Introduction to Differentiable Probabilistic Models

Bill Watson

S&P Global

July 25, 2019

Primer: Standard Machine Learning

- Usually, we are given a set $\mathcal{D} = \{X, y\}$

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1m} \\ x_{21} & x_{22} & \cdots & x_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nm} \end{bmatrix} \qquad y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

where X is our data matrix, and y are our labels.

Primer: Standard Machine Learning

- Usually, we are given a set $\mathcal{D} = \{X, y\}$

$$X = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1m} \\ x_{21} & x_{22} & \dots & x_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{nm} \end{bmatrix} \qquad y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

  where X is our data matrix, and y are our labels.

- Attempt to fit a model f parameterized by θ with respect to an objective function $\mathcal{L}$

$$\theta^* = \underset{\theta}{\operatorname{argmin}}\ \mathcal{L}\big(f(X;\theta),\ y\big)$$

Reframed as a Probabilistic Model

- Consider our model is a probability distribution Q
  - No longer have labels y
  - But have probabilities and sampling

Reframed as a Probabilistic Model

- Consider our model is a probability distribution Q
  - No longer have labels y
  - But have probabilities and sampling
- Consider that our data is sampled from the real world P:

$$X \sim P$$
$$\theta^* = \underset{\theta}{\text{argmin}} \ \mathcal{L}\big(f(X;\theta)\big)$$

Reframed as a Probabilistic Model

- Consider our model is a probability distribution Q
  - No longer have labels y
  - But have probabilities and sampling
- Consider that our data is sampled from the real world P:

$$X \sim P$$

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \ \mathcal{L}\big(f(X;\theta)\big)$$

- Examples:
  - Classification: Fitting two multinomial distributions
  - Regression: Fitting a Normal centered around the line of best fit

How do we "fit" Distributions?

- Fitting two distributions implies minimizing their difference, i.e. "distance"
- This "distance" is known as the divergence between the true distribution P and the learned distribution Q.

How do we "fit" Distributions?

- Fitting two distributions implies minimizing their difference, i.e. "distance"
- This "distance" is known as the divergence between the true distribution P and the learned distribution Q.
- Divergences must satisfy 2 properties:
  - $D(P \parallel Q) \geq 0 \quad \forall P, Q \in S$
  - $D(P \parallel Q) = 0 \iff P = Q$

The Kullback-Leibler Divergence

- The KL Divergence for distributions P and Q is defined as:

$$D_{KL}(P \parallel Q) = \int_{-\infty}^{\infty} p(x) \log \left(\frac{p(x)}{q(x)} \right) \, dx$$

The Kullback-Leibler Divergence

- The KL Divergence for distributions P and Q is defined as:

$$D_{KL}(P \parallel Q) = \int_{-\infty}^{\infty} p(x) \log \left(\frac{p(x)}{q(x)} \right) \, dx$$

- Note: the KL Divergence is NOT symmetric:

$$D_{KL}(P \parallel Q) \neq D_{KL}(Q \parallel P)$$

- Hence, this direction is known as the Forward KL

The Kullback-Leibler Divergence

- The KL Divergence for distributions P and Q is defined as:

$$D_{KL}(P \parallel Q) = \int_{-\infty}^{\infty} p(x) \log \left(\frac{p(x)}{q(x)} \right) \, dx$$

- Note: the KL Divergence is NOT symmetric:

$$D_{KL}(P \parallel Q) \neq D_{KL}(Q \parallel P)$$

- Hence, this direction is known as the Forward KL
- But we will come back to this later...

Digression: Monte Carlo Integration

$$D_{KL}(P \parallel Q) = \int_{-\infty}^{\infty} p(x) \log \left(\frac{p(x)}{q(x)} \right) \, dx$$

Digression: Monte Carlo Integration

$$
\begin{aligned}
D_{KL}(P \parallel Q) &= \int_{-\infty}^{\infty} p(x) \log\left(\frac{p(x)}{q(x)}\right)\, dx \\
&= \mathbb{E}_{x \sim P}\left[\log\left(\frac{p(x)}{q(x)}\right)\right]
\end{aligned}
$$

Digression: Monte Carlo Integration

$$
\begin{aligned}
D_{KL}(P \parallel Q) &= \int_{-\infty}^{\infty} p(x) \log\left(\frac{p(x)}{q(x)}\right) dx \\
&= \mathbb{E}_{x \sim P}\left[\log\left(\frac{p(x)}{q(x)}\right)\right] \\
&\approx \frac{1}{N} \sum_{i=1}^{N} \log\left(\frac{p(x_i)}{q(x_i)}\right) \quad x_i \sim P
\end{aligned}
$$

