Black-box $\alpha$-divergence Minimization

José Miguel Hernández–Lobato$^1$
Harvard Intelligent Probabilistic Systems Group
School of Engineering and Applied Sciences
Harvard University

http://jmhl.org, jmh@seas.harvard.edu

Joint work with
Yingzhen Li$^1$, Mark Rowland, Daniel Hernández-Lobato,

$^1$Equal contributors.
\[ p(y|\text{Data}) = \int p(y|\theta)p(\theta|\text{Data}) d\theta \]  

Inference algorithm
Model's predictive distribution

\[ p(y|\text{Data}) = \int p(y|\theta)p(\theta|\text{Data}) \, d\theta \]
Inference algorithm

Posterior distribution

Model's predictive distribution

\[ p(y|\text{Data}) = \int p(y|\theta)p(\theta|\text{Data}) \, d\theta \]

Inference algorithm
\[ p(y|\text{Data}) = \int p(y|\theta)p(\theta|\text{Data})d\theta \]

Inference algorithm

Approximation: \( q(\theta) = \) [diagram]
**α-divergence**

\[
D_\alpha(p||q) = \int_\theta \frac{(\alpha p(\theta) + (1 - \alpha)q(\theta) - p(\theta)^\alpha q(\theta)^{1-\alpha})}{\alpha(1 - \alpha)} d\theta .
\]

[Amari, 1985].
**α-divergence**

\[
D_{\alpha}(p||q) = \frac{\int_{\theta} \left( \alpha p(\theta) + (1 - \alpha)q(\theta) - p(\theta)^{\alpha} q(\theta)^{1-\alpha} \right) d\theta}{\alpha(1 - \alpha)}.
\]

[Amari, 1985].

![Figure source: [Minka, 2005]](image-url)
\( \alpha \)-divergence

\[
D_\alpha(p||q) = \int_\theta \left( \alpha p(\theta) + (1 - \alpha)q(\theta) - p(\theta)^\alpha q(\theta)^{1-\alpha} \right) d\theta \div \alpha(1 - \alpha).
\]

[Amari, 1985].

Variational Bayes (VB)

0.5 10

q tends to fit a mode of \( p \)

q tends to fit globally

Expectation propagation (EP)

Variational Bayes (VB)

KL(q||p)

KL(p||q)

Figure source: [Minka, 2005].
\( \alpha \)-divergence

\[
D_\alpha(p||q) = \frac{\int_\theta (\alpha p(\theta) + (1 - \alpha)q(\theta) - p(\theta)^\alpha q(\theta)^{1-\alpha})}{\alpha(1 - \alpha)} d\theta.
\]

[Amari, 1985].

Figure source: [Minka, 2005].
Variational Bayes

\[ \lim_{\alpha \to 0} D_\alpha(p||q) = KL(q||p) = -E_q[\log p(\theta)] - H[q] \]
Variational Bayes

\[
\lim_{\alpha \to 0} D_\alpha(p\|q) = KL(q\|p) = -E_q[\log p(\theta)] - H[q]
\]
Variational Bayes

\[ \lim_{\alpha \to 0} D_\alpha(p \| q) = KL(q \| p) = -\mathbb{E}_q[\log p(\theta)] - H[q] \]
Variational Bayes

\[
\lim_{{\alpha \to 0}} D_\alpha(p\|q) = KL(q\|p) = -E_q[\log p(\theta)] - H[q]
\]

There are automatic (black-box) tools for minimizing this objective using

- Stochastic optimization.
- Automatic differentiation.

[Kucukelbir et al., 2015, Ranganath et al., 2014, Salimans et al., 2013].
Variational Bayes

\[
\lim_{\alpha \to 0} D_\alpha(p \| q) = KL(q \| p) = -\mathbb{E}_q[\log p(\theta)] - H[q]
\]

There are automatic (black-box) tools for minimizing this objective using

- Stochastic optimization.
- Automatic differentiation.

[Kucukelbir et al., 2015, Ranganath et al., 2014, Salimans et al., 2013].

