Probabilistic Machine Learning and Bayesian Modeling

Piyush Rai

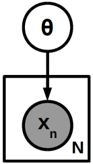
SIGML-IITK: Machine Learning Research Day

August 28, 2016

(Probabilistic) Machine Learning

- Assume data $\mathbf{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ generated from a probabilistic model:

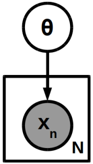
$$\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N \sim p(\boldsymbol{x}|\theta)$$



(Probabilistic) Machine Learning

- Assume data $\mathbf{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ generated from a probabilistic model:

$$\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N \sim p(\boldsymbol{x}|\theta)$$

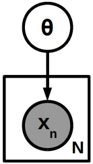


- Goal: To estimate the model and its parameters θ (and make predictions)

(Probabilistic) Machine Learning

- Assume data $\mathbf{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ generated from a probabilistic model:

$$\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N \sim p(\boldsymbol{x}|\theta)$$

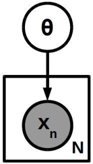


- Goal: To estimate the model and its parameters θ (and make predictions)
- What data looks like: Modeled by a likelihood function $p(\boldsymbol{x}|\theta)$
  - Measures data fit (or "loss") w.r.t. the given parameter θ

(Probabilistic) Machine Learning

- Assume data $\mathbf{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ generated from a probabilistic model:

$$\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N \sim p(\boldsymbol{x}|\theta)$$



- Goal: To estimate the model and its parameters θ (and make predictions)
- What data looks like: Modeled by a likelihood function $p(\boldsymbol{x}|\theta)$
  - Measures data fit (or "loss") w.r.t. the given parameter θ
- What parameters look like: Modeled by a prior distribution $p(\theta)$
  - Also corresponds to imposing a regularizer over θ

Optimization vs Inference

Learning as Optimization

Learning as (Bayesian) Inference

Optimization vs Inference

Learning as Optimization

- Parameter θ is a fixed unknown

Learning as (Bayesian) Inference

Optimization vs Inference

Learning as Optimization

- Parameter θ is a fixed unknown
- Find a point estimate (a single best answer) for θ by minimize a "loss"

Learning as (Bayesian) Inference

Optimization vs Inference

Learning as Optimization

- Parameter θ is a fixed unknown
- Find a point estimate (a single best answer) for θ by minimize a "loss"

$$\hat{\theta} = \arg\min_{\theta \in \Theta} \text{Loss}(\mathbf{X}; \theta)$$

Learning as (Bayesian) Inference

Optimization vs Inference

Learning as Optimization

- Parameter θ is a fixed unknown
- Find a point estimate (a single best answer) for θ by minimize a "loss"

$$\hat{\theta} = \arg\min_{\theta \in \Theta} \text{Loss}(\mathbf{X}; \theta) \quad \underline{\text{or}} \quad \underbrace{\arg\max_{\theta} \log p(\mathbf{X}|\theta)}_{\text{Maximum Likelihood}}$$

Learning as (Bayesian) Inference

Optimization vs Inference

Learning as Optimization

- Parameter θ is a fixed unknown
- Find a point estimate (a single best answer) for θ by minimize a "loss"

$$\hat{\theta} = \arg\min_{\theta \in \Theta} \text{Loss}(\mathbf{X};\theta) \quad \underline{\text{or}} \quad \underbrace{\arg\max_{\theta} \log p(\mathbf{X}|\theta)}_{\text{Maximum Likelihood}} \quad \underline{\text{or}} \quad \underbrace{\arg\max_{\theta}[\log p(\mathbf{X}|\theta) + \log p(\theta)]}_{\text{Maximum-a-Posteriori Estimation}}$$

Learning as (Bayesian) Inference

Optimization vs Inference

Learning as Optimization

- Parameter θ is a fixed unknown
- Find a point estimate (a single best answer) for θ by minimize a "loss"

$$\hat{\theta} = \arg\min_{\theta \in \Theta} \text{Loss}(\mathbf{X}; \theta) \quad \underline{\text{or}} \quad \underbrace{\arg\max_{\theta} \log p(\mathbf{X}|\theta)}_{\text{Maximum Likelihood}} \quad \underline{\text{or}} \quad \underbrace{\arg\max_{\theta} [\log p(\mathbf{X}|\theta) + \log p(\theta)]}_{\text{Maximum-a-Posteriori Estimation}}$$

Learning as (Bayesian) Inference

- Treat the parameter θ as a random variable with a prior distribution $p(\theta)$

