LET'S DO BAYESIAN MACHINE LEARNING

A 10 minute sprint from practitioner's perspective

Shlomo Kashani

PyData 2017, Tel-Aviv, Faburary 16, 2017.

WWW.DEEP-ML.COM
github.com/QuantScientist
twitter.com/QuantScientist

GREAT LET'S DO IT, I want to become a Bayesian BUT WAIT...

Isn't Bayesian Machine Learning really difficult!?:

[1] http://twiecki.github.io/blog/2014/01/02/visualizing-mcmc/

[2] http://edwardlib.org/tutorials/ppc

[3] www.stat.columbia.edu/~gelman/research/published/stan_jebs_2.pdf

[4] https://arxiv.org/abs/1111.4246

GREAT LET'S DO IT, I want to become a Bayesian BUT WAIT...

Isn't Bayesian Machine Learning really difficult!?:

- Metropolis–Hastings algorithm!? [1]

[1] http://twiecki.github.io/blog/2014/01/02/visualizing-mcmc/

[2] http://edwardlib.org/tutorials/ppc

[3] www.stat.columbia.edu/~gelman/research/published/stan_jebs_2.pdf

[4] https://arxiv.org/abs/1111.4246

GREAT LET'S DO IT, I want to become a Bayesian BUT WAIT...

Isn't Bayesian Machine Learning really difficult!?:

- Metropolis–Hastings algorithm!? [1]
- Posterior Predictive Distribution!? [2]

[1] http://twiecki.github.io/blog/2014/01/02/visualizing-mcmc/

[2] http://edwardlib.org/tutorials/ppc

[3] www.stat.columbia.edu/~gelman/research/published/stan_jebs_2.pdf

[4] https://arxiv.org/abs/1111.4246

GREAT LET'S DO IT, I want to become a Bayesian BUT WAIT...

Isn't Bayesian Machine Learning really difficult!?:

- Metropolis–Hastings algorithm!? [1]
- Posterior Predictive Distribution!? [2]
- Hamiltonian Monte Carlo (HMC)!? [3]

[1] http://twiecki.github.io/blog/2014/01/02/visualizing-mcmc/

[2] http://edwardlib.org/tutorials/ppc

[3] www.stat.columbia.edu/~gelman/research/published/stan_jebs_2.pdf

[4] https://arxiv.org/abs/1111.4246

GREAT LET'S DO IT, I want to become a Bayesian BUT WAIT...

Isn't Bayesian Machine Learning really difficult!?:

- Metropolis–Hastings algorithm!? [1]
- Posterior Predictive Distribution!? [2]
- Hamiltonian Monte Carlo (HMC)!? [3]
- No-U-Turn Sampler (NUTS)!? [4]

[1] http://twiecki.github.io/blog/2014/01/02/visualizing-mcmc/

[2] http://edwardlib.org/tutorials/ppc

[3] www.stat.columbia.edu/~gelman/research/published/stan_jebs_2.pdf

[4] https://arxiv.org/abs/1111.4246

GREAT LET'S DO IT, I want to become a Bayesian BUT WAIT...

Isn't Bayesian Machine Learning really difficult!?:

- Metropolis–Hastings algorithm!? [1]
- Posterior Predictive Distribution!? [2]
- Hamiltonian Monte Carlo (HMC)!? [3]
- No-U-Turn Sampler (NUTS)!? [4]

[1] http://twiecki.github.io/blog/2014/01/02/visualizing-mcmc/

[2] http://edwardlib.org/tutorials/ppc

[3] www.stat.columbia.edu/~gelman/research/published/stan_jebs_2.pdf

[4] https://arxiv.org/abs/1111.4246

TODAY'S 10 MINUTE GOAL

In the next 10 mins you will:

- Gain a better understanding of Bayesian Machine Learning, Learn (a bit) about the capabilities of PyMC3 and PyStan Start using BML (... maybe).
- Bayesian Machine Learning Problems
- Basic Theory
- The Beta-Binomial model
- Bayesian Lasso/Ridge Logistic Regression
- PyMC3, PyStan

TODAY'S 10 MINUTE GOAL

In the next 10 mins you will:

- Gain a better understanding of Bayesian Machine Learning, Learn (a bit) about the capabilities of PyMC3 and PyStan Start using BML (... maybe).
- Bayesian Machine Learning Problems
- Basic Theory
- The Beta-Binomial model
- Bayesian Lasso/Ridge Logistic Regression
- PyMC3, PyStan

TODAY'S 10 MINUTE GOAL

In the next 10 mins you will:

- Gain a better understanding of Bayesian Machine Learning, Learn (a bit) about the capabilities of PyMC3 and PyStan Start using BML (... maybe).
- Bayesian Machine Learning Problems
- Basic Theory
- The Beta-Binomial model
- Bayesian Lasso/Ridge Logistic Regression
- PyMC3, PyStan

TODAY'S 10 MINUTE GOAL

In the next 10 mins you will:

- Gain a better understanding of Bayesian Machine Learning, Learn (a bit) about the capabilities of PyMC3 and PyStan Start using BML (... maybe).
- Bayesian Machine Learning Problems
- Basic Theory
- The Beta-Binomial model
- Bayesian Lasso/Ridge Logistic Regression
- PyMC3, PyStan

TODAY'S 10 MINUTE GOAL

In the next 10 mins you will:

- Gain a better understanding of Bayesian Machine Learning, Learn (a bit) about the capabilities of PyMC3 and PyStan Start using BML (... maybe).
- Bayesian Machine Learning Problems
- Basic Theory
- The Beta-Binomial model
- Bayesian Lasso/Ridge Logistic Regression
- PyMC3, PyStan

TODAY'S 10 MINUTE GOAL

In the next 10 mins you will:

- Gain a better understanding of Bayesian Machine Learning, Learn (a bit) about the capabilities of PyMC3 and PyStan Start using BML (... maybe).
- Bayesian Machine Learning Problems
- Basic Theory
- The Beta-Binomial model
- Bayesian Lasso/Ridge Logistic Regression
- PyMC3, PyStan

TODAY'S 10 MINUTE GOAL

In the next 10 mins you will:

