NeuroBayes: Marrying Deep Learning and Bayesian framework

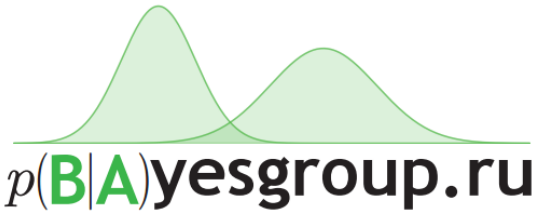
Dmitry P. Vetrov

Research professor in HSE

Leading researcher at Yandex

Head of Bayesian methods research group

http://bayesgroup.ru



Deep Learning

- Revolution in machine learning
- Deep neural networks approach to human intelligence on a number of problems
- May solve quite non-standard problems such as image2caption and artistic style tranfer

A woman is throwing a frisbee in a park.

A little girl sitting on a bed with a teddy bear.

Reasons of DNN success

- Size really matters: the bigger data the better
- Effective large-scale optimization algorithms
- New regularization techniques

Reasons of DNN success

- Size really matters: the bigger data the better
- Effective large-scale optimization algorithms
- New regularization techniques

WORK IN PROGRESS

- Memory networks
- Deep reinforcement learning
- One-shot learning (transfer learning)

Bayesian framework

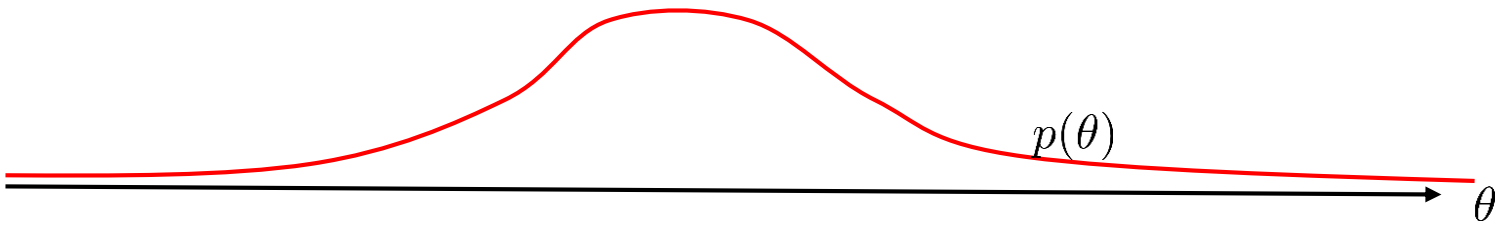
- Treats everything as random variables
- Allow to encode our ignorance in terms of distributions
- Makes use of **Bayes theorem**

$$p(\theta|X) = \frac{p(X|\theta)p(\theta)}{\int p(X|\theta)p(\theta)d\theta} \qquad \texttt{Posterior} = \frac{\texttt{Likelihood} \times \texttt{Prior}}{\texttt{Evidence}}$$

Bayesian framework

- Treats everything as random variables
- Allow to encode our ignorance in terms of distributions
- Makes use of **Bayes theorem**

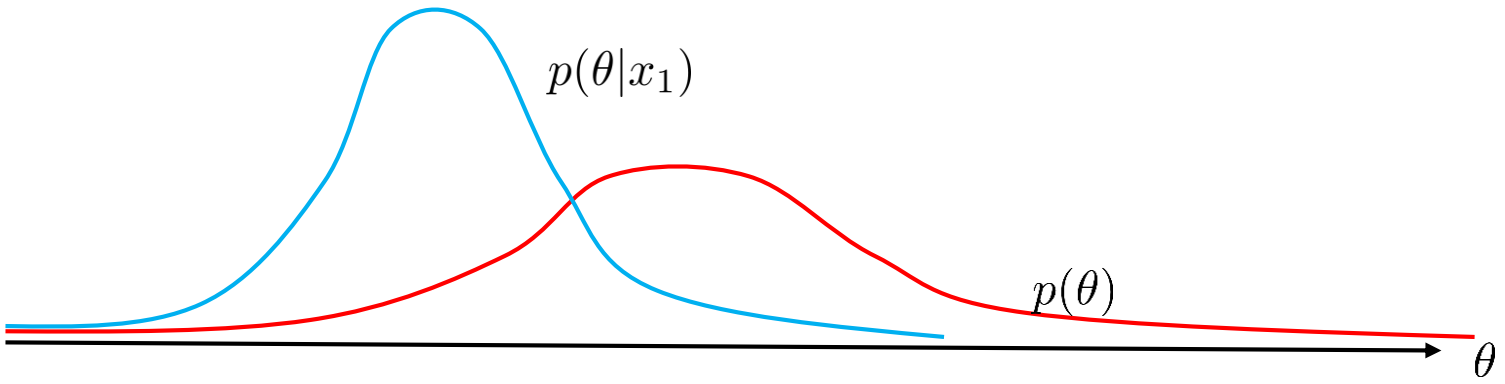
$$p(\theta|X) = \frac{p(X|\theta)p(\theta)}{\int p(X|\theta)p(\theta)d\theta} \qquad \texttt{Posterior} = \frac{\texttt{Likelihood} \times \texttt{Prior}}{\texttt{Evidence}}$$



