of this experiment is to significantly reduce the amount of training the estimator undergoes between each NELBO estimation.
- The combination of f-divergence and estimator parametrisation that trains the fastest will have the highest accuracy, corresponding to the highest posterior convergence.
- Estimator training rate changed to 0.00004 and posterior training rate increased to 0.0002.
- 5000 estimator initialisation steps retained.
- Estimator to posterior iteration ratio reduced to 15:1 in prior-contrastive and 20:1 in joint-contrastive.
- Total posterior iterations reduced to 2000 in prior-contrastive and 4000 in joint-contrastive.

Inference Experiment

Comparing Undertrained Estimators

Algorithm	Mean KL Divergence	Standard Deviation
PC Reverse KL - $D_\alpha(z,x)$	1.3572	0.0136
PC Reverse KL - $r_\alpha(z,x)$	1.3607	0.0199
PC Reverse KL - $T_\alpha(z,x)$	1.3641	0.0141
PC GAN - $D_\alpha(z,x)$	1.3788	0.0258
PC GAN - $r_\alpha(z,x)$	1.3811	0.0365
PC GAN - $T_\alpha(z,x)$	1.3849	0.0450
JC Reverse KL - $D_\alpha(z,x)$	1.3786	0.0286
JC Reverse KL - $r_\alpha(z,x)$	1.3934	0.0410
JC Reverse KL - $T_\alpha(z,x)$	1.4133	0.0597
JC GAN - $D_\alpha(z,x)$	1.4017	0.0286
JC GAN - $r_\alpha(z,x)$	1.4086	0.0555
JC GAN - $T_\alpha(z,x)$	1.4214	0.0518

- Reverse KL divergence significantly better than GAN divergence.

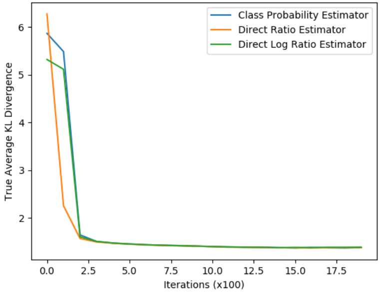
Inference Experiment

Comparing Undertrained Estimators

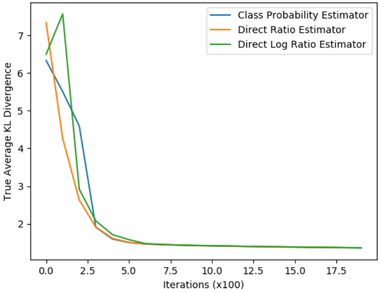
- For PC, $D_\alpha(z,x) < r_\alpha(z,x) < T_\alpha(z,x)$ in terms of mean KL divergence but not by a significant amount.
- Significant in JC. Likely because likelihood term is a factor in PC but JC is entirely based on density ratio.
- Standard deviation of class probability estimator consistently better than other two estimator parametrisations.
- f-divergence used is more significant than estimator parametrisation.
- Optimal combination is reverse KL divergence with class probability estimator.