Digression: Monte Carlo Integration

$$
\begin{aligned}
D_{KL}(P \parallel Q) &= \int_{-\infty}^{\infty} p(x) \log\left(\frac{p(x)}{q(x)}\right)\, dx \\
&= \mathbb{E}_{x\sim P}\left[\log\left(\frac{p(x)}{q(x)}\right)\right] \\
&\approx \frac{1}{N}\sum_{i=1}^{N} \log\left(\frac{p(x_i)}{q(x_i)}\right) \quad x_i \sim P
\end{aligned}
$$

Algorithm 1 $\mathbb{E}_{x\sim P}[f(x)]$
Expectation of $f(x)$ with respect to P

1: $x_1, \ldots, x_n \sim P$ independently
2: **return** $\frac{1}{N}\sum_{x_i} f(x_i)$

Digression: KL to Cross-Entropy

- Note: Maximum Likelihood Estimation is equivalent to minimizing the Forward KL

Digression: KL to Cross-Entropy

- Note: Maximum Likelihood Estimation is equivalent to minimizing the Forward KL
- The KL Divergence can be decomposed into familiar terms:

$$\underset{Q}{\operatorname{argmin}}\, D_{KL}(P \parallel Q) = \underset{Q}{\operatorname{argmin}} \sum_{x \in \mathcal{X}} p(x) \log\left(\frac{p(x)}{q(x)}\right)$$

Digression: KL to Cross-Entropy

- Note: Maximum Likelihood Estimation is equivalent to minimizing the Forward KL
- The KL Divergence can be decomposed into familiar terms:

$$
\begin{aligned}
\operatorname*{argmin}_Q D_{KL}(P \parallel Q) &= \operatorname*{argmin}_Q \sum_{x \in \mathcal{X}} p(x) \log \left(\frac{p(x)}{q(x)} \right) \\
&= \operatorname*{argmin}_Q - \sum_{x \in \mathcal{X}} p(x) \log q(x) \\
&\qquad + \sum_{x \in \mathcal{X}} p(x) \log p(x)
\end{aligned}
$$

Digression: KL to Cross-Entropy

- Note: Maximum Likelihood Estimation is equivalent to minimizing the Forward KL
- The KL Divergence can be decomposed into familiar terms:

$$
\begin{aligned}
\operatorname*{argmin}_{Q} D_{KL}(P \parallel Q) &= \operatorname*{argmin}_{Q} \sum_{x \in \mathcal{X}} p(x) \log \left(\frac{p(x)}{q(x)} \right) \\
&= \operatorname*{argmin}_{Q} - \sum_{x \in \mathcal{X}} p(x) \log q(x) \\
&\qquad + \sum_{x \in \mathcal{X}} p(x) \log p(x) \\
&= \operatorname*{argmin}_{Q} \underbrace{H(P, Q)}_{Cross-Entropy} - \underbrace{H(P)}_{Entropy}
\end{aligned}
$$

Digression: KL to Cross-Entropy

- Note: Maximum Likelihood Estimation is equivalent to minimizing the Forward KL
- The KL Divergence can be decomposed into familiar terms:

$$\begin{aligned}
\operatorname*{argmin}_Q D_{KL}(P \parallel Q) &= \operatorname*{argmin}_Q \sum_{x \in \mathcal{X}} p(x) \log\left(\frac{p(x)}{q(x)}\right) \\
&= \operatorname*{argmin}_Q - \sum_{x \in \mathcal{X}} p(x) \log q(x) \\
&\qquad + \sum_{x \in \mathcal{X}} p(x) \log p(x) \\
&= \operatorname*{argmin}_Q \underbrace{H(P, Q)}_{Cross-Entropy} - \underbrace{H(P)}_{Entropy} \\
&= \operatorname*{argmin}_Q \underbrace{H(P, Q)}_{Cross-Entropy}
\end{aligned}$$

Digression: KL to Cross-Entropy

$$H(P, Q) = -\sum_{x \in \mathcal{X}} p(x) \log q(x)$$

- If we consider $P(y_i = 1|x_i) = p_i$ and $Q(y_i = 1|x_i) = \sigma(f_\theta(x_i))$:

$$\operatorname*{argmin}_\theta D_{KL}(P \parallel Q) =$$
$$\operatorname*{argmin}_\theta -\Big[p_i \log \sigma\big(f_\theta(x_i)\big) + (1 - p_i) \log(1 - \sigma\big(f_\theta(x_i)\big))\Big]$$