- Gain a better understanding of Bayesian Machine Learning, Learn (a bit) about the capabilities of PyMC3 and PyStan Start using BML (... maybe).
- Bayesian Machine Learning Problems
- Basic Theory
- The Beta-Binomial model
- Bayesian Lasso/Ridge Logistic Regression
- PyMC3, PyStan

TODAY'S 10 MINUTE GOAL

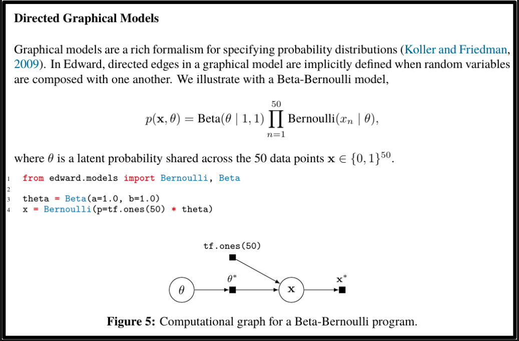
In the next 10 mins you will:

- Gain a better understanding of Bayesian Machine Learning, Learn (a bit) about the capabilities of PyMC3 and PyStan Start using BML (... maybe).
- Bayesian Machine Learning Problems
- Basic Theory
- The Beta-Binomial model
- Bayesian Lasso/Ridge Logistic Regression
- PyMC3, PyStan

TODAY'S 10 MINUTE GOAL

In the next 10 mins you will:

- Edward [5] stands on a class of its own and based on TensorFlow. Read the paper about Deep Probabilistic Programming with Edward. [6]



Directed Graphical Models

Graphical models are a rich formalism for specifying probability distributions (Koller and Friedman, 2009). In Edward, directed edges in a graphical model are implicitly defined when random variables are composed with one another. We illustrate with a Beta-Bernoulli model,

$$p(\mathbf{x}, \theta) = \text{Beta}(\theta \mid 1, 1) \prod_{n=1}^{50} \text{Bernoulli}(x_n \mid \theta),$$

where θ is a latent probability shared across the 50 data points $\mathbf{x} \in \{0, 1\}^{50}$.

```
from edward.models import Bernoulli, Beta

theta = Beta(a=1.0, b=1.0)
x = Bernoulli(p=tf.ones(50) * theta)
```

Figure 5: Computational graph for a Beta-Bernoulli program.

[7]

[5] https://arxiv.org/abs/1610.09787

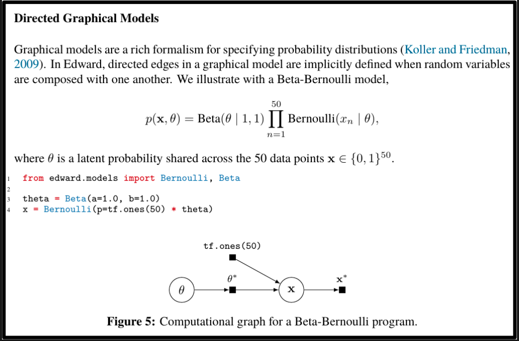
[6] https://arxiv.org/pdf/1701.03757.pdf

[7] http://austinrochford.com/resources/talks/dydata-dc-2016-variational-python.slides.html

TODAY'S 10 MINUTE GOAL

In the next 10 mins you will:

- Edward [5] stands on a class of its own and based on TensorFlow. Read the paper about Deep Probabilistic Programming with Edward. [6]



Directed Graphical Models

Graphical models are a rich formalism for specifying probability distributions (Koller and Friedman, 2009). In Edward, directed edges in a graphical model are implicitly defined when random variables are composed with one another. We illustrate with a Beta-Bernoulli model,

$$p(\mathbf{x}, \theta) = \text{Beta}(\theta \mid 1, 1) \prod_{n=1}^{50} \text{Bernoulli}(x_n \mid \theta),$$

where θ is a latent probability shared across the 50 data points $\mathbf{x} \in \{0, 1\}^{50}$.

```
from edward.models import Bernoulli, Beta

theta = Beta(a=1.0, b=1.0)
x = Bernoulli(p=tf.ones(50) * theta)
```

Figure 5: Computational graph for a Beta-Bernoulli program.

[7]

[5] https://arxiv.org/abs/1610.09787

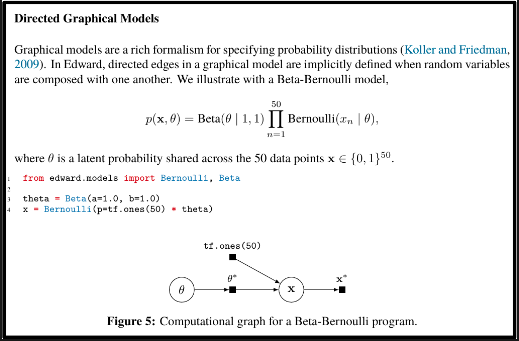
[6] https://arxiv.org/pdf/1701.03757.pdf

[7] http://austinrochford.com/resources/talks/dydata-dc-2016-variational-python.slides.html

TODAY'S 10 MINUTE GOAL

In the next 10 mins you will:

- Edward [5] stands on a class of its own and based on TensorFlow. Read the paper about Deep Probabilistic Programming with Edward. [6]



Directed Graphical Models

Graphical models are a rich formalism for specifying probability distributions (Koller and Friedman, 2009). In Edward, directed edges in a graphical model are implicitly defined when random variables are composed with one another. We illustrate with a Beta-Bernoulli model,

$$p(\mathbf{x}, \theta) = \text{Beta}(\theta \mid 1, 1) \prod_{n=1}^{50} \text{Bernoulli}(x_n \mid \theta),$$

where θ is a latent probability shared across the 50 data points $\mathbf{x} \in \{0, 1\}^{50}$.

```
from edward.models import Bernoulli, Beta

theta = Beta(a=1.0, b=1.0)
x = Bernoulli(p=tf.ones(50) * theta)
```

Figure 5: Computational graph for a Beta-Bernoulli program.