Bayesian framework

- Treats everything as random variables
- Allow to encode our ignorance in terms of distributions
- Makes use of **Bayes theorem**

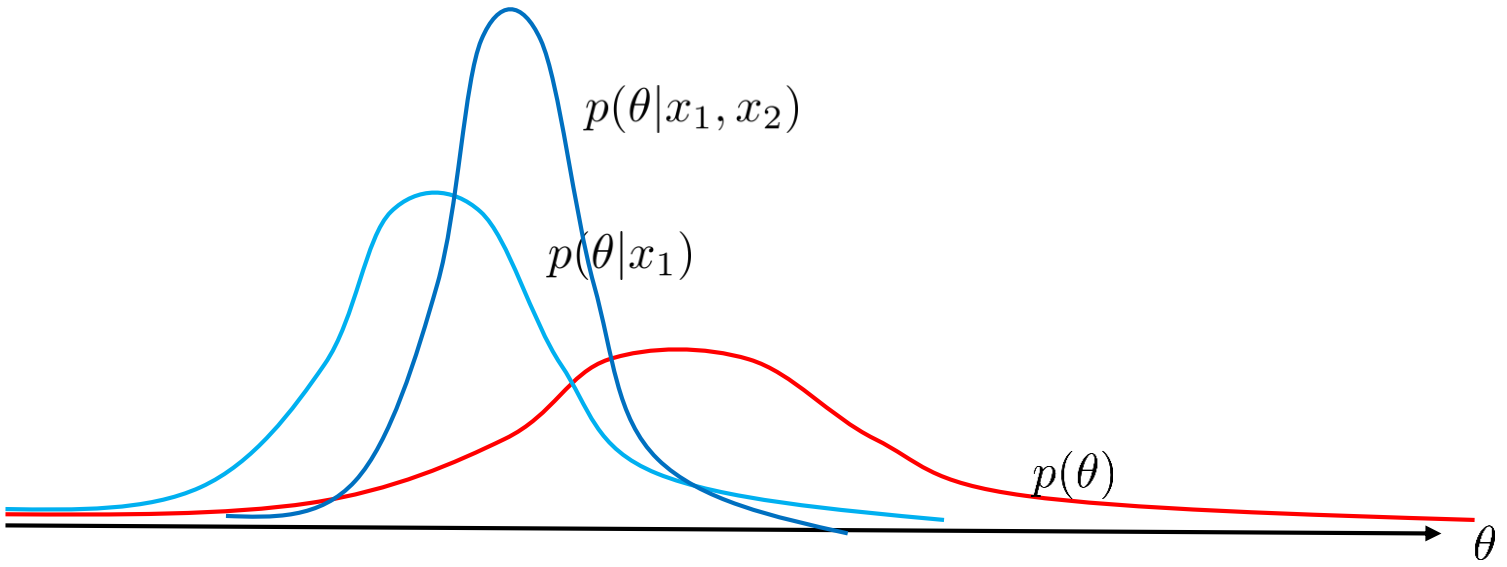
$$p(\theta|X) = \frac{p(X|\theta)p(\theta)}{\int p(X|\theta)p(\theta)d\theta} \qquad \texttt{Posterior} = \frac{\texttt{Likelihood} \times \texttt{Prior}}{\texttt{Evidence}}$$