Inference Experiment

Prior-Contrastive KL Divergence Plots



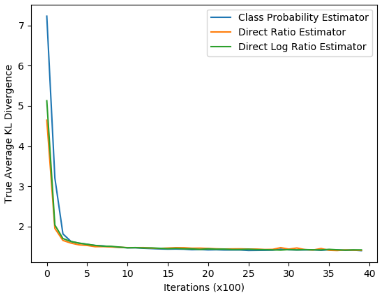
(a) GAN Divergence



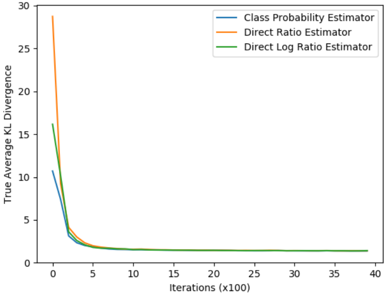
(b) Reverse KL Divergence

Inference Experiment

Joint-Contrastive KL Divergence Plots

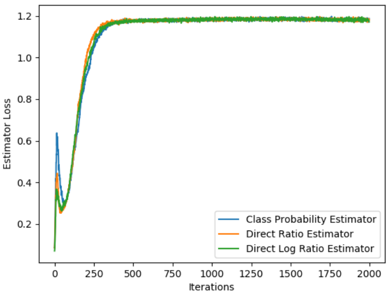


(a) GAN Divergence

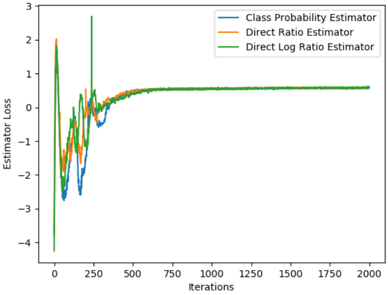


(b) Reverse KL Divergence

Inference Experiment
Prior-Contrastive Estimator Loss Plots



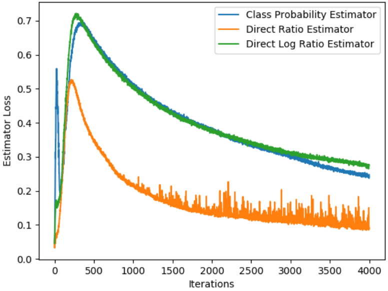
(a) GAN Divergence



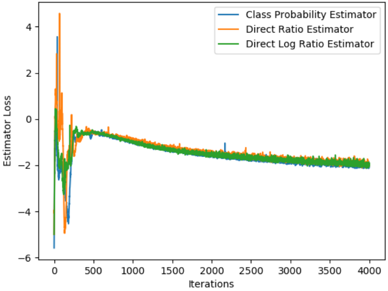
(b) Reverse KL Divergence

Inference Experiment

Joint-Contrastive Estimator Loss Plots



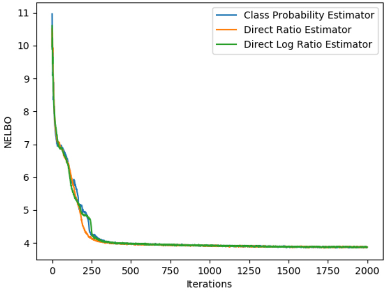
(a) GAN Divergence



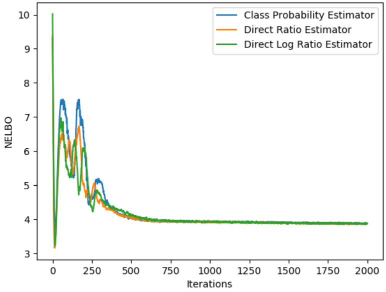
(b) Reverse KL Divergence

Inference Experiment

Prior-Contrastive NELBO Plots



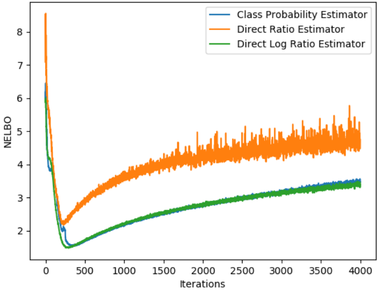
(a) GAN Divergence



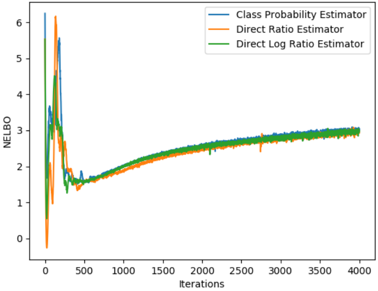
(b) Reverse KL Divergence

Inference Experiment

Joint-Contrastive NELBO Plots



(a) GAN Divergence



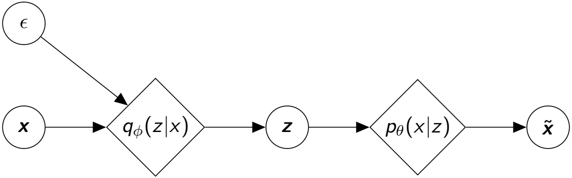
(b) Reverse KL Divergence

Outline

Generation Experiment

Autoencoders

- Likelihood $p_\theta(x|z)$ is now a neural network.
- Posterior $q_\phi(z|x)$ represents data x as lower dimensional latent z.
- Likelihood $p_\theta(x|z)$ reconstructs data $\tilde{x}$ from z.
- Generate new data $\tilde{x}$ using z from $p(z)$.



$$\min_{\theta,\phi} -\mathbb{E}_{q_\phi(z|x)q^*(x)}[\log p_\theta(x|z)] + \mathbb{E}_{q^*(x)}[KL(q_\phi(z|x)||p(z))]$$

- MNIST dataset - 28×28 grey-scale images of handwritten digits
- Not doing joint-contrastive cause unintuitive to 'pretend' we don't know likelihood function.
- Again use low estimator to posterior training ratio.
- Use reconstruction error $\|x - \tilde{x}\|^2$ as metric.
- Perform experiment with low dimensional latent space (2 dimensions) and high dimensional latent space (20 dimensions).