7

[5] https://arxiv.org/abs/1610.09787

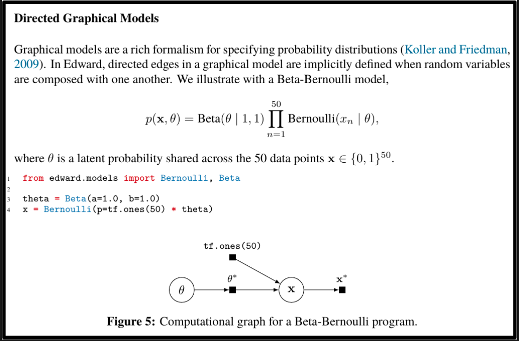
[6] https://arxiv.org/pdf/1701.03757.pdf

[7] http://austinrochford.com/resources/talks/dydata-dc-2016-variational-python.slides.html

TODAY'S 10 MINUTE GOAL

In the next 10 mins you will:

- Edward [5] stands on a class of its own and based on TensorFlow. Read the paper about Deep Probabilistic Programming with Edward. [6]



Directed Graphical Models

Graphical models are a rich formalism for specifying probability distributions (Koller and Friedman, 2009). In Edward, directed edges in a graphical model are implicitly defined when random variables are composed with one another. We illustrate with a Beta-Bernoulli model,

$$p(\mathbf{x}, \theta) = \text{Beta}(\theta \mid 1, 1) \prod_{n=1}^{50} \text{Bernoulli}(x_n \mid \theta),$$

where θ is a latent probability shared across the 50 data points $\mathbf{x} \in \{0, 1\}^{50}$.

```
from edward.models import Bernoulli, Beta

theta = Beta(a=1.0, b=1.0)
x = Bernoulli(p=tf.ones(50) * theta)
```

Figure 5: Computational graph for a Beta-Bernoulli program.

[7]

[5] https://arxiv.org/abs/1610.09787

[6] https://arxiv.org/pdf/1701.03757.pdf

[7] http://austinrochford.com/resources/talks/dydata-dc-2016-variational-python.slides.html

BAYESIAN MACHINE LEARNING MODELS

What can you model?

- Binary Classification [8]
- Bayesian Deep Learning [9]
- Bayesian Bandits, Contextual Bandits [10]
- Thompson Sampling
- Churn Prediction
- Variational Bayesian Inference, ADVI ...
- Change-point detection [11]
- Hierarchical Bayesian Models [12]

[8] http://blog.booleanbiotech.com/linear_regression_experiments.html

[9] http://bayesiandeeplearning.org/

[10] https://people.orie.cornell.edu/pfrazier/Presentations/2012.10.INFORMS.Bandit.pdf

[11] http://people.duke.edu/~ccc14/sta-663-2016/16C_PyMC3.html

[12] https://github.com/jonsedar/pymc3_vs_pystan/blob/master/40_HierarchicalLinearRegression.ipynb

BAYESIAN MACHINE LEARNING MODELS

What can you model?

- Binary Classification [8]
- Bayesian Deep Learning [9]
- Bayesian Bandits, Contextual Bandits [10]
- Thompson Sampling
- Churn Prediction
- Variational Bayesian Inference, ADVI ...
- Change-point detection [11]
- Hierarchical Bayesian Models [12]

[8] http://blog.booleanbiotech.com/linear_regression_experiments.html

[9] http://bayesiandeeplearning.org/

[10] https://people.orie.cornell.edu/pfrazier/Presentations/2012.10.INFORMS.Bandit.pdf

[11] http://people.duke.edu/~ccc14/sta-663-2016/16C_PyMC3.html

[12] https://github.com/jonsedar/pymc3_vs_pystan/blob/master/40_HierarchicalLinearRegression.ipynb

BAYESIAN MACHINE LEARNING MODELS

What can you model?

- Binary Classification [8]
- Bayesian Deep Learning [9]
- Bayesian Bandits, Contextual Bandits [10]
- Thompson Sampling
- Churn Prediction
- Variational Bayesian Inference, ADVI ...
- Change-point detection [11]
- Hierarchical Bayesian Models [12]

[8] http://blog.booleanbiotech.com/linear_regression_experiments.html

[9] http://bayesiandeeplearning.org/

[10] https://people.orie.cornell.edu/pfrazier/Presentations/2012.10.INFORMS.Bandit.pdf

[11] http://people.duke.edu/~ccc14/sta-663-2016/16C_PyMC3.html

[12] https://github.com/jonsedar/pymc3_vs_pystan/blob/master/40_HierarchicalLinearRegression.ipynb

BAYESIAN MACHINE LEARNING MODELS

What can you model?

- Binary Classification [8]
- Bayesian Deep Learning [9]
- Bayesian Bandits, Contextual Bandits [10]
- Thompson Sampling
- Churn Prediction
- Variational Bayesian Inference, ADVI ...
- Change-point detection [11]
- Hierarchical Bayesian Models [12]

[8] http://blog.booleanbiotech.com/linear_regression_experiments.html

[9] http://bayesiandeeplearning.org/

[10] https://people.orie.cornell.edu/pfrazier/Presentations/2012.10.INFORMS.Bandit.pdf

[11] http://people.duke.edu/~ccc14/sta-663-2016/16C_PyMC3.html

[12] https://github.com/jonsedar/pymc3_vs_pystan/blob/master/40_HierarchicalLinearRegression.ipynb

BAYESIAN MACHINE LEARNING MODELS

What can you model?

- Binary Classification [8]
- Bayesian Deep Learning [9]
- Bayesian Bandits, Contextual Bandits [10]
- Thompson Sampling
- Churn Prediction
- Variational Bayesian Inference, ADVI ...
- Change-point detection [11]
- Hierarchical Bayesian Models [12]

[8] http://blog.booleanbiotech.com/linear_regression_experiments.html

[9] http://bayesiandeeplearning.org/

[10] https://people.orie.cornell.edu/pfrazier/Presentations/2012.10.INFORMS.Bandit.pdf

[11] http://people.duke.edu/~ccc14/sta-663-2016/16C_PyMC3.html

[12] https://github.com/jonsedar/pymc3_vs_pystan/blob/master/40_HierarchicalLinearRegression.ipynb

BAYESIAN MACHINE LEARNING MODELS

What can you model?