Optimization vs Inference

Learning as Optimization

- Parameter θ is a fixed unknown
- Find a point estimate (a single best answer) for θ by minimize a "loss"

$$\hat{\theta} = \arg\min_{\theta \in \Theta} \text{Loss}(\mathbf{X};\theta) \quad \underline{\text{or}} \quad \underbrace{\arg\max_{\theta} \log p(\mathbf{X}|\theta)}_{\text{Maximum Likelihood}} \quad \underline{\text{or}} \quad \underbrace{\arg\max_{\theta}[\log p(\mathbf{X}|\theta) + \log p(\theta)]}_{\text{Maximum-a-Posteriori Estimation}}$$

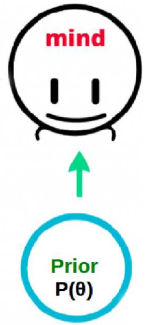
Learning as (Bayesian) Inference

- Treat the parameter θ as a random variable with a prior distribution $p(\theta)$
- Infer the full posterior distribution over the parameters using **Bayes rule**

$$p(\theta|\mathbf{X}) = \frac{p(\mathbf{X}|\theta)p(\theta)}{p(\mathbf{X})} \propto \text{Likelihood} \times \text{Prior}$$

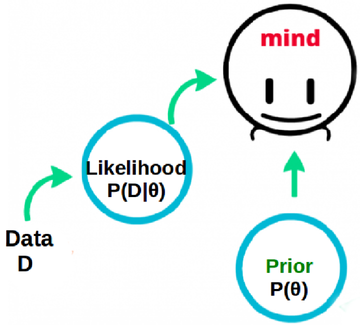
Bayesian Learning

- Bayesian inference is used to update the prior (our prior belief about θ) and gives the posterior



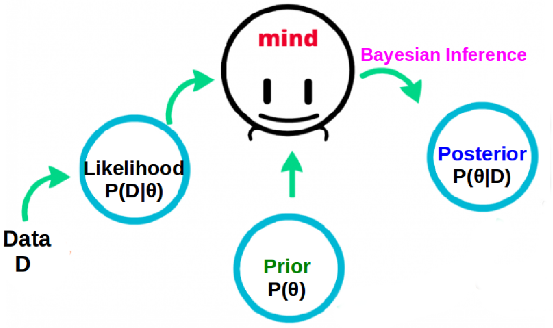
Bayesian Learning

- Bayesian inference is used to update the prior (our prior belief about θ) and gives the posterior



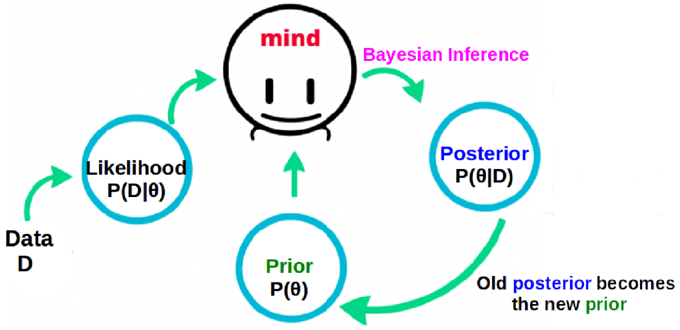
Bayesian Learning

- Bayesian inference is used to update the prior (our prior belief about θ) and gives the posterior



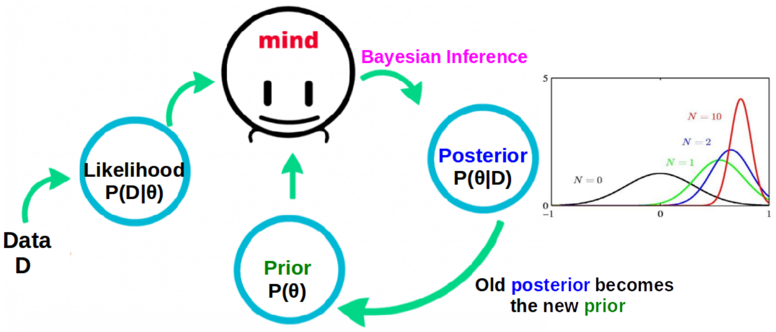
Bayesian Learning

- Bayesian inference is used to update the prior (our prior belief about θ) and gives the posterior



