Convolutional neural networks and independent component analysis of multi-filter imaging data, part II

Laurence Perreault Levasseur

What are the uncertainties of the output parameters?

Figure 1: Comparison of estimated parameters with their true values. The estimated values of the Einstein radius $\theta_E$ (a) and the $x$ and $y$ components of the complex ellipticity $\varepsilon_x$ and $\varepsilon_y$ (b and c) are shown on the $y$ axis; the true values are shown on the $x$ axis. The red dashed line marks the $y=x$ diagonal, on which perfectly recovered parameters should lie. The shaded blue areas represent the 68% and 95% intervals of the parameters recovered from a test set that the network has not been trained on. The small grey dots show the parameters of 10,000 test samples. The coloured data points and their error bars (95% confidence) correspond to real HST images of gravitational lenses, with the true parameters set to previously published values.17

WHAT ARE THE UNCERTAINTIES OF THE OUTPUT PARAMETERS?

SOURCES OF ERRORS IN THE PREDICTIONS:

1- **Aleatoric**.
Inherent corruptions to the input data: noise, PSF blurring, etc.

2- **Epistemic**.
Errors made by the networks: these could be due to insufficient training, network architecture, etc.
WHAT IS THE LOG-LIKELIHOOD OF THE NETWORK OUTPUT, $\mathcal{L}(y_n, \hat{y}_n(x_n, \omega))$?

We approximate the likelihood with an analytic distribution.

Assume Gaussian:

$$\mathcal{L}(y_n, \hat{y}_n(x_n, \omega)) \propto \sum_k \frac{-1}{2\sigma_k^2} ||y_{n,k} - \hat{y}_{n,k}(x_n, \omega)||^2 - \frac{1}{2} \log \sigma_k^2$$
EPISTEMIC UNCERTAINTIES

STANDARD NEURAL NETWORKS:
WEIGHT HAVE FIXED, DETERMINISTIC VALUES

INPUT

HIDDEN

OUTPUT

1 0.8
0.4
0.3
0.2
0.9
0.5

1
0.73
1.3
0.79

1.3
0.79

0.8
0.69

0.3
0.5
0.5
0.9

0
EPISTEMIC UNCERTAINTIES

BAYESIAN NEURAL NETWORKS:
INSTEAD OF FIX VALUES, WEIGHTS ARE DEFINED BY PROBABILITY DISTRIBUTIONS
Variational Inference

Replace $P(\omega)$ by a distribution with a simple analytic form, $q(\omega)$, (e.g., a Gaussian).
**Variational Inference**

\[
p(y|x, X, Y) = \int p(y|x, \omega) p(\omega|X, Y) \, d\omega
\]

\[
p(y|x) \approx \int p(y|x, \omega) q(\omega) \, d\omega
\]

\[
\mathcal{L}_{VI} = \int q(\omega) \log p(Y|X, \omega) \, d\omega - \text{KL}(q(\omega) \| p(\omega))
\]

**Determine** \(q(\omega)\) **during training, by optimizing the KL divergence**
Variational Inference

\[ p(y|x, X, Y) = \int p(y|x, \omega) p(\omega|X, Y) \, d\omega \]

\[ p(y|x) \approx \int p(y|x, \omega) q(\omega) \, d\omega \]

\[ \mathcal{L}_{VI} = \int q(\omega) \log p(Y|X, \omega) \, d\omega - KL(q(\omega)||p(\omega)) \]

Once \( q(\omega) \) is determined (network is trained) we can draw Monte Carlo samples by generating realizations of \( q(\omega) \)
\[
    p(y|x, X, Y) = \int p(y|x, \omega) p(\omega|X, Y) \, d\omega
\]

\[
    p(y|x) \approx \int p(y|x, \omega) q(\omega) \, d\omega
\]

\[
    \mathcal{L}_{\text{VI}} = \int q(\omega) \log p(Y|X, \omega) \, d\omega - \text{KL}(q(\omega)||p(\omega))
\]

\[
    \mathcal{L}_{\text{VI}} \sim \sum_{n=1}^{N} \mathcal{L}(y_n, \hat{y}_n(x_n, \omega)) - \lambda \sum_{i} \|\omega_i\|^2
\]

Variational Inference

Log-likelihood of the output parameters for the training set.
Easiest Variational Distribution

Bernoulli distribution:
Every weight is either zero (with probability $p$) or some value, $w$, (with probability $1-p$)

The variational parameter is the value of the weight when it is not zero.

That is DROPOUT!
RECAP

1- Place dropout before every weight layer.
2- Train with dropout, optimizing the log likelihood

\[ \mathcal{L}(y_n, \hat{y}_n(x_n, \omega)) \propto \sum_k \frac{-1}{2\sigma_k^2} ||y_{n,k} - \hat{y}_{n,k}(x_n, \omega)||^2 - \frac{1}{2} \log \sigma_k^2 \]

3- At test time, keep dropout on. Perform Monte Carlo dropout: input the data multiple times, perform dropout and collect the outputs.
4- Add your aleatoric uncertainty (the sigma above) to the sample.
5- Done
EXAMPLE

UNCERTAINTIES ON THE MAGNIFICATION OF LENSES

Figure 1. Predicted 68.3% uncertainties for lensing flux magnification, $\mu_F$, as a function of the true value of this parameter. The orange, blue, and black correspond to examples where the true values fall within the 68.3, 95.5, and 99.7% confidence intervals respectively.