- Binary Classification [8]
- Bayesian Deep Learning [9]
- Bayesian Bandits, Contextual Bandits [10]
- Thompson Sampling
- Churn Prediction
- Variational Bayesian Inference, ADVI ...
- Change-point detection [11]
- Hierarchical Bayesian Models [12]

[8] http://blog.booleanbiotech.com/linear_regression_experiments.html

[9] http://bayesiandeeplearning.org/

[10] https://people.orie.cornell.edu/pfrazier/Presentations/2012.10.INFORMS.Bandit.pdf

[11] http://people.duke.edu/~ccc14/sta-663-2016/16C_PyMC3.html

[12] https://github.com/jonsedar/pymc3_vs_pystan/blob/master/40_HierarchicalLinearRegression.ipynb

BAYESIAN MACHINE LEARNING MODELS

What can you model?

- Binary Classification [8]
- Bayesian Deep Learning [9]
- Bayesian Bandits, Contextual Bandits [10]
- Thompson Sampling
- Churn Prediction
- Variational Bayesian Inference, ADVI ...
- Change-point detection [11]
- Hierarchical Bayesian Models [12]

[8] http://blog.booleanbiotech.com/linear_regression_experiments.html

[9] http://bayesiandeeplearning.org/

[10] https://people.orie.cornell.edu/pfrazier/Presentations/2012.10.INFORMS.Bandit.pdf

[11] http://people.duke.edu/~ccc14/sta-663-2016/16C_PyMC3.html

[12] https://github.com/jonsedar/pymc3_vs_pystan/blob/master/40_HierarchicalLinearRegression.ipynb

BAYESIAN MACHINE LEARNING MODELS

What can you model?

- Binary Classification [8]
- Bayesian Deep Learning [9]
- Bayesian Bandits, Contextual Bandits [10]
- Thompson Sampling
- Churn Prediction
- Variational Bayesian Inference, ADVI ...
- Change-point detection [11]
- Hierarchical Bayesian Models [12]

[8] http://blog.booleanbiotech.com/linear_regression_experiments.html

[9] http://bayesiandeeplearning.org/

[10] https://people.orie.cornell.edu/pfrazier/Presentations/2012.10.INFORMS.Bandit.pdf

[11] http://people.duke.edu/~ccc14/sta-663-2016/16C_PyMC3.html

[12] https://github.com/jonsedar/pymc3_vs_pystan/blob/master/40_HierarchicalLinearRegression.ipynb

BAYES' RULE, PRIOR, LIKELIHOOD, POSTERIOR

- Bayes' rule in terms of probabilities of simple events A and B:

$$P(B|A) = \frac{P(A|B)P(B)}{P(A)} = \frac{P(A|B)P(B)}{P(A|B)P(B) + P(A|B^c)P(B^c)}$$

- Frequentist statistics assumes that parameter θ value is fixed. Find the set of parameters under which the data are most likely using MLE.

BAYE'S RULE, PRIOR, LIKELIHOOD, POSTERIOR

Bayes' theorem

- We can replace A and B by model parameters θ and the data y
- Therefore we get $p(\theta|y) = \frac{p(\theta) \times p(y|\theta)}{p(y)}$
- Where:
  - $p(y|\theta)$ ... LIKELIHOOD
  - $p(\theta)$ ... PRIOR DISTRIBUTION=Chosen to reflect prior knowledge we have about the parameter **before** we see any evidence.
  - $p(\theta|y)$ ... POSTERIOR DISTRIBUTION=Only after we see our evidence, this is the main thing we're after here: the distribution of our unknown quantity.
  - $p(y)$ ... a nasty little stuff, an Intractable Integral

BAYE'S RULE, PRIOR, LIKELIHOOD, POSTERIOR

Bayes' theorem

- We can replace A and B by model parameters θ and the data y
- Therefore we get $p(\theta|y) = \frac{p(\theta)\times p(y|\theta)}{p(y)}$
- Where:
  - $p(y|\theta)$ ... LIKELIHOOD
  - $p(\theta)$ ... PRIOR DISTRIBUTION=Chosen to reflect prior knowledge we have about the parameter **before** we see any evidence.
  - $p(\theta|y)$ ... POSTERIOR DISTRIBUTION=Only after we see our evidence, this is the main thing we're after here: the distribution of our unknown quantity.
  - $p(y)$ ... a nasty little stuff, an Intractable Integral

BAYE'S RULE, PRIOR, LIKELIHOOD, POSTERIOR

Bayes' theorem

- We can replace A and B by model parameters θ and the data y
- Therefore we get $p(\theta|y) = \frac{p(\theta) \times p(y|\theta)}{p(y)}$
- Where:
  - $p(y|\theta)$ ... LIKELIHOOD
  - $p(\theta)$ ... PRIOR DISTRIBUTION=Chosen to reflect prior knowledge we have about the parameter **before** we see any evidence.
  - $p(\theta|y)$ ... POSTERIOR DISTRIBUTION=Only after we see our evidence, this is the main thing we're after here: the distribution of our unknown quantity.
  - $p(y)$ ... a nasty little stuff, an Intractable Integral

BAYE'S RULE, PRIOR, LIKELIHOOD, POSTERIOR

Bayes' theorem

- We can replace A and B by model parameters θ and the data y
- Therefore we get $p(\theta|y) = \frac{p(\theta) \times p(y|\theta)}{p(y)}$
- Where:
  - $p(y|\theta)$ ... LIKELIHOOD
  - $p(\theta)$ ... PRIOR DISTRIBUTION=Chosen to reflect prior knowledge we have about the parameter **before** we see any evidence.
  - $p(\theta|y)$ ... POSTERIOR DISTRIBUTION=Only after we see our evidence, this is the main thing we're after here: the distribution of our unknown quantity.
  - $p(y)$ ... a nasty little stuff, an Intractable Integral

THE WELL KNOWN DEBATE

The difference between Frequentist statistics and Bayesian statistics

- Fully Bayesian methods assume that parameter value θ is random.