Can we have similar tools for other values of \( \alpha \)?
Local $\alpha$-divergence minimization (Power EP)

Approximates

\[ p(\theta) \propto p_0(\theta) \prod_{n=1}^{N} f_n(\theta) \quad \text{with} \quad q(\theta) \propto p_0(\theta) \prod_{n=1}^{N} \tilde{f}_n(\theta) \]

[Minka, 2004]
Local $\alpha$-divergence minimization (Power EP)

Approximates

$$p(\theta) \propto p_0(\theta) \prod_{n=1}^{N} f_n(\theta)$$

with

$$q(\theta) \propto p_0(\theta) \prod_{n=1}^{N} \tilde{f}_n(\theta)$$

[Minka, 2004]
Local $\alpha$-divergence minimization (Power EP)

Approximates

\[
p(\theta) \propto p_0(\theta) \prod_{n=1}^{N} f_n(\theta)
\]

with

\[
q(\theta) \propto p_0(\theta) \prod_{n=1}^{N} \tilde{f}_n(\theta)
\]

[Minka, 2004]

The $\tilde{f}_n$ are tuned by minimizing the local $\alpha$-divergences

\[
D_\alpha[p_n||q] \quad \text{for } n = 1, \ldots, N,
\]

where

\[
p_n(\theta) \propto f_n(\theta) \prod_{j \neq n} \tilde{f}_j(\theta)
\]

\[
q(\theta) \propto \tilde{f}_n(\theta) \prod_{j \neq n} \tilde{f}_j(\theta)
\]
The Power-EP approximation to the **evidence** is given by

\[
\log Z_{\text{PEP}} = \log Z_q + \sum_{n=1}^{N} \frac{1}{\alpha} \log E_q \left[ \left( \frac{f_n(\theta)}{\tilde{f}_n(\theta)} \right)^\alpha \right],
\]

The power-EP solution for \( q \) can be obtained by solving

\[
\max_q \min_{\tilde{f}_1, \ldots, \tilde{f}_N} \log Z_{\text{PEP}} \text{ subject to } q(\theta) = p_0(\theta) N \prod_{n=1}^{N} \tilde{f}_n(\theta).
\]

Solved with double-loop algorithm \[\text{[Heskes et al., 2002]}\].

Too slow in practice!
The Power-EP approximation to the evidence is given by

\[
\log Z_{\text{PEP}} = \log Z_q + \sum_{n=1}^{N} \frac{1}{\alpha} \log E_q \left[ \left( \frac{f_n(\theta)}{\tilde{f}_n(\theta)} \right)^\alpha \right],
\]

The power-EP solution for \( q \) can be obtained by solving

\[
\max_q \min_{\tilde{f}_1, \ldots, \tilde{f}_N} \log Z_{\text{PEP}} \quad \text{subject to} \quad q(\theta) = p_0(\theta) \prod_{n=1}^{N} \tilde{f}_n(\theta).
\]

Solved with double-loop algorithm [Heskes et al., 2002]. Too slow in practice!
The Power-EP approximation to the evidence is given by

\[
\log Z_{\text{PEP}} = \log Z_q + \sum_{n=1}^{N} \frac{1}{\alpha} \log E_q \left[ \left( \frac{f_n(\theta)}{\tilde{f}_n(\theta)} \right)^\alpha \right],
\]

The power-EP solution for \( q \) can be obtained by solving

\[
\max_q \min_{\tilde{f}_1, \ldots, \tilde{f}_N} \log Z_{\text{PEP}} \quad \text{subject to} \quad q(\theta) = p_0(\theta) \prod_{n=1}^{N} \tilde{f}_n(\theta).
\]

Solved with double-loop algorithm [Heskes et al., 2002].
The Power-EP approximation to the evidence is given by

\[
\log Z_{\text{PEP}} = \log Z_q + \sum_{n=1}^{N} \frac{1}{\alpha} \log E_q \left[ \left( \frac{f_n(\theta)}{\tilde{f}_n(\theta)} \right)^\alpha \right],
\]

The power-EP solution for \( q \) can be obtained by solving

\[
\max_q \min_{\tilde{f}_1, \ldots, \tilde{f}_N} \log Z_{\text{PEP}} \quad \text{subject to} \quad q(\theta) = p_0(\theta) \prod_{n=1}^{N} \tilde{f}_n(\theta).
\]

Solved with double-loop algorithm [Heskes et al., 2002]. Too slow in practice!
Main contribution

\[ p(\theta) \propto p_0(\theta) f_1(\theta) f_2(\theta) f_3(\theta) \quad \approx \quad q(\theta) \propto p_0(\theta) \tilde{f}_1(\theta) \tilde{f}_2(\theta) \tilde{f}_3(\theta) \]

We tie the factor approximations

\[ p(\theta) \propto p_0(\theta) f_1(\theta) f_2(\theta) f_3(\theta) \quad \approx \quad q(\theta) \propto p_0(\theta) \tilde{f}(\theta)^N \]
Main contribution