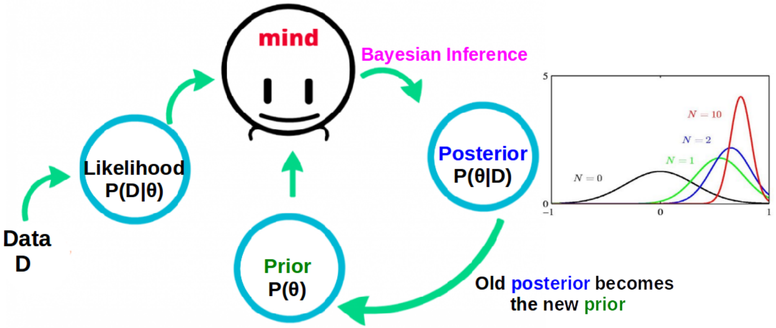
Bayesian Learning

- Bayesian inference is used to update the prior (our prior belief about θ) and gives the posterior



Bayesian Learning

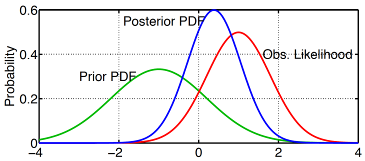
- Bayesian inference is used to update the prior (our prior belief about θ) and gives the posterior



A naturally "online" learning setting

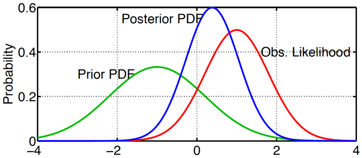
Why be Bayesian?

- Can model the uncertainty in θ via the full parameter posterior $p(\theta|\mathbf{X})$



Why be Bayesian?

- Can model the uncertainty in θ via the full parameter posterior $p(\theta|\mathbf{X})$

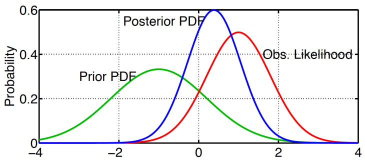


- Can make prediction y_* for test data x_* **by averaging over the posterior** of θ

$$\underbrace{p(y_*|x_*, X, Y)}_{\text{predictive posterior}} = \int p(y_*|x_*, \theta)p(\theta|X, Y)d\theta$$

Why be Bayesian?

- Can model the uncertainty in θ via the full parameter posterior $p(\theta|\mathbf{X})$



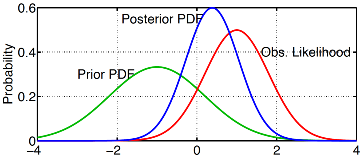
- Can make prediction y_* for test data x_* **by averaging over the posterior** of θ

$$\underbrace{p(y_*|x_*, X, Y)}_{\text{predictive posterior}} = \int p(y_*|x_*, \theta)p(\theta|X, Y)d\theta$$

- Less risk of overfitting since we aren't "fitting" any single parameter to the data

Why be Bayesian?

- Can model the uncertainty in θ via the full parameter posterior $p(\theta|\mathbf{X})$



- Can make prediction y_* for test data x_* **by averaging over the posterior** of θ

$$\underbrace{p(y_*|x_*, X, Y)}_{\text{predictive posterior}} = \int p(y_*|x_*, \theta)p(\theta|X, Y)d\theta$$

- Less risk of overfitting since we aren't "fitting" any single parameter to the data
- Can also use **uncertainty info** in the parameter posterior or predictive posterior to decide which future observations to acquire next (also known as active learning)

Why be Bayesian?

- Sequential data acquisition or "active learning"
- Consider a linear regression task:
  - Can check confidence of the learned model on the label of a test example

$$
\begin{aligned}
p(y|\boldsymbol{x},\theta) &= \text{Normal}(y|\theta^\top \boldsymbol{x},\sigma^2) && \text{Likelihood} \\
p(\theta|\lambda) &= \text{Normal}(\theta|0,\lambda^2) && \text{Prior}
\end{aligned}
$$

Why be Bayesian?

- Sequential data acquisition or "active learning"
- Consider a linear regression task:
  - Can check confidence of the learned model on the label of a test example

$$
\begin{aligned}
p(y|\boldsymbol{x}, \theta) &= \text{Normal}(y|\theta^\top \boldsymbol{x}, \sigma^2) && \text{Likelihood} \\
p(\theta|\lambda) &= \text{Normal}(\theta|0, \lambda^2) && \text{Prior} \\
p(\theta|Y, \mathbf{X}) &= \text{Normal}(\theta|\mu_\theta, \Sigma_\theta) && \text{Parameter posterior}
\end{aligned}
$$

Why be Bayesian?