$$\pi(\theta|y) \quad = \quad \frac{f(y|\theta)\pi(\theta)}{\int f(y|t)\pi(t)\,dt}$$

THE WELL KNOWN DEBATE

The difference between Frequentist statistics and Bayesian statistics

- Fully Bayesian methods assume that parameter value θ is random.

$$\begin{aligned} \pi(\theta|y) &= \frac{f(y|\theta)\pi(\theta)}{\int f(y|t)\pi(t)\,dt} \\ &\propto f(y|\theta)\pi(\theta) \end{aligned}$$

- Reflecting the fact that we conditioning on a random variable θ.
- Mathematically:

$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{evidence}}$$

THE WELL KNOWN DEBATE

The difference between Frequentist statistics and Bayesian statistics

- Fully Bayesian methods assume that parameter value θ is random.

$$\begin{aligned}\pi(\theta|y) &= \frac{f(y|\theta)\pi(\theta)}{\int f(y|t)\pi(t)\,dt} \\ &\propto f(y|\theta)\pi(\theta)\end{aligned}$$

- Reflecting the fact that we conditioning on a random variable θ.
- Mathematically:

$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{evidence}}$$

THE WELL KNOWN DEBATE

The difference between Frequentist statistics and Bayesian statistics

- Fully Bayesian methods assume that parameter value θ is random.

$$\begin{aligned}\pi(\theta|y) &= \frac{f(y|\theta)\pi(\theta)}{\int f(y|t)\pi(t)\,dt} \\ &\propto f(y|\theta)\pi(\theta)\end{aligned}$$

- Reflecting the fact that we conditioning on a random variable θ.
- Mathematically:

$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{evidence}}$$

THE WELL KNOWN DEBATE

The difference between Frequentist statistics and Bayesian statistics

- Fully Bayesian methods assume that parameter value θ is random.

$$\begin{aligned}\pi(\theta|y) &= \frac{f(y|\theta)\pi(\theta)}{\int f(y|t)\pi(t)\,dt} \\ &\propto f(y|\theta)\pi(\theta)\end{aligned}$$

- Reflecting the fact that we conditioning on a random variable θ.
- Mathematically:

$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{evidence}}$$

Why become a Bayesian? THE BETA BINOMIAL MODEL

- As a data-scientist, you are working on data related to a Real Time Bidding (RTB) system. An SSP (Supply Side Platform), sends you, the bidder, n bid requests, y of which you have won. Your parameter of interest is θ,the proportion of bid requests that you win. Sample likelihood follows a Binomial distribution: $Y|\theta \sim \text{Binomial}(n, \theta)$
- We use a Beta distribution for the prior: $\theta \sim \text{Beta}(\alpha, \beta)$
- What is the posterior distribution?
- How do you make predictions using the posterior distribution?. Imagine that we have m additional bids. What is the probability that exactly j of these we win?

Why become a Bayesian? THE BETA BINOMIAL MODEL

- As a data-scientist, you are working on data related to a Real Time Bidding (RTB) system. An SSP (Supply Side Platform), sends you, the bidder, n bid requests, y of which you have won. Your parameter of interest is θ,the proportion of bid requests that you win. Sample likelihood follows a Binomial distribution: $Y|\theta \sim \text{Binomial}(n, \theta)$
- We use a Beta distribution for the prior: $\theta \sim \text{Beta}(\alpha, \beta)$
- What is the posterior distribution?
- How do you make predictions using the posterior distribution?. Imagine that we have m additional bids. What is the probability that exactly j of these we win?

Why become a Bayesian? THE BETA BINOMIAL MODEL

- As a data-scientist, you are working on data related to a Real Time Bidding (RTB) system. An SSP (Supply Side Platform), sends you, the bidder, n bid requests, y of which you have won. Your parameter of interest is θ,the proportion of bid requests that you win. Sample likelihood follows a Binomial distribution: $Y|\theta \sim \text{Binomial}(n, \theta)$
- We use a Beta distribution for the prior: $\theta \sim \text{Beta}(\alpha, \beta)$
- What is the posterior distribution?
- How do you make predictions using the posterior distribution?. Imagine that we have m additional bids. What is the probability that exactly j of these we win?

Why become a Bayesian? THE BETA BINOMIAL MODEL

- As a data-scientist, you are working on data related to a Real Time Bidding (RTB) system. An SSP (Supply Side Platform), sends you, the bidder, n bid requests, y of which you have won. Your parameter of interest is θ,the proportion of bid requests that you win. Sample likelihood follows a Binomial distribution: $Y|\theta \sim \text{Binomial}(n, \theta)$
- We use a Beta distribution for the prior: $\theta \sim \text{Beta}(\alpha, \beta)$
- What is the posterior distribution?
- How do you make predictions using the posterior distribution?. Imagine that we have m additional bids. What is the probability that exactly j of these we win?

THE BETA BINOMIAL MODEL

- The Beta distribution is a Conjugate prior to binomial likelihood and Uninformative if $a = b = 1 a = b = 1$ (exactly like a uniform distribution). The parameters for this distribution are α and β.

$$g(\theta; \alpha, \beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \theta^{\alpha-1}(1-\theta)^{\beta-1}$$

- Therefore, the posterior distribution:

$$\begin{aligned} h(\theta|\mathbf{y}) &\propto f(\mathbf{y}|\theta)g(\theta) \\ &= \left[\binom{n}{y}\theta^{y}(1-\theta)^{n-y}\right]\left[\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}p^{\alpha-1}(1-\theta)^{\beta-1}\right] \\ &\propto \theta^{y}(1-\theta)^{n-y}\theta^{\alpha-1}(1-\theta)^{\beta-1} \\ &= \theta^{\alpha+y-1}(1-\theta)^{n+\beta-y-1} \end{aligned}$$

Is just another beta distribution:

THE BETA BINOMIAL MODEL

- The Beta distribution is a Conjugate prior to binomial likelihood and Uninformative if $a = b = 1a = b = 1$ (exactly like a uniform distribution). The parameters for this distribution are α and β.

$$g(\theta; \alpha, \beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\theta^{\alpha-1}(1-\theta)^{\beta-1}$$

- Therefore, the posterior distribution:

$$\begin{aligned} h(\theta|\mathbf{y}) &\propto f(\mathbf{y}|\theta)g(\theta) \\ &= \left[\binom{n}{y}\theta^{y}(1-\theta)^{n-y}\right]\left[\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}p^{\alpha-1}(1-\theta)^{\beta-1}\right] \\ &\propto \theta^{y}(1-\theta)^{n-y}\theta^{\alpha-1}(1-\theta)^{\beta-1} \\ &= \theta^{\alpha+y-1}(1-\theta)^{n+\beta-y-1} \end{aligned}$$

Is just another beta distribution:

THE BETA BINOMIAL MODEL:Beta prior

$$\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\theta^{\alpha-1}(1-\theta)^{\beta-1}$$

- It is supported on $[0, 1]$.