- This is the Binary Cross-Entropy Loss

Forward KL: Learning a Normal Distribution (Initial)

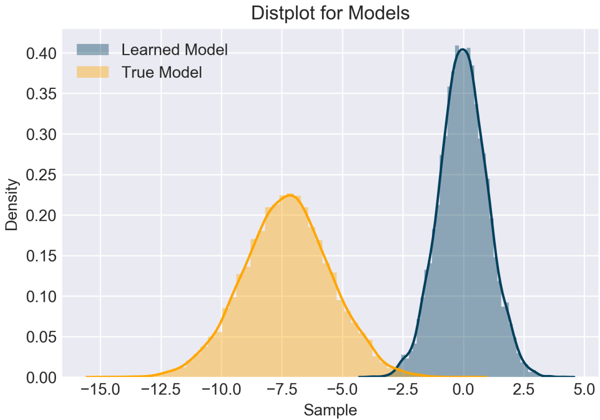
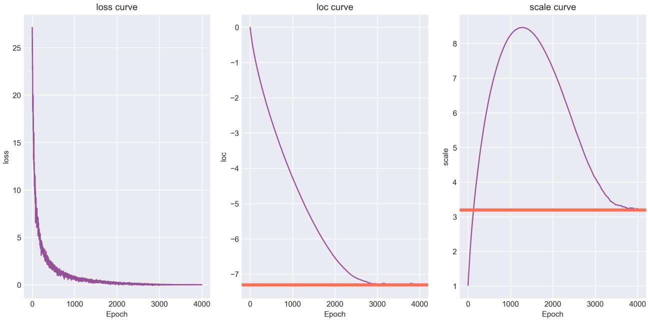


Figure: $P \sim \mathcal{N}(-7.3, 3.2)$, $Q \sim \mathcal{N}(0, 1)$

Forward KL: Learning a Normal Distribution (Results)



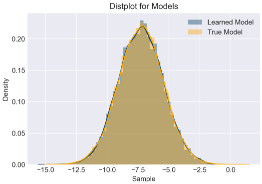


Figure: $P \sim \mathcal{N}(-7.3, 3.2)$, $Q \sim \mathcal{N}(-7.28, 3.24)$

Digression: Gaussian Mixture Models

- We can build a K multi-modal distribution, with weights π, as follows:

$$\begin{aligned} z &\sim \text{Categorical}(\pi) \\ x \mid z = k &\sim \text{Normal}(\mu_k, \sigma_k) \end{aligned}$$

- We can calculate log probabilities by marginalizing out z:

$$\log p(x) = \log \sum_{k=1}^{K} \underbrace{p(z = k)}_{\text{Categorical}} \cdot \underbrace{p(x \mid z = k)}_{\text{Normal}}$$

Digression: Mixture Models (Visual)

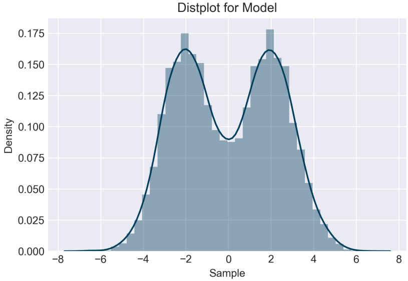


Figure: 2 Mixture Components, Even Weights

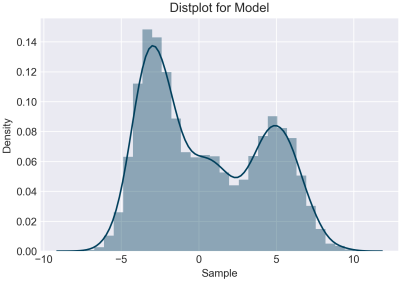


Figure: 3 Mixture Components, Uneven Weights

Forward KL: Learning a Bimodal (Initial)

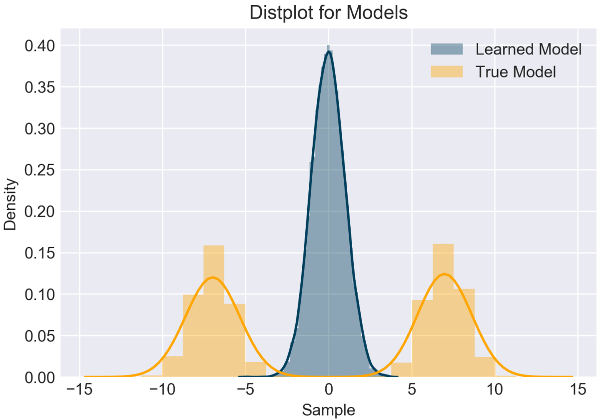


Figure: $P \sim \{\mathcal{N}(-7.3, 1.4),\ \mathcal{N}(7.3, 1.4)\}$
$Q \sim \mathcal{N}(0, 1)$