Bayesian framework

- Treats everything as random variables
- Allow to encode our ignorance in terms of distributions
- Makes use of **Bayes theorem**

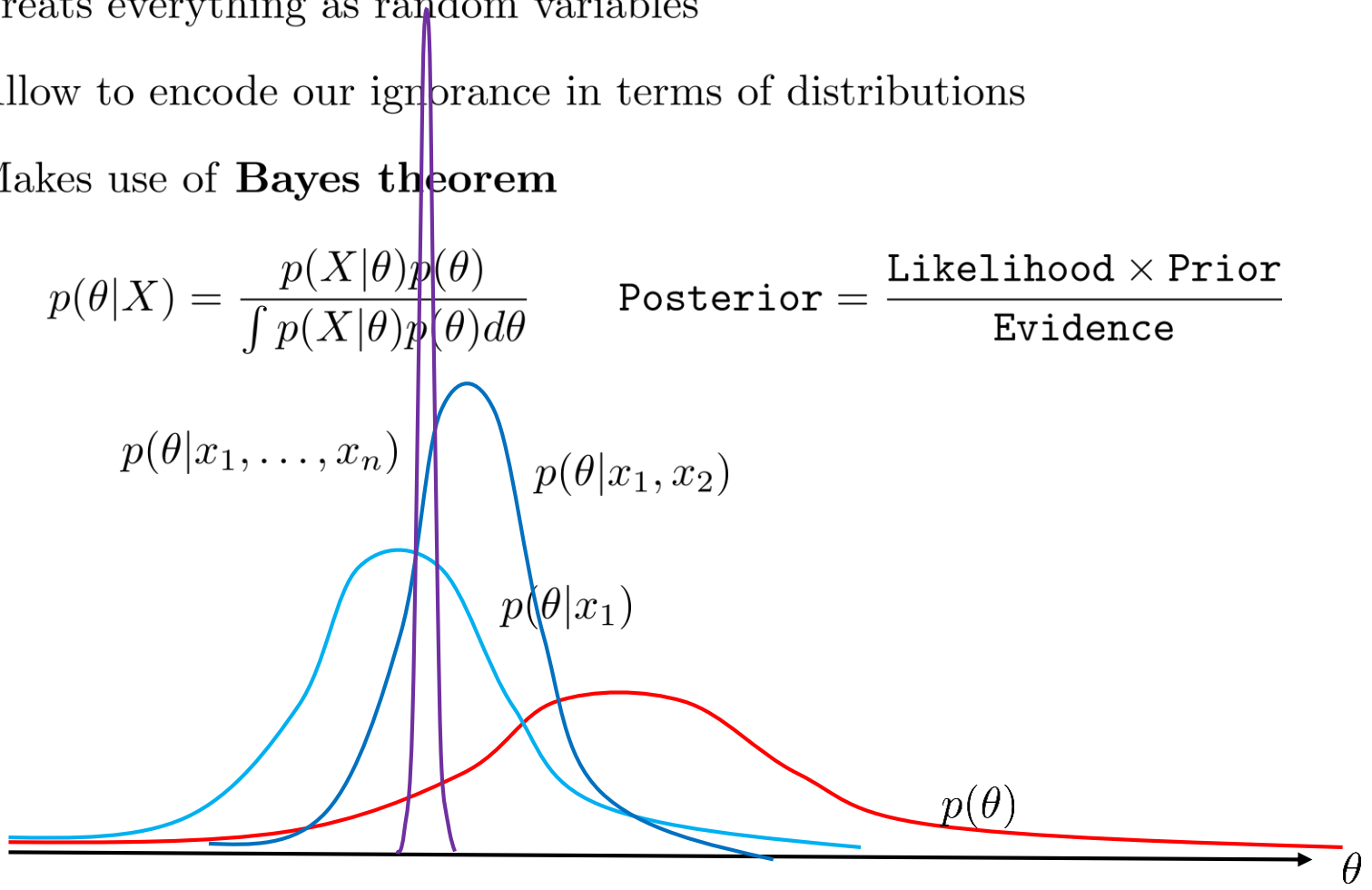
$$p(\theta|X) = \frac{p(X|\theta)p(\theta)}{\int p(X|\theta)p(\theta)d\theta} \qquad \texttt{Posterior} = \frac{\texttt{Likelihood} \times \texttt{Prior}}{\texttt{Evidence}}$$