THE BETA BINOMIAL MODEL:Beta prior

$$\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\theta^{\alpha-1}(1-\theta)^{\beta-1}$$

- It is supported on $[0,1]$.
- The Expectation is $E(\theta) = \frac{\alpha}{\alpha+\beta}$

THE BETA BINOMIAL MODEL:Beta prior

$\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\theta^{\alpha-1}(1-\theta)^{\beta-1}$

- It is supported on $[0,1]$.
- The Expectation is $E(\theta)=\frac{\alpha}{\alpha+\beta}$
- The Variance is $Var(\theta)=\frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}$.

THE BETA BINOMIAL MODEL:Beta prior

$$\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\theta^{\alpha-1}(1-\theta)^{\beta-1}$$

- It is supported on $[0,1]$.
- The Expectation is $E(\theta)=\frac{\alpha}{\alpha+\beta}$
- The Variance is $Var(\theta)=\frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}$.
- It can assume many shapes depending on α and β.

THE BETA BINOMIAL MODEL:Beta prior

$\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\theta^{\alpha-1}(1-\theta)^{\beta-1}$

- It is supported on $[0,1]$.
- The Expectation is $E(\theta) = \frac{\alpha}{\alpha+\beta}$
- The Variance is $Var(\theta) = \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}$.
- It can assume many shapes depending on α and β.
- Special case when $\alpha = \beta = 1$, it's uniform.

THE BETA BINOMIAL MODEL:Beta prior

$$\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\theta^{\alpha-1}(1-\theta)^{\beta-1}$$

- It is supported on $[0,1]$.
- The Expectation is $E(\theta) = \frac{\alpha}{\alpha+\beta}$
- The Variance is $Var(\theta) = \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}$.
- It can assume many shapes depending on α and β.
- Special case when $\alpha = \beta = 1$, it's uniform.

THE BETA BINOMIAL MODEL: Posteriors are betas

- The posterior distribution is from the sample family as the prior distribution. This is a very special case; we say that the prior is Conjugate to the data distribution.
  - The Prior was Beta(α, β).

THE BETA BINOMIAL MODEL: Posteriors are betas

- The posterior distribution is from the sample family as the prior distribution. This is a very special case; we say that the prior is Conjugate to the data distribution.
  - The Prior was Beta(α, β).
  - The Posterior is Beta as well(α', β').

THE BETA BINOMIAL MODEL: Posteriors are betas

- The posterior distribution is from the sample family as the prior distribution. This is a very special case; we say that the prior is Conjugate to the data distribution.
  - The Prior was Beta(α, β).
  - The Posterior is Beta as well(α', β').
  - The Prior and posterior have the same family of distributions.

THE BETA BINOMIAL MODEL: Posteriors are betas

- The posterior distribution is from the sample family as the prior distribution. This is a very special case; we say that the prior is Conjugate to the data distribution.
  - The Prior was Beta(α, β).
  - The Posterior is Beta as well(α', β').
  - The Prior and posterior have the same family of distributions.
  - The Beta is a *conjugate prior* for the Bernoulli model.

THE BETA BINOMIAL MODEL: Posteriors are betas

- The posterior distribution is from the sample family as the prior distribution. This is a very special case; we say that the prior is Conjugate to the data distribution.
  - The Prior was Beta(α, β).
  - The Posterior is Beta as well(α', β').
  - The Prior and posterior have the same family of distributions.
  - The Beta is a *conjugate prior* for the Bernoulli model.
  - Posterior was obtained by inspection.

THE BETA BINOMIAL MODEL: Posteriors are betas

- The posterior distribution is from the sample family as the prior distribution. This is a very special case; we say that the prior is Conjugate to the data distribution.
  - The Prior was Beta(α, β).
  - The Posterior is Beta as well(α', β').
  - The Prior and posterior have the same family of distributions.
  - The Beta is a *conjugate prior* for the Bernoulli model.
  - Posterior was obtained by inspection.
  - Conjugate priors are very convenient.

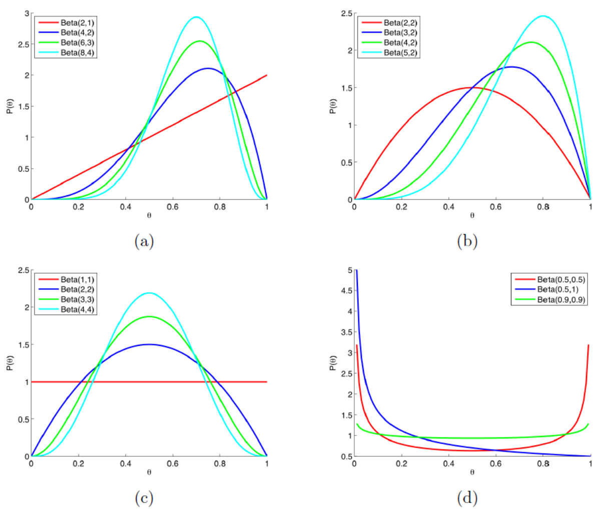
THE BETA BINOMIAL MODEL: Posteriors are betas

- The posterior distribution is from the sample family as the prior distribution. This is a very special case; we say that the prior is Conjugate to the data distribution.
  - The Prior was Beta(α, β).
  - The Posterior is Beta as well(α', β').
  - The Prior and posterior have the same family of distributions.
  - The Beta is a *conjugate prior* for the Bernoulli model.
  - Posterior was obtained by inspection.
  - Conjugate priors are very convenient.
  - There are conjugate priors for many models.