Forward KL: Learning a Bimodal (Results)

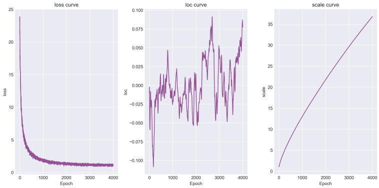


Figure: $Q \sim \mathcal{N}(0.08, 36.76)$

Forward KL: Zero-Avoiding

$$D_{KL}(P \parallel Q) = \int_{-\infty}^{\infty} \overbrace{p(x)}^{\text{Constant}} \log \left(\frac{\overbrace{p(x)}^{\text{Constant}}}{\underbrace{q(x)}_{\textit{Variable}}} \right) dx$$

- $p(x)$ is constant-valued, $q(x)$ is variable
- If Q does not support P, then we will sample a point that has a low probability with respect to Q

Forward KL: Zero-Avoiding

$$D_{KL}(P \parallel Q) = \int_{-\infty}^{\infty} \overbrace{p(x)}^{\text{Constant}} \log \left(\frac{\overbrace{p(x)}^{\text{Constant}}}{\underbrace{q(x)}_{\textit{Variable}}} \right) dx$$

- $p(x)$ is constant-valued, $q(x)$ is variable
- If Q does not support P, then we will sample a point that has a low probability with respect to Q
- As $q(x) \to 0$, our loss $D_{KL} \to \infty$

Forward KL: Zero-Avoiding

$$D_{KL}(P \parallel Q) = \int_{-\infty}^{\infty} \overbrace{p(x)}^{\text{Constant}} \log \left(\frac{\overbrace{p(x)}^{\text{Constant}}}{\underbrace{q(x)}_{\textit{Variable}}} \right) dx$$

- $p(x)$ is constant-valued, $q(x)$ is variable
- If Q does not support P, then we will sample a point that has a low probability with respect to Q
- As $q(x) \to 0$, our loss $D_{KL} \to \infty$
- Hence, the optimal solution is for Q to cover P, i.e. averaging

Forward KL: Loss Landscape

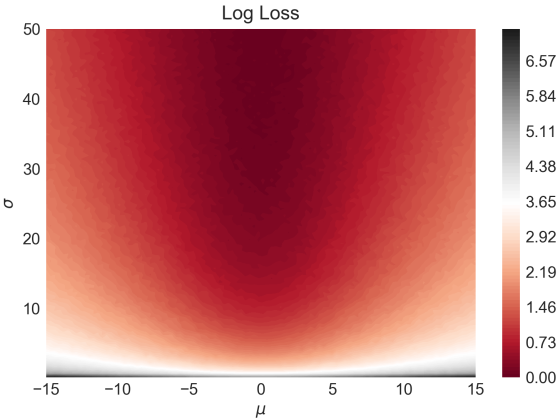


Figure: Loss Landscape for Forward KL Divergence

Directionality: Reverse KL

$$D_{KL}(Q \parallel P) = \int_{-\infty}^{\infty} q(x) \log \left(\frac{q(x)}{p(x)} \right) \, dx$$
$$= \mathbb{E}_{x \sim Q} \left[\log \left(\frac{q(x)}{p(x)} \right) \right]$$

- The Reverse KL will sample from Q, and evaluate the log probabilities from P and Q
- Recall: KL Divergence is not symmetric, and this has drastic implications...

Digression: Differentiable Sampling via the Reparameterization Trick

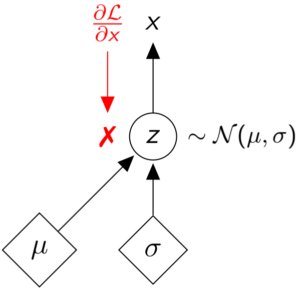


Figure: Original Form

Digression: Differentiable Sampling via the Reparameterization Trick

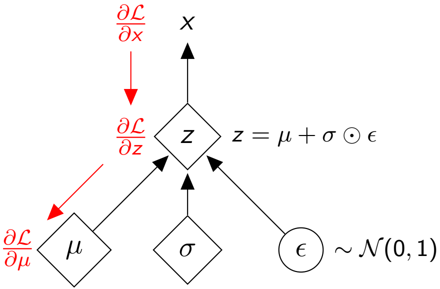


Figure: Reparameterized Version

Digression: Common Reparameterization Tricks

	Reparameterized
$\mathcal{N}(\mu, \sigma)$	$\mu + \sigma \cdot \mathcal{N}(0, 1)$
$\mathrm{Uniform}(a, b)$	$a + (b - a) \cdot \mathrm{Uniform}(0, 1)$
$\mathrm{Exp}(\lambda)$	$\mathrm{Exp}(1)/\lambda$
$\mathrm{Cauchy}(\mu, \gamma)$	$\mu + \gamma \cdot \mathrm{Cauchy}(0, 1)$