We tie the factor approximations

\[ p(\theta) \propto p_0(\theta) f_1(\theta) f_2(\theta) f_3(\theta) \approx q(\theta) \propto p_0(\theta) \tilde{f}_1(\theta) \tilde{f}_2(\theta) \tilde{f}_3(\theta) \]

\[ p(\theta) \propto p_0(\theta) f_1(\theta) f_2(\theta) f_3(\theta) \approx q(\theta) \propto p_0(\theta) \tilde{f}(\theta)^N \]

- \textbf{max} \ \textbf{min} \ \text{problem} \rightarrow \textbf{max} \ \text{problem, no double-loop needed!}
Main contribution

\[
p(\theta) \propto p_0(\theta) f_1(\theta) f_2(\theta) f_3(\theta) \quad \approx \quad q(\theta) \propto p_0(\theta) \tilde{f}_1(\theta) \tilde{f}_2(\theta) \tilde{f}_3(\theta)
\]

\[
p(\theta) \propto p_0(\theta) f_1(\theta) f_2(\theta) f_3(\theta) \quad \approx \quad q(\theta) \propto p_0(\theta) \tilde{f}(\theta)^N
\]

- **max min** problem $\rightarrow$ **max** problem, **no double-loop needed!**

- **Memory saving** scales as $\mathcal{O}(N)$. 
\[
\log Z_q + \sum_{n=1}^{N} \frac{1}{\alpha} \log \mathbb{E}_q \left[ \left( \frac{f_n(\theta)}{\tilde{f}_n(\theta)} \right)^\alpha \right]
\]
\[
\log Z_q + \sum_{n=1}^{N} \frac{1}{\alpha} \log E_q \left[ \left( \frac{f_n(\theta)}{\tilde{f}(\theta)} \right)^\alpha \right]
\]
\[ \log Z_q + \sum_{n=1}^{N} \frac{1}{\alpha} \log \mathbb{E}_q \left[ \left( \frac{f_n(\theta)}{\tilde{f}_n(\theta)} \right)^{\alpha} \right] \]

\[ \log Z_q + \sum_{n=1}^{N} \frac{1}{\alpha} \log \mathbb{E}_q \left[ \left( \frac{f_n(\theta)}{\tilde{f}(\theta)} \right)^{\alpha} \right] \]
\[
\log Z_q + \sum_{n=1}^{N} \frac{1}{\alpha} \log E_q \left[ \left( \frac{\tilde{f}(\theta)}{f_n(\theta)} \right)^\alpha \right]
\]

\[
\downarrow
\]

\[
\log Z_q + \sum_{n=1}^{N} \frac{1}{\alpha} \log E_q \left[ \left( \frac{f_n(\theta)}{\tilde{f}(\theta)} \right)^\alpha \right]
\]

\[
\downarrow
\]

\[
\log Z_q + \frac{N}{|S|} \sum_{n \in S} \frac{1}{\alpha} \log \frac{1}{K} \sum_{k=1}^{K} \left( \frac{f_n(\theta_k)}{\tilde{f}(\theta_k)} \right)^\alpha
\]

\[\theta_1, \ldots, \theta_K \sim q\]
\[
\log Z_q + \sum_{n=1}^{N} \frac{1}{\alpha} \log E_q \left( \left( \frac{f_n(\theta)}{\tilde{f}(\theta)} \right)^{\alpha} \right)
\]

\[
\log Z_q + \sum_{n=1}^{N} \frac{1}{\alpha} \log E_q \left( \left( \frac{f_n(\theta)}{\tilde{f}(\theta)} \right)^{\alpha} \right)
\]

\[
\log Z_q + \frac{N}{|S|} \sum_{n \in S} \frac{1}{\alpha} \log \frac{1}{K} \sum_{k=1}^{K} \left( \frac{f_n(\theta_k)}{\tilde{f}(\theta_k)} \right)^{\alpha}
\]

\[\theta_1, \ldots, \theta_K \sim q\]

**Stochastic** estimate of the objective for **automatic, scalable** inference.
\[
\log Z_q + \sum_{n=1}^{N} \frac{1}{\alpha} \log \mathbb{E}_q \left[ \left( \frac{f_n(\theta)}{\tilde{f}(\theta)} \right)^\alpha \right] \\
\log Z_q + \sum_{n=1}^{N} \frac{1}{\alpha} \log \mathbb{E}_q \left[ \left( \frac{f_n(\theta)}{\tilde{f}(\theta)} \right)^\alpha \right] \\
\log Z_q + \frac{N}{|S|} \sum_{n \in S} \frac{1}{\alpha} \log \frac{1}{K} \sum_{k=1}^{K} \left( \frac{f_n(\theta_k)}{\tilde{f}(\theta_k)} \right)^\alpha
\]

\[\theta_1, \ldots, \theta_K \sim q\]