- Sequential data acquisition or "active learning"
- Consider a linear regression task:
  - Can check confidence of the learned model on the label of a test example

$$
\begin{aligned}
p(y|\boldsymbol{x},\theta) &= \text{Normal}(y|\theta^\top \boldsymbol{x},\sigma^2) && \text{Likelihood} \\
p(\theta|\lambda) &= \text{Normal}(\theta|0,\lambda^2) && \text{Prior} \\
p(\theta|Y,\mathbf{X}) &= \text{Normal}(\theta|\mu_\theta,\Sigma_\theta) && \text{Parameter posterior} \\
p(y_*|\boldsymbol{x}_*,Y,\mathbf{X}) &= \text{Normal}(y_*|\mu_*,\sigma_*^2) && \text{Predictive posterior} \\
\mu_* &= \mu_\theta^\top \boldsymbol{x}_* && \text{Predictive mean} \\
\sigma_*^2 &= \sigma^2 + \boldsymbol{x}_*^\top \Sigma_\theta \boldsymbol{x}_* && \text{Predictive variance}
\end{aligned}
$$

Why be Bayesian?

- Sequential data acquisition or "active learning"
- Consider a linear regression task:
  - Can check confidence of the learned model on the label of a test example

$$
\begin{aligned}
p(y|\boldsymbol{x},\theta) &= \text{Normal}(y|\theta^\top \boldsymbol{x},\sigma^2) && \text{Likelihood} \\
p(\theta|\lambda) &= \text{Normal}(\theta|0,\lambda^2) && \text{Prior} \\
p(\theta|Y,\mathbf{X}) &= \text{Normal}(\theta|\mu_\theta,\Sigma_\theta) && \text{Parameter posterior} \\
p(y_*|\boldsymbol{x}_*,Y,\mathbf{X}) &= \text{Normal}(y_*|\mu_*,\sigma_*^2) && \text{Predictive posterior} \\
\mu_* &= \mu_\theta^\top \boldsymbol{x}_* && \text{Predictive mean} \\
\sigma_*^2 &= \sigma^2 + \boldsymbol{x}_*^\top \Sigma_\theta \boldsymbol{x}_* && \text{Predictive variance}
\end{aligned}
$$

- Can now choose which observations to acquire next for updating the model

Why be Bayesian?

- Consider "black-box" optimization, i.e., optimizing a function whose form is not known and/or derivatives can't be computed (only the function's values can be measured at a small set of points)

Why be Bayesian?

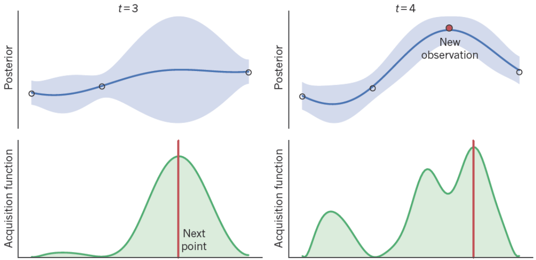
- Consider "black-box" optimization, i.e., optimizing a function whose form is not known and/or derivatives can't be computed (only the function's values can be measured at a small set of points)
- **Bayesian Optimization:** Simultaneously learn the function while finding its optima

Why be Bayesian?

- Consider "black-box" optimization, i.e., optimizing a function whose form is not known and/or derivatives can't be computed (only the function's values can be measured at a small set of points)
- **Bayesian Optimization:** Simultaneously learn the function while finding its optima
  - .. using as few measurements as possible (uncertainty in function's current estimate helps)

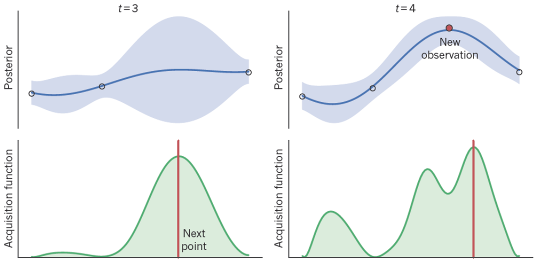
Why be Bayesian?

- Consider "black-box" optimization, i.e., optimizing a function whose form is not known and/or derivatives can't be computed (only the function's values can be measured at a small set of points)
- **Bayesian Optimization:** Simultaneously learn the function while finding its optima
  - .. using as few measurements as possible (uncertainty in function's current estimate helps)



Why be Bayesian?

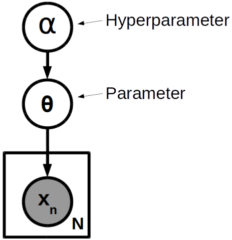
- Consider "black-box" optimization, i.e., optimizing a function whose form is not known and/or derivatives can't be computed (only the function's values can be measured at a small set of points)
- **Bayesian Optimization:** Simultaneously learn the function while finding its optima
  - .. using as few measurements as possible (uncertainty in function's current estimate helps)



- Many applications in "explore and exploit" style problems

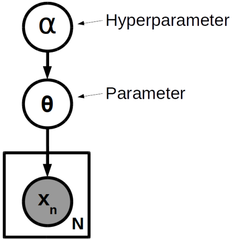
Why be Bayesian?