Digression: Common Reparameterization Tricks

	Reparameterized
$\mathcal{N}(\mu, \sigma)$	$\mu + \sigma \cdot \mathcal{N}(0, 1)$
$\mathrm{Uniform}(a, b)$	$a + (b - a) \cdot \mathrm{Uniform}(0, 1)$
$\mathrm{Exp}(\lambda)$	$\mathrm{Exp}(1)/\lambda$
$\mathrm{Cauchy}(\mu, \gamma)$	$\mu + \gamma \cdot \mathrm{Cauchy}(0, 1)$
$\mathrm{Laplace}(\mu, b)$	$u \sim \mathrm{Uniform}(-1, 1)$ $\mu - b \cdot \mathrm{sgn}(u) \cdot \ln\left[1 - \lvert u \rvert\right]$

Digression: Common Reparameterization Tricks

	Reparameterized
$\mathcal{N}(\mu, \sigma)$	$\mu + \sigma \cdot \mathcal{N}(0, 1)$
$\text{Uniform}(a, b)$	$a + (b - a) \cdot \text{Uniform}(0, 1)$
$\text{Exp}(\lambda)$	$\text{Exp}(1)/\lambda$
$\text{Cauchy}(\mu, \gamma)$	$\mu + \gamma \cdot \text{Cauchy}(0, 1)$
$\text{Laplace}(\mu, b)$	$u \sim \text{Uniform}(-1, 1)$ $\mu - b \cdot \text{sgn}(u) \cdot \ln\left[1 - \lvert u \rvert\right]$
$\text{Categorical}(\pi)$[1]	✗

[1] Can be approximated with Gumbel Softmax

Reverse KL: Learning a Bimodal (Initial)

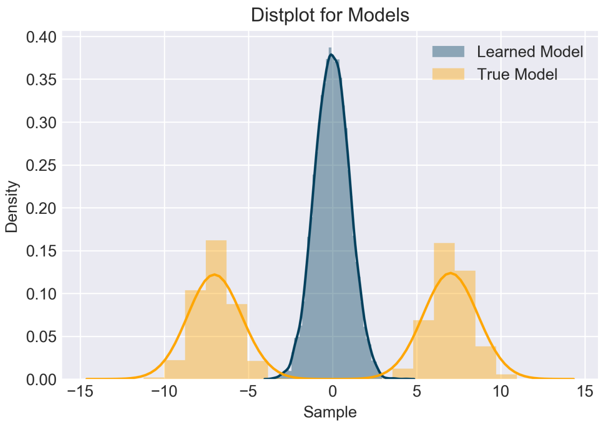
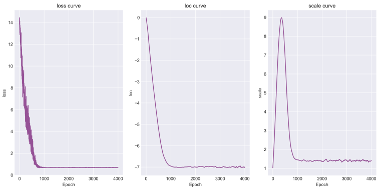


Figure: $P \sim \{\mathcal{N}(-7.3, 1.4),\ \mathcal{N}(7.3, 1.4)\}$
$Q \sim \mathcal{N}(0, 1)$

Reverse KL: Learning a Bimodal (Results)



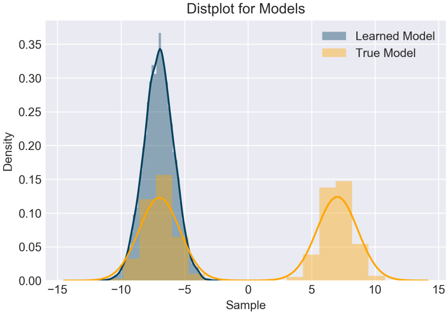


Figure: $Q \sim \mathcal{N}(-7.02, 1.41)$

Reverse KL: Learning a Bimodal Attempt 2 (Results)

