Course Logistics and Introduction

Piyush Rai

Probabilistic Machine Learning (CS772A)

Aug 1, 2017

Course Logistics

- **Course name:** Probabilistic Machine Learning (CS772A) - “PML”
- **Timing and Venue:** Tue/Th 6:00-7:30pm, RM-101
- **Instructor:** Piyush Rai (Email: piyush@cse.iitk.ac.in; please prefix email subject with CS772A)
- **Instructor Office Hours:** Wednesday 11:00-12:00pm (KD-319), or by appointment.

Course Logistics

- **Course name:** Probabilistic Machine Learning (CS772A) - "PML"
- **Timing and Venue:** Tue/Th 6:00-7:30pm, RM-101
- **Instructor:** Piyush Rai (Email: piyush@cse.iitk.ac.in; please prefix email subject with CS772A)
- **Instructor Office Hours:** Wednesday 11:00-12:00pm (KD-319), or by appointment.
- **TAs:** Gundeep Arora, Smrithi Prabhu, Vinay Verma

Course Logistics

- **Course name:** Probabilistic Machine Learning (CS772A) - "PML"
- **Timing and Venue:** Tue/Th 6:00-7:30pm, RM-101
- **Instructor:** Piyush Rai (Email: piyush@cse.iitk.ac.in; please prefix email subject with CS772A)
- **Instructor Office Hours:** Wednesday 11:00-12:00pm (KD-319), or by appointment.
- **TAs:** Gundeep Arora, Smrithi Prabhu, Vinay Verma
- **Course website:** `https://goo.gl/WufUSL` (where course material and readings will be posted)

Course Logistics

- **Course name:** Probabilistic Machine Learning (CS772A) - "PML"
- **Timing and Venue:** Tue/Th 6:00-7:30pm, RM-101
- **Instructor:** Piyush Rai (Email: piyush@cse.iitk.ac.in; please prefix email subject with CS772A)
- **Instructor Office Hours:** Wednesday 11:00-12:00pm (KD-319), or by appointment.
- **TAs:** Gundeep Arora, Smrithi Prabhu, Vinay Verma
- **Course website:** `https://goo.gl/WufUSL` (where course material and readings will be posted)
- **Discussion site:** Piazza (`https://goo.gl/wyM541`). Please register and participate actively.

Course Logistics

- **Course name:** Probabilistic Machine Learning (CS772A) - "PML"
- **Timing and Venue:** Tue/Th 6:00-7:30pm, RM-101
- **Instructor:** Piyush Rai (Email: piyush@cse.iitk.ac.in; please prefix email subject with CS772A)
- **Instructor Office Hours:** Wednesday 11:00-12:00pm (KD-319), or by appointment.
- **TAs:** Gundeep Arora, Smrithi Prabhu, Vinay Verma
- **Course website:** `https://goo.gl/WufUSL` (where course material and readings will be posted)
- **Discussion site:** Piazza (`https://goo.gl/wyM541`). Please register and participate actively.
- **Attendance Policy:** None, but missing too many classes will make it hard for you to catch up with the material.

Course Logistics

- **Course name:** Probabilistic Machine Learning (CS772A) - "PML"
- **Timing and Venue:** Tue/Th 6:00-7:30pm, RM-101
- **Instructor:** Piyush Rai (Email: piyush@cse.iitk.ac.in; please prefix email subject with CS772A)
- **Instructor Office Hours:** Wednesday 11:00-12:00pm (KD-319), or by appointment.
- **TAs:** Gundeep Arora, Smrithi Prabhu, Vinay Verma
- **Course website:** `https://goo.gl/WufUSL` (where course material and readings will be posted)
- **Discussion site:** Piazza (`https://goo.gl/wyM541`). Please register and participate actively.
- **Attendance Policy:** None, but missing too many classes will make it hard for you to catch up with the material.
- **Auditing?** Please let me know your email id to be added to the mailing list. You may also submit homeworks and appear for exams (we'll try to grade but it is not guaranteed)

Grading Policy

- 5 homework assignments: 30%
  - Pen and paper based + some programming in MATLAB/Python

Grading Policy

- 5 homework assignments: 30%
  - Pen and paper based + some programming in MATLAB/Python
- Midterm exam: 20%

Grading Policy

- 5 homework assignments: 30%
  - Pen and paper based + some programming in MATLAB/Python
- Midterm exam: 20%
- Final exam: 25%

Grading Policy

- 5 homework assignments: 30%
  - Pen and paper based + some programming in MATLAB/Python
- Midterm exam: 20%
- Final exam: 25%
- Class Project: 25%

Grading Policy

- 5 homework assignments: 30%
  - Pen and paper based + some programming in MATLAB/Python
- Midterm exam: 20%
- Final exam: 25%
- Class Project: 25%
  - To be done in groups of 2-3

Grading Policy

- 5 homework assignments: 30%
  - Pen and paper based + some programming in MATLAB/Python
- Midterm exam: 20%
- Final exam: 25%
- Class Project: 25%
  - To be done in groups of 2-3
  - Producing publishable work in the class project may earn you a direct A grade

Grading Policy

- 5 homework assignments: 30%
  - Pen and paper based + some programming in MATLAB/Python
- Midterm exam: 20%
- Final exam: 25%
- Class Project: 25%
  - To be done in groups of 2-3
  - Producing publishable work in the class project may earn you a direct A grade
- Class/Piazza participation: 5% (bonus, based on instructor's discretion)

Grading Policy

- 5 homework assignments: 30%
  - Pen and paper based + some programming in MATLAB/Python
- Midterm exam: 20%
- Final exam: 25%
- Class Project: 25%
  - To be done in groups of 2-3
  - Producing publishable work in the class project may earn you a direct A grade
- Class/Piazza participation: 5% (bonus, based on instructor's discretion)
- Note: Exams will be closed-book (an A4 size cheat-sheet allowed)

Course Project

- Worth 25% of your total grade for this course

Course Project

- Worth 25% of your total grade for this course
- A list of possible projects will soon be shared (by next week)
  - Many options: Research project, implemention of research papers, survey of a topic, etc.