ARE THESE UNCERTAINTIES ACCURATE?

Coverage probability of a confidence interval is the proportion of the time that the interval contains the true value of interest.

For an accurate interval estimator, the coverage probability is equal to its confidence level.
COVERAGE PROBABILITIES

Some examples

Figure 4. A visual inspection of five test images with increasing uncertainties in their obtained parameters. As expected, lensing configurations with multiple opposing images and close to Einstein rings result in more precise estimates, while configurations similar to panel 5, with only a pair of compact images, have large uncertainties. All images contain similar noise levels. The uncertainty of each configuration (averaged over all parameters) is given in each panel.
WHAT ABOUT THE LIGHT OF THE LENSING GALAXY?
WHAT ABOUT THE LIGHT OF THE LENsing GALAXY?

Usually people fit a model (e.g., Sersic) to the light distribution of the lens galaxy and remove it from the data. This is:

1) Time consuming, requiring another non-linear optimization problem
2) Not automated, requiring guesses for starting points, a choice of an appropriate profile, etc.
3) Often leaves high residuals (Galaxies aren’t exactly Sersic, or King, etc.)
4) They don’t take advantage of color difference in the two sources.
Independent Component Analysis (ICA)

In signal processing, independent component analysis (ICA) is a computational method for separating a multivariate signal into additive subcomponents. This is done by assuming that the subcomponents are non-Gaussian signals and that they are statistically independent from each other. ICA is a special case of blind source separation. A common example application is the "cocktail party problem" of listening in on one person's speech in a noisy room.
INDEPENDENT COMPONENT ANALYSIS (ICA)

\[ O_1 = W_{11} S_1 + W_{12} S_2 \]
\[ O_2 = W_{21} S_1 + W_{22} S_2 \]
How does ICA work?

ICA takes advantage of the Central Limit Theorem: Addition of two or more non-Gaussian random variables results in a more Gaussian distribution that the individual distributions.

We examine the distribution of pixel values.
HOW DOES ICA WORK?
How does ICA work?

For any value of $X$, the distribution of $Y$ is unchanged
INDEPENDENT COMPONENT ANALYSIS (ICA)

Image in filter 1 → Put through ICA → Component 1

Image in filter 2 → Component 2
INDEPENDENT COMPONENT ANALYSIS (ICA)

- Fast (fraction of a second)
- Automated (no interaction needed)
- Can handle complex morphologies
- Takes advantage of color information
USE ANOTHER MACHINE LEARNING TOOL: INDEPENDENT COMPONENT ANALYSIS (ICA)

ICA Ambiguities

- ICA can’t determine the absolute scaling of the components.
- ICA can’t determine the sign of the components.
- ICA can’t determine an order for the components.
APPLICATION OF ICA TO TWO BLENDED GALAXIES

OBSERVATIONS

FILTER 1

FILTER 2
APPLICATION OF ICA TO TWO BLENDED GALAXIES

**Observations**

*Filter 1*

*Filter 2*

**ICA Components**

*Component 1*

*Component 2*
APPLICATION OF ICA TO TWO BLENDED GALAXIES

Observations | ICA Components | Truth
---|---|---
Filter 1 | Component 1 | Galaxy 1
Filter 2 | Component 2 | Galaxy 2
ICA IS NOT LIMITED TO TWO COMPONENTS:
APPLICATION OF ICA TO FOUR BLENDED GALAXIES

Observations

ICA components
ICA rotates the data to maximize the non-Gaussianity of each axis.

There are various measures of non-Gaussianity used, such as kurtosis or negative entropy.

Perhaps for specific astronomical applications, a tailored measure of non-Gaussianity could improve performance.
ICA LIMITATIONS:
1. INDEPENDENCE

**Truth**

- Galaxy 1
- Galaxy 2

**Observations**

- Filter 1
- Filter 2

**ICA components**

- Component 1
- Component 2
ICA LIMITATIONS:
1. INDEPENDENCE

<table>
<thead>
<tr>
<th>Truth</th>
<th>Observations</th>
<th>ICA components</th>
</tr>
</thead>
<tbody>
<tr>
<td>Galaxy 1</td>
<td>Filter 1</td>
<td>Component 1</td>
</tr>
<tr>
<td>Galaxy 2</td>
<td>Filter 2</td>
<td>Component 2</td>
</tr>
</tbody>
</table>
ICA LIMITATIONS:

1. INDEPENDENCE.
2. NON-GAUSSIAN SIGNALS ONLY: e.g., WE CAN’T APPLY IT TO CMB DATA.
3. NO PRIORS:
   1. CAN USE DEEP NETWORKS TO DO THE DE-BELNDING, IMPOSING PRIORS THROUGH THEIR TRAINING SETS.
ICA LIMITATIONS:

3. PRIOR
DEEP CONVOLUTIONAL NEURAL NETWORKS FOR COMPONENT SEPARATION
Deep convolutional neural networks for component separation

Unlike ICA, the network will separate components taking their priors into account. These priors come from training data.
THANK YOU