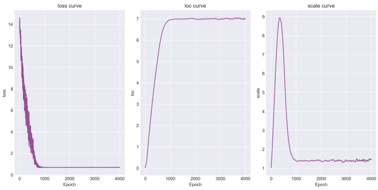


Figure: $Q \sim \mathcal{N}(7.01, 1.46)$

Reverse KL: Zero-Forcing

$$D_{KL}(Q \parallel P) = \int_{-\infty}^{\infty} q(x) \log\left(\frac{q(x)}{p(x)}\right) dx$$

- Unlike the Forward KL, Reverse KL is Zero-forcing
- Why? Because we no longer suffer a penalty from $q(x) = 0$
- However, if $p(x) = 0$, then the optimal value for $q(x)$ is 0
- Result $\implies$ Mode Collapse

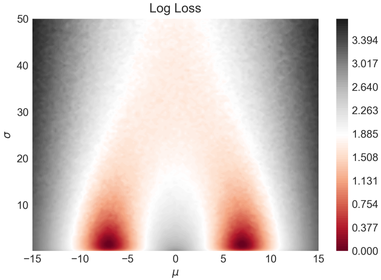


Figure: Loss Landscape for Reverse KL Divergence

Jensen - Shannon Divergence: A Symmetric Divergence

$$
\begin{aligned}
\mathrm{JSD}(P \parallel Q) &= \frac{1}{2} D_{KL}(P \parallel M) + \frac{1}{2} D_{KL}(Q \parallel M) \\
M &= \frac{1}{2}(P + Q)
\end{aligned}
$$

- The JS Divergence is a symmetrized version of the KL Divergence
- M is the average of distributions P and Q, and can be represented as a Mixture Model

Jensen - Shannon Divergence: Bimodal (Initial)

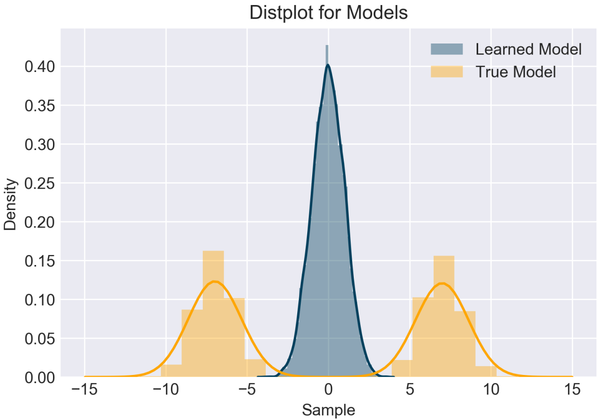


Figure: $P \sim \{\mathcal{N}(-7.3, 1.4),\ \mathcal{N}(7.3, 1.4)\}$
$Q \sim \mathcal{N}(0, 1)$

Jensen - Shannon Divergence: Bimodal (Result)

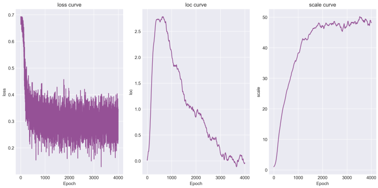


Figure: $Q \sim \mathcal{N}(-0.04, 48.20)$

Jensen - Shannon Divergence: Right Shift (Attempt 2)

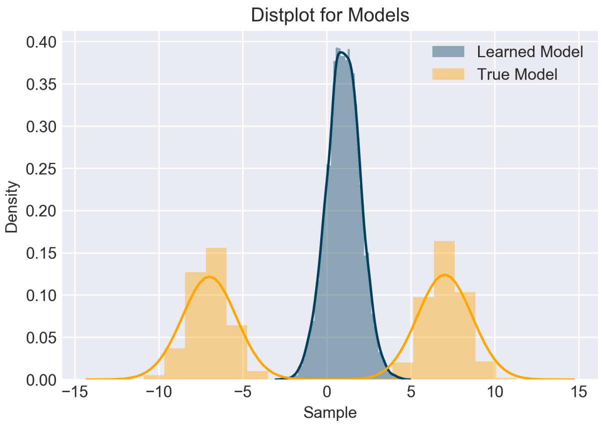
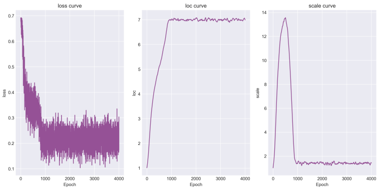


Figure: $P \sim \{\mathcal{N}(-7.3, 1.4),\ \mathcal{N}(7.3, 1.4)\}$
$Q \sim \mathcal{N}(1, 1)$

Jensen - Shannon Divergence: Right Shift (Result)