Bayesian framework

- Treats everything as random variables
- Allow to encode our ignorance in terms of distributions
- Makes use of **Bayes theorem**

$$p(\theta|X) = \frac{p(X|\theta)p(\theta)}{\int p(X|\theta)p(\theta)d\theta} \qquad \texttt{Posterior} = \frac{\texttt{Likelihood} \times \texttt{Prior}}{\texttt{Evidence}}$$



Bayesian framework

- Treats everything as random variables
- Allow to encode our ignorance in terms of distributions
- Makes use of **Bayes theorem**

$$p(\theta|X) = \frac{p(X|\theta)p(\theta)}{\int p(X|\theta)p(\theta)d\theta} \qquad \texttt{Posterior} = \frac{\texttt{Likelihood} \times \texttt{Prior}}{\texttt{Evidence}}$$

Bayesian framework

- Treats everything as random variables
- Allow to encode our ignorance in terms of distributions
- Makes use of **Bayes theorem**

$$p(\theta|X) = \frac{p(X|\theta)p(\theta)}{\int p(X|\theta)p(\theta)d\theta} \qquad \texttt{Posterior} = \frac{\texttt{Likelihood} \times \texttt{Prior}}{\texttt{Evidence}}$$

Can only be computed for simple models

Advantages of Bayesian framework

- Regularization

  Prevents overfitting on the training data because prior does not allow to tune parameters too much

Advantages of Bayesian framework

- Regularization

  Prevents overfitting on the training data because prior does not allow to tune parameters too much

- Extensibility

  Bayesian inference results to posterior which can be now used as prior in next model

Advantages of Bayesian framework

- Regularization

  Prevents overfitting on the training data because prior does not allow to tune parameters too much

- Extensibility

  Bayesian inference results to posterior which can be now used as prior in next model

- Latent variable modelling

  Establishing additional latent variables one may restore mixture of distributions and/or learn better representations

Advantages of Bayesian framework

- Regularization

  Prevents overfitting on the training data because prior does not allow to tune parameters too much

- Extensibility

  Bayesian inference results to posterior which can be now used as prior in next model

- Latent variable modelling

  Establishing additional latent variables one may restore mixture of distributions and/or learn better representations

- Scalability

  Stochastic variational inference allows to approximate posteriors using deep neural networks

Example: representation learning

- In well-known *word2vec* model any word is given its vector representation
- Surprizingly algebraic operations over those representations lead to logical operations over their meanings

v(‘London’) – v(‘UK’) + v(‘Australia’) = v(‘Canberra’)

v(‘King’) – v(‘Man’) + v(‘Woman’) = v(‘Queen’)

Example: representation learning

- In well-known *word2vec* model any word is given its vector representation
- Surprizingly algebraic operations over those representations lead to logical operations over their meanings

v(‘London’) – v(‘UK’) + v(‘Australia’) = v(‘Canberra’)

v(‘King’) – v(‘Man’) + v(‘Woman’) = v(‘Queen’)

v(‘Putin’) – v(‘Russia’) + v(‘France’) = v(?)

Example: representation learning

- In well-known *word2vec* model any word is given its vector representation
- Surprizingly algebraic operations over those representations lead to logical operations over their meanings

v(‘London’) – v(‘UK’) + v(‘Australia’) = v(‘Canberra’)

v(‘King’) – v(‘Man’) + v(‘Woman’) = v(‘Queen’)

v(‘Putin’) – v(‘Russia’) + v(‘France’) = v(?) // Hollande or Sarkozy???

Our research

- Multi-sense extension of word2vec
- Compression of DNN using tensor calculus and Bayesian dropout
- Acceleration of DNN
- One-shot learning
- Improving variational auto-enocoders

Industrial partners

- Yandex
- Schlumberger
- Sberbank
- Samsung
- Kaspersky lab