**Stochastic** estimate of the objective for **automatic, scalable** inference.

**Biased estimator!**
Regression with neural networks and 2D Gaussian

Predictive distribution

Example with 2D Gaussians

Ground truth

Exact Posterior

Weight 1

Weight 2
Regression with neural networks and 2D Gaussian

Alpha = -1.00

Predictive distribution

Example with 2D Gaussians

Weight 1

Weight 2
Regression with neural networks and 2D Gaussian

\[ \text{Alpha} = -0.95 \]

\[ \begin{array}{c|c|c}
  \text{VB} & \text{EP} \\
  \hline
  -1.0 & \vdots & \vdots \\
  -0.5 & \vdots & \vdots \\
  0.0 & \vdots & \vdots \\
  0.5 & \vdots & \vdots \\
  1.0 & \vdots & \vdots \\
\end{array} \]

Predictive distribution

- Ground truth
- Mean predictions
- 3 standard deviations

Example with 2D Gaussians

Weight 1

Weight 2
Regression with neural networks and 2D Gaussian

\[ \text{Alpha} = -0.87 \]

Predictive distribution

Example with 2D Gaussians

- Ground truth
- Mean predictions
- 3 standard deviations

Weights 1 and 2
Regression with neural networks and 2D Gaussian

\[ \text{Predictive distribution} \]

- Ground truth
- Mean predictions
- 3 standard deviations

\[ \text{Example with 2D Gaussians} \]

- Exact Posterior
- Approximation

Alpha = −0.79
Regression with neural networks and 2D Gaussian

Alpha = −0.72

Predictive distribution

Ground truth
Mean predictions
3 standard deviations

Example with 2D Gaussians

Exact Posterior
Approximation
Regression with neural networks and 2D Gaussian

\[ \text{Alpha} = -0.64 \]

Example with 2D Gaussians

- Exact Posterior
- Approximation

Ground truth
Mean predictions
3 standard deviations
Regression with neural networks and 2D Gaussian

\[ \text{Alpha} = -0.56 \]

\[ \begin{array}{c|c|c}
\text{VB} & \text{EP} \\
\hline
-1.0 & \hline
-0.5 & \text{Alpha} \\
0.0 & \\
0.5 & \\
1.0 & \\
\end{array} \]

**Predictive distribution**

- Ground truth
- Mean predictions
- 3 standard deviations

**Example with 2D Gaussians**

- Exact Posterior
- Approximation

Weight 1

Weight 2
Regression with neural networks and 2D Gaussian

\[ \text{Alpha} = -0.49 \]

\[ \begin{array}{c|c|c}
\text{VB} & \text{EP} \\
\hline
-1.0 & \text{\textbullet} \\
-0.5 & \text{\textbullet} \\
0.0 & \text{\textbullet} \\
0.5 & \text{\textbullet} \\
1.0 & \text{\textbullet}
\end{array} \]

\text{Predictive distribution}

\[ \text{Ground truth} \]
\[ \text{Mean predictions} \]
\[ \text{3 standard deviations} \]

\text{Example with 2D Gaussians}

\[ \text{Exact Posterior} \]
\[ \text{Approximation} \]

\[ \text{Weight 1} \]
\[ \text{Weight 2} \]
Regression with neural networks and 2D Gaussian

Alpha = -0.41

Example with 2D Gaussians

Weight 1

Weight 2
Regression with neural networks and 2D Gaussian

Predictive distribution

Example with 2D Gaussians

Ground truth
Mean predictions
3 standard deviations

Exact Posterior
Approximation

Weight 1
Weight 2

Alpha = -0.33

-1.0 -0.5 0.0 0.5 1.0

EP
VB
Regression with neural networks and 2D Gaussian

Alpha = -0.26

Predictive distribution

Example with 2D Gaussians

Ground truth
Mean predictions
3 standard deviations

Exact Posterior
Approximation
Regression with neural networks and 2D Gaussian

Alpha = −0.18

Predictive distribution

Example with 2D Gaussians

Ground truth
Mean predictions
3 standard deviations

Exact Posterior
Approximation

Weight 1
Weight 2
Regression with neural networks and 2D Gaussian

Predictive distribution

Example with 2D Gaussians

Ground truth
Mean predictions
3 standard deviations

Weight 1
Weight 2
Regression with neural networks and 2D Gaussian

Alpha = 0.00

Predictive distribution

Example with 2D Gaussians

Weight 1

Weight 2

Ground truth
Mean predictions
3 standard deviations
Regression with neural networks and 2D Gaussian

\[ \text{Predictive distribution} \]