Generation Experiment

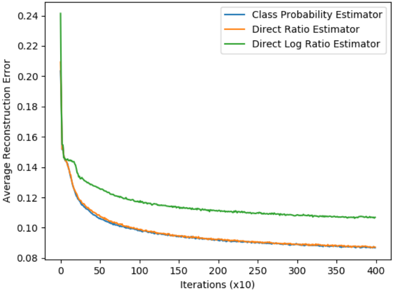
Results - low dimensional latent space

Algorithm	Mean Reconstruction Error	Standard Deviation
PC Reverse KL - $D_\alpha(z, x)$	0.0866	0.0015
PC Reverse KL - $r_\alpha(z, x)$	0.0871	0.0021
PC Reverse KL - $T_\alpha(z, x)$	0.0873	0.0016
PC GAN - $D_\alpha(z, x)$	0.0867	0.0013
PC GAN - $r_\alpha(z, x)$	0.0872	0.0015
PC GAN - $T_\alpha(z, x)$	0.1068	0.0020

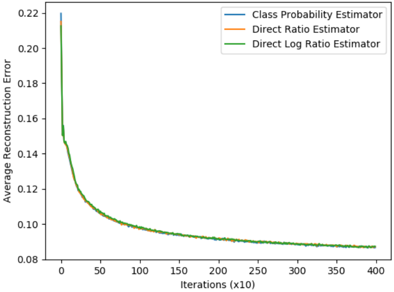
- Mostly insignificant but consistent results.
- Log ratio estimator for GAN is significantly worse.

Generation Experiment

Reconstruction Errors



(a) GAN Divergence



(b) Reverse KL Divergence

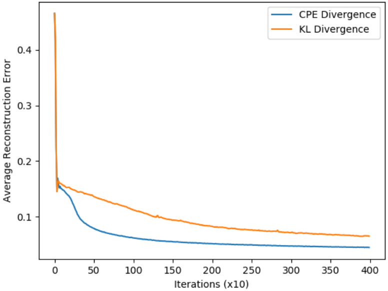
Generation Experiment

Results - high dimensional latent space

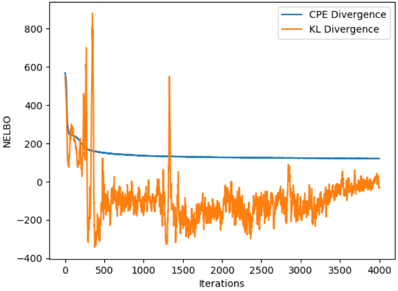
- Direct ratio and direct log ratio estimators attempted to store numbers exceeding float64(max).
- Exponential of $T_\alpha(z, x)$ taken in loss function.
- $D_\alpha(z, x)$ ranges in $(0, 1)$.
- Value before sigmoid activation function for $D_\alpha(z, x)$ is log density ratio.
- Class probability estimator is the best.

Generation Experiment

Results - high dimensional latent space



(a) Reconstruction Error



(b) NELBO

Algorithm	Mean Reconstruction Error	Standard Deviation
PC Reverse KL - $D_\alpha(z, x)$	0.0444	0.0017
PC GAN - $D_\alpha(z, x)$	0.0647	0.0019

- Each estimator loss function is a convex functional that reaches its minimum when estimator is optimal.
- Estimator parametrisation affects its output space and gradient of loss function with respect to the estimator.
- Choice of f-divergence affects gradient of loss function with respect to estimator.

Further Estimator Loss Function Analysis

Estimator Parametrisation

Need some plots xD

- Higher second derivative corresponds to faster convergence.
- Taking second functional derivative of estimator loss function with respect to estimator, we can only make one certain comparison: for the GAN divergence, the class probability estimator has a strictly higher second derivative than the direct ratio estimator.
- The density ratio changes every time the posterior is optimised, and the estimator must catch up. It can be shown that the class probability estimator has a strictly lower displacement than the direct ratio estimator, that is, $|D^*_{final} - D^*_{init}| < |r^*_{final} - r^*_{init}|$.

- Again observing the second functional derivatives, we can only observe that in the direct ratio estimator parametrisation, the reverse KL divergence is strictly higher than the GAN divergence.
- Nowozin's f-GAN paper also shows empirically that the reverse KL divergence is superior when it is additionally used to optimize the posterior.

Summary

- The first main message of your talk in one or two lines.
- The second main message of your talk in one or two lines.
- Perhaps a third message, but not more than that.

- Outlook
  - Something you haven't solved.
  - Something else you haven't solved.

For Further Reading I

A. Author.
Handbook of Everything.
Some Press, 1990.

S. Someone.
On this and that.
Journal of This and That, 2(1):50–100, 2000.