- **Hierarchical model construction:** parameters can depend on hyperparameters



Why be Bayesian?

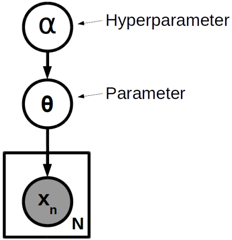
- **Hierarchical model construction:** parameters can depend on hyperparameters



- In a Bayesian setting, hyperparameters need not be tuned but can be **inferred** from data
  - .. by maximizing the marginal likelihood $p(\mathbf{X}|\alpha) = \int p(\mathbf{X}|\theta)p(\theta|\alpha)d\theta$

Why be Bayesian?

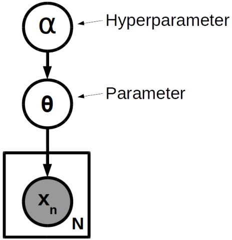
- **Hierarchical model construction:** parameters can depend on hyperparameters



- In a Bayesian setting, hyperparameters need not be tuned but can be **inferred** from data
  - .. by maximizing the marginal likelihood $p(\mathbf{X}|\alpha) = \int p(\mathbf{X}|\theta)p(\theta|\alpha)d\theta$
  - Caveat: Marginal likelihood may need to be approximated (if integral is intractable)

Why be Bayesian?

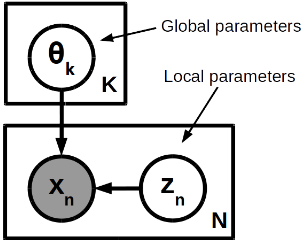
- **Hierarchical model construction:** parameters can depend on hyperparameters



- In a Bayesian setting, hyperparameters need not be tuned but can be **inferred** from data
  - .. by maximizing the marginal likelihood $p(\mathbf{X}|\alpha) = \int p(\mathbf{X}|\theta)p(\theta|\alpha)d\theta$
  - Caveat: Marginal likelihood may need to be approximated (if integral is intractable)
- Examples: Learning the sparsity hyperparameter in sparse regression, learning kernel hyperparameters in kernel methods, and many unsupervised learning problems

Why be Bayesian?

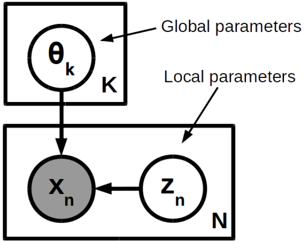
- Can do generative modeling using latent variables that "explain" data



- For each data point $\boldsymbol{x}_n$, we can learn a compact "descriptor" feature representation $\boldsymbol{z}_n$

Why be Bayesian?

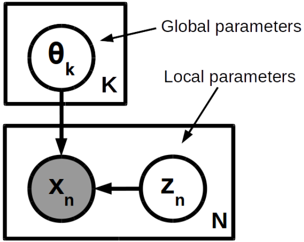
- Can do generative modeling using latent variables that "explain" data



- For each data point $\boldsymbol{x}_n$, we can learn a compact "descriptor" feature representation $\boldsymbol{z}_n$
- Used in many problems, especially unsupervised learning: Gaussian mixture model, probabilistic principal component analysis, factor analysis, topic models, deep feature learning, etc.

Why be Bayesian?

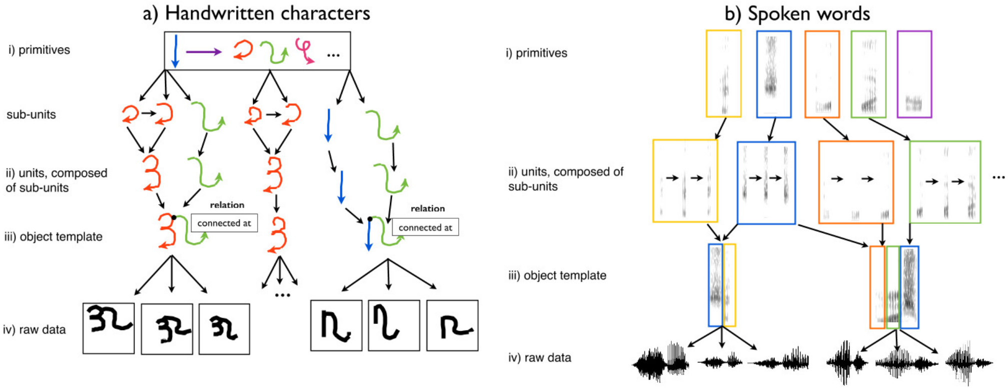
- Can do generative modeling using latent variables that "explain" data