- Ground truth
- Mean predictions
- 3 standard deviations

\[ \text{Example with 2D Gaussians} \]

- Exact Posterior
- Approximation

\[ \text{Weight 1} \]

\[ \text{Weight 2} \]
Regression with neural networks and 2D Gaussian

![Graph showing predictive distribution and example with 2D Gaussians]

- Alpha = 0.13
- Predictive distribution
- Ground truth
- Mean predictions
- 3 standard deviations

Example with 2D Gaussians

- Weight 1
- Weight 2
Regression with neural networks and 2D Gaussian

Alpha = 0.21

Predictive distribution

- Ground truth
- Mean predictions
- 3 standard deviations

Example with 2D Gaussians

- Exact Posterior
- Approximation

Weight 1
Weight 2
Regression with neural networks and 2D Gaussian

\[ \text{Alpha} = 0.28 \]

Predictive distribution

Example with 2D Gaussians

- Ground truth
- Mean predictions
- 3 standard deviations

Weight 1

Weight 2

Exact Posterior
Approximation
Regression with neural networks and 2D Gaussian

\[ \text{Alpha} = 0.36 \]

Predictive distribution

Example with 2D Gaussians

Ground truth
Mean predictions
3 standard deviations

Weight 1

Weight 2

Exact Posterior
Approximation
Regression with neural networks and 2D Gaussian

\[ \text{Alpha} = 0.44 \]

**Predictive distribution**

- Ground truth
- Mean predictions
- 3 standard deviations

**Example with 2D Gaussians**

- Weight 1
- Weight 2
Regression with neural networks and 2D Gaussian

Alpha = 0.51

Predictive distribution

Example with 2D Gaussians

Weight 1

Weight 2

Ground truth
Mean predictions
3 standard deviations

Exact Posterior
Approximation
Regression with neural networks and 2D Gaussian

\[ \alpha = 0.59 \]

Example with 2D Gaussians

- Ground truth
- Mean predictions
- 3 standard deviations

Predictive distribution

\[ f(x) \]

Weight 1

Weight 2
Regression with neural networks and 2D Gaussian

Alpha = 0.67

Predictive distribution

Example with 2D Gaussians
Regression with neural networks and 2D Gaussian

Alpha = 0.74

Predictive distribution

Ground truth
Mean predictions
3 standard deviations

Example with 2D Gaussians

Weight 1
Weight 2
Regression with neural networks and 2D Gaussian

Alpha = 0.82

Predictive distribution

Example with 2D Gaussians

Weight 1

Weight 2
Regression with neural networks and 2D Gaussian

Alpha = 0.90

Predictive distribution

Example with 2D Gaussians

Ground truth
Mean predictions
3 standard deviations

Exact Posterior
Approximation

Weight 1
Weight 2

X
Regression with neural networks and 2D Gaussian

Alpha = 0.97

Predictive distribution

Ground truth
Mean predictions
3 standard deviations

Example with 2D Gaussians

Weight 1
Weight 2
Regression with neural networks and 2D Gaussian

Alpha = 1.00

Predictive distribution

Example with 2D Gaussians

Weight 1

Weight 2
Harvard clean energy project. Data from 60,000 organic photovoltaics.

Bayesian neural networks, 2 hidden layers with 400 units each.
Harvard clean energy project. Data from 60,000 organic photovoltaics.

Bayesian neural networks, 2 hidden layers with 400 units each.

**Table:** Average Test Error and Test Log-likelihood.

<table>
<thead>
<tr>
<th></th>
<th>$\alpha=1.0$ (EP)</th>
<th>$\alpha=0.5$</th>
<th>$\alpha=0$</th>
<th>VB</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Avg. Error</strong></td>
<td>1.28±0.01</td>
<td><strong>1.08±0.01</strong></td>
<td>1.13±0.01</td>
<td>1.14±0.01</td>
</tr>
<tr>
<td><strong>Avg. Log-likelihood</strong></td>
<td>-0.93±0.01</td>
<td><strong>-0.74±0.01</strong></td>
<td>-1.39±0.03</td>
<td>-1.38±0.02</td>
</tr>
</tbody>
</table>
The value $\alpha = 0.5$ seems to be consistently better than $\alpha = 1$ (EP) or $\alpha = 0$ (VB).
\[ \alpha = 0.5 \]

Variational Bayes

\[ \frac{1}{\alpha} \log E_{q_{\text{cav}}} [f_n(\theta)^{\alpha}] \quad E_q [\log f_n(\theta)] \]

Depeweg et al. [2016]
Take home message

Black-box $\alpha$-divergence minimization...