THE BETA BINOMIAL MODEL: Posteriors are betas

- The posterior distribution is from the sample family as the prior distribution. This is a very special case; we say that the prior is Conjugate to the data distribution.
  - The Prior was Beta(α, β).
  - The Posterior is Beta as well(α', β').
  - The Prior and posterior have the same family of distributions.
  - The Beta is a *conjugate prior* for the Bernoulli model.
  - Posterior was obtained by inspection.
  - Conjugate priors are very convenient.
  - There are conjugate priors for many models.
  - There are also important models for which conjugate priors do not exist.
- The general name for the expected distribution over future observations is the posterior predictive distribution- we make predictions using this distribution. That is, if the prior distribution is beta and the likelihood is binomial,

THE BETA BINOMIAL MODEL: Priors



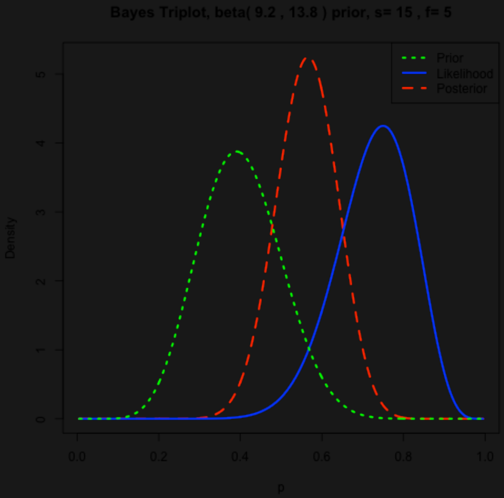
(a)

(b)

(c)

(d)

THE BETA BINOMIAL MODEL: Posterior update



JUST A LITTLE PYTHON CODE SNIPPET:PyMC3

Common probabilistic programming tools

JUST A LITTLE PYTHON CODE SNIPPET:PyMC3

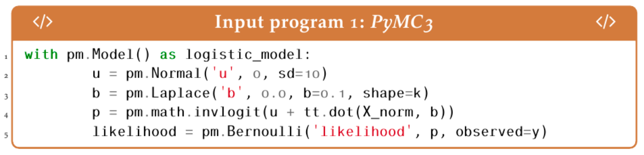
Common probabilistic programming tools

- Define The Model

JUST A LITTLE PYTHON CODE SNIPPET:PyMC3

Common probabilistic programming tools

- Define The Model



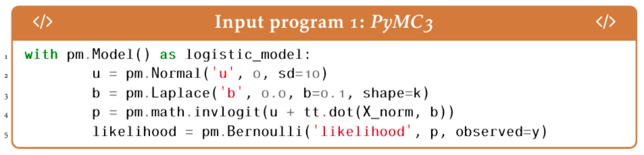
</> Input program 1: *PyMC3* </>

```
with pm.Model() as logistic_model:
        u = pm.Normal('u', 0, sd=10)
        b = pm.Laplace('b', 0.0, b=0.1, shape=k)
        p = pm.math.invlogit(u + tt.dot(X_norm, b))
        likelihood = pm.Bernoulli('likelihood', p, observed=y)
```

JUST A LITTLE PYTHON CODE SNIPPET:PyMC3

Common probabilistic programming tools

- Define The Model



</> **Input program 1:** ***PyMC3*** </>

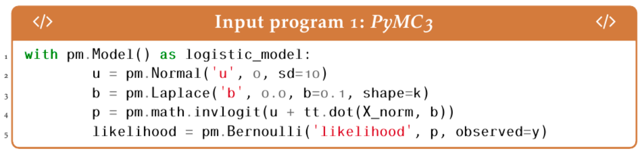
```
with pm.Model() as logistic_model:
        u = pm.Normal('u', 0, sd=10)
        b = pm.Laplace('b', 0.0, b=0.1, shape=k)
        p = pm.math.invlogit(u + tt.dot(X_norm, b))
        likelihood = pm.Bernoulli('likelihood', p, observed=y)
```

- Run The Model

JUST A LITTLE PYTHON CODE SNIPPET:PyMC3

Common probabilistic programming tools

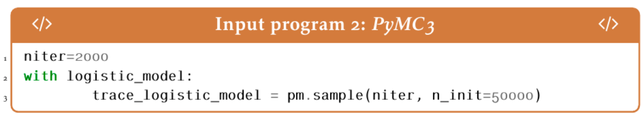
- Define The Model



Input program 1: ***PyMC3***

```
with pm.Model() as logistic_model:
        u = pm.Normal('u', 0, sd=10)
        b = pm.Laplace('b', 0.0, b=0.1, shape=k)
        p = pm.math.invlogit(u + tt.dot(X_norm, b))
        likelihood = pm.Bernoulli('likelihood', p, observed=y)
```

- Run The Model



Input program 2: ***PyMC3***

```
niter=2000
with logistic_model:
        trace_logistic_model = pm.sample(niter, n_init=50000)
```

BAYESIAN LOGISTIC REGRESSION

The choice of the Bayesian Prior dictates if it is Lasso OR Ridge [13]

- Regularized regression methods can provide a better model fit by including a penalization parameter in the cost function. The cost function seeks a parameter (θ) to minimize the sum of squared errors ($J(\theta)$),

$$J(\theta) = \sum_{i=1}^{m} (y_i - \hat{y}_i)^2$$

- Parameter estimates can become inflated when the model over fits the data or there is collinearity. The magnitude of the estimates can be controlled by introducing a regularization term λ.

[13] https://www.ariddell.org/horseshoe-prior-with-stan.html

BAYESIAN LOGISTIC REGRESSION

The choice of the Bayesian Prior dictates if it is Lasso OR Ridge [13]

- Regularized regression methods can provide a better model fit by including a penalization parameter in the cost function. The cost function seeks a parameter (θ) to minimize the sum of squared errors $(J(\theta))$,

$$J(\theta) = \sum_{i=1}^{m} (y_i - \hat{y}_i)^2$$

- Parameter estimates can become inflated when the model over fits the data or there is collinearity. The magnitude of the estimates can be controlled by introducing a regularization term λ.