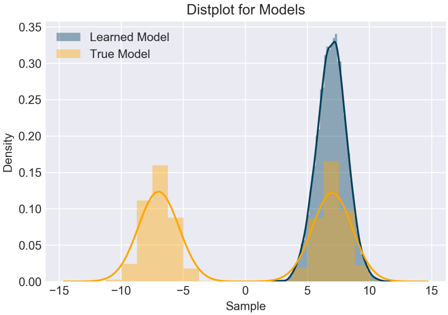


Figure: $Q \sim \mathcal{N}(6.99, 1.43)$

Jensen - Shannon Divergence: Left Shift (Attempt 3)

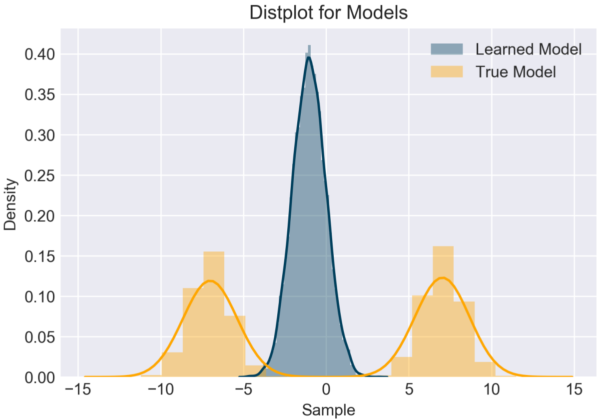
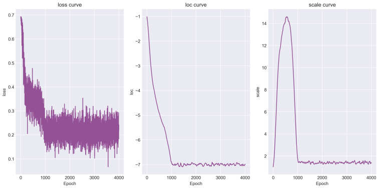


Figure: $P \sim \{\mathcal{N}(-7.3, 1.4),\ \mathcal{N}(7.3, 1.4)\}$
$Q \sim \mathcal{N}(-1, 1)$

Jensen - Shannon Divergence: Left Shift (Result)



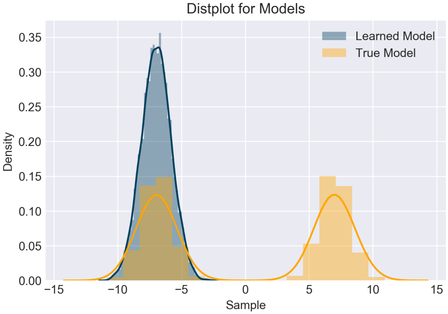


Figure: $Q \sim \mathcal{N}(-6.98, 1.38)$

Jensen - Shannon Divergence Loss

$$
\begin{aligned}
\mathrm{JSD}(P \parallel Q) &= \frac{1}{2} D_{KL}(P \parallel M) + \frac{1}{2} D_{KL}(Q \parallel M) \\
M &= \frac{1}{2}(P + Q)
\end{aligned}
$$

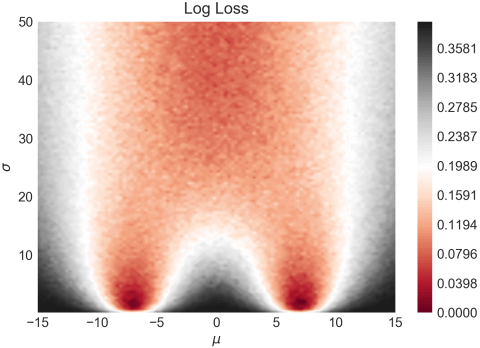


Figure: Loss Landscape for JS Divergence

A Family of Divergences: f-Divergence

- KL Divergence is a special case of the f-divergence
- The f-divergence is a family of divergences that can be written as:

$$D_f(P \parallel Q) = \int \overbrace{q(x)}^{\text{Weight}} f \underbrace{\left(\frac{p(x)}{q(x)}\right)}_{\text{Odds Ratio}} dx$$

A Family of Divergences: f-Divergence

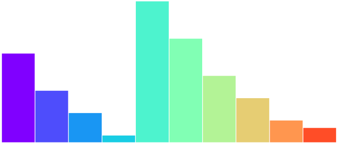
$$D_f(P \parallel Q) = \int \overbrace{q(x)}^{\text{Weight}} f\underbrace{\left(\frac{p(x)}{q(x)}\right)}_{\text{Odds Ratio}} dx$$