1. generalizes VB and an EP-like algorithm.
2. better than any of them in prediction problems with neural nets.
3. straightforward to apply in very complex models.

• Important applications in probabilistic programming to go beyond MCMC and VB.

Stan


References II


Average Bias in the Gradient.

$K = 5$

\[
\begin{array}{ccc}
\alpha = 1.0 & \alpha = 0.5 & \alpha = 0 \\
\end{array}
\]

\[
\begin{array}{cccc}
\log_{10} \text{bias} & -5 & -4 & -3 \\
\end{array}
\]

$K = 10$

\[
\begin{array}{ccc}
\alpha = 1.0 & \alpha = 0.5 & \alpha = 0 \\
\end{array}
\]

\[
\begin{array}{cccc}
\log_{10} \text{bias} & -5 & -4 & -3 \\
\end{array}
\]
Average Bias in the Gradient.

\[ K = 5 \]

\[
\begin{array}{ccc}
\log_{10} \text{bias} & \alpha = 1.0 & \alpha = 0.5 & \alpha = 0 \\
-5 & -4 & -3 & -2 & -1 & 0 & 1
\end{array}
\]

\[ K = 10 \]

\[
\begin{array}{ccc}
\log_{10} \text{bias} & \alpha = 1.0 & \alpha = 0.5 & \alpha = 0 \\
-5 & -4 & -3 & -2 & -1 & 0 & 1
\end{array}
\]

Average Standard Deviation in the Gradient.

\[ K = 5 \]

\[
\begin{array}{ccc}
\log_{10} \text{std. dev.} & \alpha = 1.0 & \alpha = 0.5 & \alpha = 0 \\
-5 & -4 & -3 & -2 & -1 & 0 & 1
\end{array}
\]

\[ K = 10 \]

\[
\begin{array}{ccc}
\log_{10} \text{std. dev.} & \alpha = 1.0 & \alpha = 0.5 & \alpha = 0 \\
-5 & -4 & -3 & -2 & -1 & 0 & 1
\end{array}
\]
Average Bias in the Gradient.

\( K = 5 \)

\[
\begin{align*}
\log_{10} \text{bias} \\
\alpha = 1.0 & \quad \alpha = 0.5 & \quad \alpha = 0
\end{align*}
\]

\( K = 10 \)

\[
\begin{align*}
\log_{10} \text{bias} \\
\alpha = 1.0 & \quad \alpha = 0.5 & \quad \alpha = 0
\end{align*}
\]

Average Standard Deviation in the Gradient.

\( K = 5 \)

\[
\begin{align*}
\log_{10} \text{standard dev.} \\
\alpha = 1.0 & \quad \alpha = 0.5 & \quad \alpha = 0
\end{align*}
\]

\( K = 10 \)

\[
\begin{align*}
\log_{10} \text{standard dev.} \\
\alpha = 1.0 & \quad \alpha = 0.5 & \quad \alpha = 0
\end{align*}
\]
Multi-class classification

Bayesian neural networks, 2 hidden layers with 400 units each.

**Table:** Average Test Error and Test Log-likelihood in MNIST.

<table>
<thead>
<tr>
<th>MNIST</th>
<th>( \alpha = 1.0 )</th>
<th>( \alpha = 0.5 )</th>
<th>( \alpha = 0 )</th>
<th>VB</th>
<th>( \alpha = -1.0 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error</td>
<td>0.0151±0.0001</td>
<td>0.0144±0.0001</td>
<td>0.0136±0.0001</td>
<td>0.0136±0.0001</td>
<td><strong>0.0133±0.0001</strong></td>
</tr>
<tr>
<td>Log-likelihood</td>
<td>-0.0551±0.0004</td>
<td>-0.0509±0.0003</td>
<td>-0.0468±0.0002</td>
<td>-0.0468±0.0002</td>
<td><strong>-0.0447±0.0002</strong></td>
</tr>
</tbody>
</table>