[13] https://www.ariddell.org/horseshoe-prior-with-stan.html

BAYESIAN LOGISTIC REGRESSION

The choice of the Bayesian Prior dictates if it is Lasso OR Ridge [13]

- Regularized regression methods can provide a better model fit by including a penalization parameter in the cost function. The cost function seeks a parameter (θ) to minimize the sum of squared errors $(J(\theta))$,

$$J(\theta) = \sum_{i=1}^{m} (y_i - \hat{y}_i)^2$$

- Parameter estimates can become inflated when the model over fits the data or there is collinearity. The magnitude of the estimates can be controlled by introducing a regularization term λ.

[13] https://www.ariddell.org/horseshoe-prior-with-stan.html

BAYESIAN LOGISTIC REGRESSION:LASSO

LASSO (Least Absolute Shrinkage and Selection Operator) regression

- Using the absolute value penalty:

$$J(\theta) = \sum_{i=1}^{m} (y_i - \hat{y}i)^2 + \lambda \sum_{i=1}^{n} |\theta_j|$$

- In LASSO regression some coefficients are set to exactly zero. From a Bayesian perspective, this is equivalent to assigning a zero-mean $\mu \sim$ Laplace(λ) distribution $=$

$$f(x; \mu, \theta) = \frac{1}{2\theta} \exp\left(-\frac{|x - \mu|}{\theta} \right)$$

 on the parameter vector. Because LASSO regression sets some coefficients to exactly zero, it is sometimes used to conduct feature selection.

BAYESIAN LOGISTIC REGRESSION:LASSO

LASSO (Least Absolute Shrinkage and Selection Operator) regression

- Using the absolute value penalty:

$$J(\theta) = \sum_{i=1}^{m} (y_i - \hat{y}i)^2 + \lambda \sum_{i=1}^{n} |\theta_j|$$

- In LASSO regression some coefficients are set to exactly zero. From a Bayesian perspective, this is equivalent to assigning a zero-mean $\mu \sim \text{Laplace}(\lambda)$ distribution $=$

$$f(x; \mu, \theta) = \frac{1}{2\theta} \exp\left(-\frac{|x - \mu|}{\theta}\right)$$

  on the parameter vector. Because LASSO regression sets some coefficients to exactly zero, it is sometimes used to conduct feature selection.

BAYESIAN LOGISTIC REGRESSION:LASSO

LASSO (Least Absolute Shrinkage and Selection Operator) regression

- Using the absolute value penalty:

$$J(\theta) = \sum_{i=1}^{m} (y_i - \hat{y}i)^2 + \lambda \sum_{i=1}^{n} |\theta_j|$$

- In LASSO regression some coefficients are set to exactly zero. From a Bayesian perspective, this is equivalent to assigning a zero-mean $\mu \sim$ Laplace(λ) distribution $=$

$$f(x; \mu, \theta) = \frac{1}{2\theta} \exp\left(-\frac{|x - \mu|}{\theta}\right)$$

  on the parameter vector. Because LASSO regression sets some coefficients to exactly zero, it is sometimes used to conduct feature selection.

BAYESIAN LOGISTIC REGRESSION:RIDGE

Ridge

- The cost function for ridge regression:

$$J(\theta) = \sum_{i=1}^{m}(y_i - \hat{y}i)^2 + \lambda \sum_{i=1}^{n} \theta_j^2$$

- Including the regularization parameter effectively shrinks some coefficients towards zero. In squared penalty, coefficients never actually reach exactly zero. λ is used to control the bias-variance trade off. From a Bayesian perspective, this is equivalent to assigning a normally distributed prior.
- Finally, elastic net is a combination of ridge and LASSO regression:
  $J(\theta) = \sum_{i=1}^{m}(y_i - \hat{y}i)^2 + \lambda \sum_{i=1}^{n} \theta_j^2 + \lambda \sum_{i=1}^{n} |\theta_j|$

BAYESIAN LOGISTIC REGRESSION:RIDGE

Ridge

- The cost function for ridge regression:

$$J(\theta) = \sum_{i=1}^{m} (y_i - \hat{y}i)^2 + \lambda \sum_{i=1}^{n} \theta_j^2$$

- Including the regularization parameter effectively shrinks some coefficients towards zero. In squared penalty, coefficients never actually reach exactly zero. λ is used to control the bias-variance trade off. From a Bayesian perspective, this is equivalent to assigning a normally distributed prior.
- Finally, elastic net is a combination of ridge and LASSO regression:
  $J(\theta) = \sum_{i=1}^{m} (y_i - \hat{y}i)^2 + \lambda \sum_{i=1}^{n} \theta_j^2 + \lambda \sum_{i=1}^{n} |\theta_j|$

BAYESIAN LOGISTIC REGRESSION:RIDGE

Ridge

- The cost function for ridge regression:

$$J(\theta) = \sum_{i=1}^{m}(y_i - \hat{y}i)^2 + \lambda \sum_{i=1}^{n} \theta_j^2$$

- Including the regularization parameter effectively shrinks some coefficients towards zero. In squared penalty, coefficients never actually reach exactly zero. λ is used to control the bias-variance trade off. From a Bayesian perspective, this is equivalent to assigning a normally distributed prior.
- Finally, elastic net is a combination of ridge and LASSO regression:
  $J(\theta) = \sum_{i=1}^{m}(y_i - \hat{y}i)^2 + \lambda \sum_{i=1}^{n} \theta_j^2 + \lambda \sum_{i=1}^{n} |\theta_j|$

REFERENCES

- Edward: edwardlib.org
- PyMC3: http://pymc-devs.github.io/pymc3/
- Probabilistic Programming and Bayesian Methods for Hackers: `https://github.com/CamDavidsonPilon/Probabilistic-Programming-and-Bayesian-Methods-for-`
- Gelman, Andrew, et al. Bayesian data analysis. https://www.amazon.com/Bayesian-Analysis-Chapman-Statistical-Science/dp/B00I60M6H6

February 2017