- For each data point $\boldsymbol{x}_n$, we can learn a compact "descriptor" feature representation $\boldsymbol{z}_n$
- Used in many problems, especially unsupervised learning: Gaussian mixture model, probabilistic principal component analysis, factor analysis, topic models, deep feature learning, etc.
- Can also use the latent variables to infer missing data or relevance of each data point

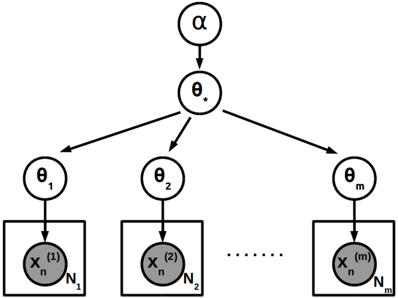
Why be Bayesian?

- Generative modeling also enables synthesizing new data from the learned model (useful in diagnosing whether the learned model is sensible or not, especially in unsupervised learning)



Why be Bayesian?

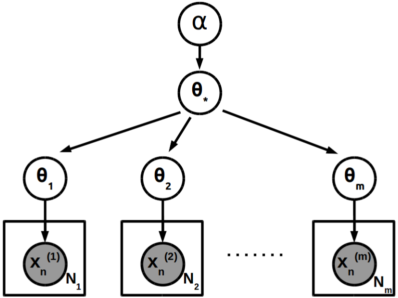
- Simple models can be neatly combined to solve more complex problems



- Allows joint learning across multiple data sets (known as multitask learning or transfer learning)

Why be Bayesian?

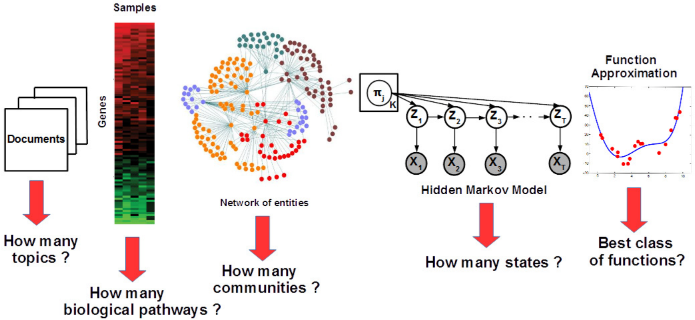
- Simple models can be neatly combined to solve more complex problems



- Allows joint learning across multiple data sets (known as multitask learning or transfer learning)
- Enables different but related models to "share statistical strength"

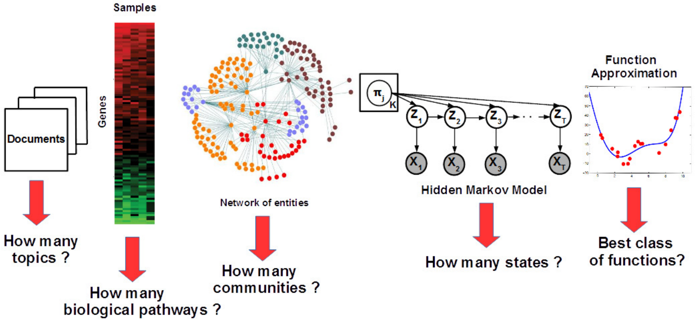
Why be Bayesian?

- **Nonparametric Bayesian Modeling:** A principled way to learn "right" model size/complexity



Why be Bayesian?

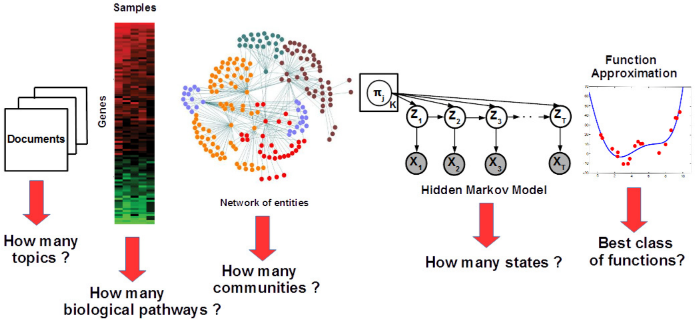
- **Nonparametric Bayesian Modeling:** A principled way to learn "right" model size/complexity



- The model size can grow with data (especially desirable for online learning settings)

Why be Bayesian?