Divergence	$f(t)$
Forward KL	$t \log t$
Reverse KL	$-\log t$
Hellinger Distance	$\left(\sqrt{t}-1\right)^2,\ 2\left(1-\sqrt{t}\right)$
Total Variation	$\frac{1}{2}\lvert t-1\rvert$
Pearson χ^2	$(t-1)^2,\ t^2-1,\ t^2-t$
Neyman χ^2 (Reverse Pearson)	$\frac{1}{t}-1,\ \frac{1}{t}-t$

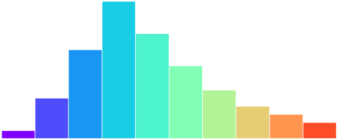
Earth Mover's Distance (Wasserstein Distance)

$$\mathrm{EMD}\left(P, Q\right) = \inf_{\gamma \in \prod} \sum_{p,q} \overbrace{\|p - q\|}^{\ell_2 \text{ Norm}} \underbrace{\gamma\left(p, q\right)}_{\text{Joint Marginal}}$$

- $\gamma\left(p, q\right)$ states how we distribute the amount of "earth" from one place q over the domain of p, or vice versa
- EMD is the minimal total amount of work it takes to transform one distribution into the other



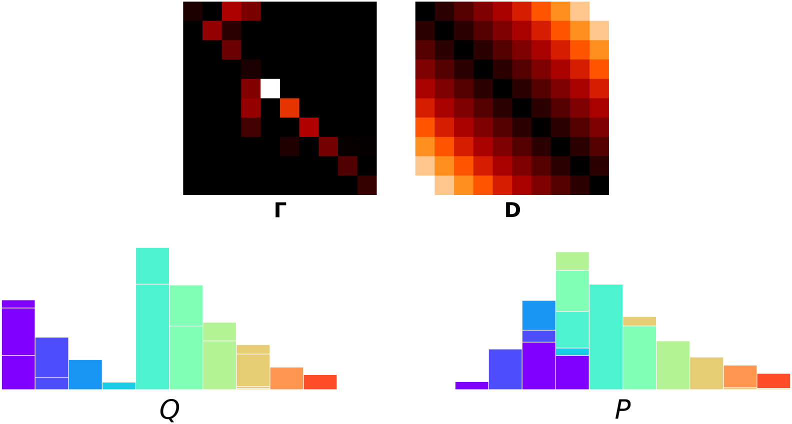
Q



P

Earth Mover's Distance (Wasserstein Distance)

$$\mathrm{EMD}(P, Q) = \inf_{\gamma \in \prod} \sum_{p,q} \|p - q\| \gamma(p, q)$$



Summary of Methods

	P		Q	
	$\log p(x)$	$x \sim P$	$\log q(x)$	$x \sim Q$
Cross-Entropy		✓	✓	
Forward KL	✓	✓	✓	
Reverse KL	✓		✓	✓
JS Divergence	✓	✓	✓	✓
f-Divergence	✓		✓	✓
Wasserstein[2]		✓		✓

[2] Wasserstein GAN

Practical Uses of Divergences

- Forward Kullback-Leibler
  - Maximum Likelihood Estimation
    - ℓ_2: Mean Squared Error (Normally Distributed)
    - ℓ_1: Mean Absolute Error (Laplace Distributed)
    - Binary Cross Entropy (Bernoulli Distributed)
    - Cross Entropy (Multinomially Distributed)
  - Log-Likelihood Models
    - PixelCNN
    - Glow
    - Variational Autoencoders

Practical Uses of Divergences

- Reverse Kullback-Leibler
  - Evidence Lower Bound (ELBO)
- Jensen-Shannon Divergence
  - Generative Adversarial Network (Original)
- Earth Mover's Distance
  - Wasserstein GAN (WGAN)
- Pearson χ^2 Divergence
  - Least Squares GAN (LSGAN)

Potpourri: Advanced Techniques

1. Invertible Transforms
   - Normalizing Flow Models
2. Expectation–Maximization
3. Variational Inference
   - ELBO
4. Adversarial Training (Forest of GANs)
5. Markov Chain Monte Carlo
   - Metropolis-Hastings
   - Gibbs Sampling
   - Hamiltonian Monte Carlo
   - NUTS

Source Code

- Repo for Differentiable Probabilistic Models
- Notebook to generate training examples
- Notebook for EMD
- LaTeX source code for presentation

Further Reading

- Machine Learning: A Probabilistic Perspective by Kevin Murphy
- Friendly Introduction to Cross-Entropy Loss
- Categorical Reparameterization with Gumbel Softmax
- Wasserstein GAN and the Kantorovich-Rubinstein Duality
- Are all GAN's created Equal?
- Tutorial on MCMC Methods