- **Nonparametric Bayesian Modeling:** A principled way to learn "right" model size/complexity



- The model size can grow with data (especially desirable for online learning settings)
- A very elegant modeling paradigm: Can be seen as an infinite limit of finite models

Big Data vs Bayesian

- Recall the basic principle of Bayesian inference

$$P(\theta|\mathbf{X}) = \frac{P(\mathbf{X}|\theta)P(\theta)}{P(\mathbf{X})} \propto \text{Likelihood} \times \text{Prior}$$

- Wouldn't the likelihood overwhelm the prior when data is "big"?
- Wouldn't the posterior concentrate on a point-mass when data is "big"?

Big Data vs Bayesian

- Recall the basic principle of Bayesian inference

$$P(\theta|\mathbf{X}) = \frac{P(\mathbf{X}|\theta)P(\theta)}{P(\mathbf{X})} \propto \text{Likelihood} \times \text{Prior}$$

- Wouldn't the likelihood overwhelm the prior when data is "big"?
- Wouldn't the posterior concentrate on a point-mass when data is "big"?
- For almost all interesting ML problems, the answer is NO. :)

Big Data vs Bayesian

- Recall the basic principle of Bayesian inference

$$P(\theta|\mathbf{X}) = \frac{P(\mathbf{X}|\theta)P(\theta)}{P(\mathbf{X})} \propto \text{Likelihood} \times \text{Prior}$$

- Wouldn't the likelihood overwhelm the prior when data is "big"?
- Wouldn't the posterior concentrate on a point-mass when data is "big"?
- For almost all interesting ML problems, the answer is NO. :)
- Real "big data" problems are **not** characterized by the size of data

Big Data vs Bayesian

- Recall the basic principle of Bayesian inference

$$P(\theta|\mathbf{X}) = \frac{P(\mathbf{X}|\theta)P(\theta)}{P(\mathbf{X})} \propto \text{Likelihood} \times \text{Prior}$$

- Wouldn't the likelihood overwhelm the prior when data is "big"?
- Wouldn't the posterior concentrate on a point-mass when data is "big"?
- For almost all interesting ML problems, the answer is NO. :)
- Real "big data" problems are **not** characterized by the size of data
    - Rather, these problems are characterized by the massive number of parameters to be learned

Big Data vs Bayesian

- Recall the basic principle of Bayesian inference

$$P(\theta|\mathbf{X}) = \frac{P(\mathbf{X}|\theta)P(\theta)}{P(\mathbf{X})} \propto \text{Likelihood} \times \text{Prior}$$

- Wouldn't the likelihood overwhelm the prior when data is "big"?
- Wouldn't the posterior concentrate on a point-mass when data is "big"?
- For almost all interesting ML problems, the answer is NO. :)
- Real "big data" problems are **not** characterized by the size of data
  - Rather, these problems are characterized by the massive number of parameters to be learned
  - However, in these problems, the amount of data available for learning each parameter is very small (e.g., a massive user-ratings data set usually has very few ratings per user)

Big Data vs Bayesian

- Recall the basic principle of Bayesian inference

$$P(\theta|\mathbf{X}) = \frac{P(\mathbf{X}|\theta)P(\theta)}{P(\mathbf{X})} \propto \text{Likelihood} \times \text{Prior}$$

- Wouldn't the likelihood overwhelm the prior when data is "big"?
- Wouldn't the posterior concentrate on a point-mass when data is "big"?
- For almost all interesting ML problems, the answer is NO. :)
- Real "big data" problems are **not** characterized by the size of data
  - Rather, these problems are characterized by the massive number of parameters to be learned
  - However, in these problems, the amount of data available for learning each parameter is very small (e.g., a massive user-ratings data set usually has very few ratings per user)
- Therefore modeling/quantifying parameter uncertainty makes all the more sense

Some Other Interesting Directions..

- Designing flexible and scalable Bayesian models for specific problems, e.g., deep learning
  - Flexibility via nonparametric Bayesian models (to adapt model sizes as warranted by data)
  - Scalability via online Bayesian inference that mirrors online optimization

$$\theta' = \theta_t + \frac{\epsilon_t}{2}\left(\nabla \log \pi_0(\theta_t) + \frac{N}{m}\sum_{n=1}^{m} \nabla \log \pi(x_n \mid \theta_t)\right) + \eta_t$$

Some Other Interesting Directions..

- Designing flexible and scalable Bayesian models for specific problems, e.g., deep learning
  - Flexibility via nonparametric Bayesian models (to adapt model sizes as warranted by data)
  - Scalability via online Bayesian inference that mirrors online optimization

$$\theta' = \theta_t + \frac{\epsilon_t}{2}\left(\nabla \log \pi_0(\theta_t) + \frac{N}{m}\sum_{n=1}^{m} \nabla \log \pi(x_n \mid \theta_t)\right) + \eta_t$$

- Designing (convex/non-convex) optimization problems to approximate a fully Bayesian model

Some Other Interesting Directions..

- Designing flexible and scalable Bayesian models for specific problems, e.g., deep learning
  - Flexibility via nonparametric Bayesian models (to adapt model sizes as warranted by data)
  - Scalability via online Bayesian inference that mirrors online optimization

$$\theta' = \theta_t + \frac{\epsilon_t}{2}\left(\nabla \log \pi_0(\theta_t) + \frac{N}{m}\sum_{n=1}^{m}\nabla \log \pi(x_n \,|\, \theta_t)\right) + \eta_t$$

- Designing (convex/non-convex) optimization problems to approximate a fully Bayesian model
- Exploring connections to prevalent concepts in other "hot" areas in ML, e.g.,
  - Dropout in Deep Learning equivalent to doing approximate Bayesian inference

Some Other Interesting Directions..

- Designing flexible and scalable Bayesian models for specific problems, e.g., deep learning
  - Flexibility via nonparametric Bayesian models (to adapt model sizes as warranted by data)
  - Scalability via online Bayesian inference that mirrors online optimization

$$\theta' = \theta_t + \frac{\epsilon_t}{2}\left(\nabla \log \pi_0(\theta_t) + \frac{N}{m}\sum_{n=1}^{m} \nabla \log \pi(x_n \mid \theta_t)\right) + \eta_t$$

- Designing (convex/non-convex) optimization problems to approximate a fully Bayesian model
- Exploring connections to prevalent concepts in other “hot” areas in ML, e.g.,
  - Dropout in Deep Learning equivalent to doing approximate Bayesian inference
- Developing automated Bayesian modeling and inference methods

Some Other Interesting Directions..

- Designing flexible and scalable Bayesian models for specific problems, e.g., deep learning
  - Flexibility via nonparametric Bayesian models (to adapt model sizes as warranted by data)
  - Scalability via online Bayesian inference that mirrors online optimization

$$\theta' = \theta_t + \frac{\epsilon_t}{2}\left(\nabla \log \pi_0(\theta_t) + \frac{N}{m}\sum_{n=1}^{m} \nabla \log \pi(x_n \mid \theta_t)\right) + \eta_t$$

- Designing (convex/non-convex) optimization problems to approximate a fully Bayesian model
- Exploring connections to prevalent concepts in other "hot" areas in ML, e.g.,
  - Dropout in Deep Learning equivalent to doing approximate Bayesian inference
- Developing automated Bayesian modeling and inference methods
  - Probabilistic Programming: Express probabilistic models via computer programs and perform automatic inference (check out **Stan**)

Some Resources

On Probabilistic/Bayesian Modeling

- Book: Pattern Recognition and Machine Learning (Chris Bishop)
- Book: Machine Learning - A Probabilistic Perspective (Kevin Murphy)
- Introductory Paper: "Bayesian Inference: An Introduction to Principles and Practice in Machine Learning" (Mike Tipping) `http://www.miketipping.com/papers/met-mlbayes.pdf`
- Introductory Paper: "Probabilistic machine learning and artificial intelligence" (Zoubin Ghahramani) `http://www.nature.com/nature/journal/v521/n7553/full/nature14541.html`
- "Roadmap" to learning about Bayesian learning (from Metaacademy): `https://www.metacademy.org/roadmaps/rgrosse/bayesian_machine_learning`

On Bayesian Optimization

- Taking the Human Out of the Loop: A Review of Bayesian Optimization (Shahriari et al, 2015) `https://www.cs.ox.ac.uk/people/nando.defreitas/publications/BayesOptLoop.pdf`

On Probabilistic Programming

- Check out Stan `http://mc-stan.org/`

On Nonparametric Bayesian Learning

- A brief tutorial: A tutorial on Bayesian nonparametric models (Gershman & Blei, 2012) `http://web.mit.edu/sjgershm/www/GershmanBlei12.pdf`

On Gaussian Processes

- Book: Gaussian Processes for Machine Learning (freely available online)
- MATLAB Package - GPML: `http://www.gaussianprocess.org/gpml/code/matlab/doc/`
- MATLAB Package - GPStuff: `http://research.cs.aalto.fi/pml/software/gpstuff/`

Thanks! Questions?