Course Project

- Worth 25% of your total grade for this course
- A list of possible projects will soon be shared (by next week)
  - Many options: Research project, implemention of research papers, survey of a topic, etc.
- You can propose your own idea (but has to be related to probabilistic ML)

Course Project

- Worth 25% of your total grade for this course
- A list of possible projects will soon be shared (by next week)
  - Many options: Research project, implemention of research papers, survey of a topic, etc.
- You can propose your own idea (but has to be related to probabilistic ML)
- Some deadlines

Course Project

- Worth 25% of your total grade for this course
- A list of possible projects will soon be shared (by next week)
  - Many options: Research project, implemention of research papers, survey of a topic, etc.
- You can propose your own idea (but has to be related to probabilistic ML)
- Some deadlines
  - Project proposal (ideally 2 pages): Worth 5%. Due on Sept 10

Course Project

- Worth 25% of your total grade for this course
- A list of possible projects will soon be shared (by next week)
  - Many options: Research project, implemention of research papers, survey of a topic, etc.
- You can propose your own idea (but has to be related to probabilistic ML)
- Some deadlines
  - Project proposal (ideally 2 pages): Worth 5%. Due on Sept 10
  - Mid-term progress report: Worth 5%. Due on Oct 12

Course Project

- Worth 25% of your total grade for this course
- A list of possible projects will soon be shared (by next week)
  - Many options: Research project, implemention of research papers, survey of a topic, etc.
- You can propose your own idea (but has to be related to probabilistic ML)
- Some deadlines
  - Project proposal (ideally 2 pages): Worth 5%. Due on Sept 10
  - Mid-term progress report: Worth 5%. Due on Oct 12
  - In-class final presentation: Worth 5%. During last week of classes

Course Project

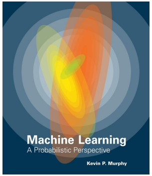
- Worth 25% of your total grade for this course
- A list of possible projects will soon be shared (by next week)
  - Many options: Research project, implemention of research papers, survey of a topic, etc.
- You can propose your own idea (but has to be related to probabilistic ML)
- Some deadlines
  - Project proposal (ideally 2 pages): Worth 5%. Due on Sept 10
  - Mid-term progress report: Worth 5%. Due on Oct 12
  - In-class final presentation: Worth 5%. During last week of classes
  - Final report: Worth 10%. Due end of semester.

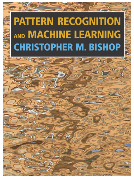
Textbook and Readings

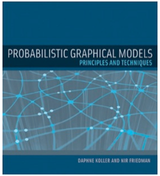
- **Textbook:** No official textbook required
  - Required reading material will be provided on the class webpage

Textbook and Readings

- **Textbook:** No official textbook required
  - Required reading material will be provided on the class webpage
- Some books that you may use as reference
  - Kevin Murphy, Machine Learning: A Probabilistic Perspective (MLAPP), The MIT Press, 2012.
  - Christopher Bishop, Pattern Recognition and Machine Learning (PRML), Springer, 2007.
  - David Barber. Bayesian Reasoning and Machine Learning, Cambridge Univ. Press, 2012.
  - Daphne Koller. Probabilistic Graphical Models, The MIT Press, 2009.

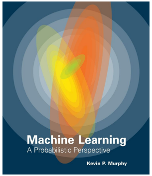


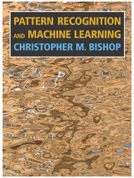


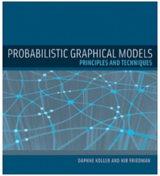


Textbook and Readings

- **Textbook:** No official textbook required
  - Required reading material will be provided on the class webpage
- Some books that you may use as reference
  - Kevin Murphy, Machine Learning: A Probabilistic Perspective (MLAPP), The MIT Press, 2012.
  - Christopher Bishop, Pattern Recognition and Machine Learning (PRML), Springer, 2007.
  - David Barber. Bayesian Reasoning and Machine Learning, Cambridge Univ. Press, 2012.
  - Daphne Koller. Probabilistic Graphical Models, The MIT Press, 2009.







- Several other monographs on some specific topics will be provided for readings

Background Expected (Important)

- You should be comfortable with concepts from probability and probability distributions
- Please brush-up the basics if you are rusty on these concepts (refer to probability refresher slides and relevant book chapters from PRML/MLAPP)

Background Expected (Important)

- You should be comfortable with concepts from probability and probability distributions
- Please brush-up the basics if you are rusty on these concepts (refer to probability refresher slides and relevant book chapters from PRML/MLAPP)
- You should also be comfortable with basics of Linear Algebra and Optimization

Background Expected (Important)

- You should be comfortable with concepts from probability and probability distributions
- Please brush-up the basics if you are rusty on these concepts (refer to probability refresher slides and relevant book chapters from PRML/MLAPP)
- You should also be comfortable with basics of Linear Algebra and Optimization
- The course assumes familiarity to material from Intro to ML (CS771)

Background Expected (Important)

- You should be comfortable with concepts from probability and probability distributions
- Please brush-up the basics if you are rusty on these concepts (refer to probability refresher slides and relevant book chapters from PRML/MLAPP)
- You should also be comfortable with basics of Linear Algebra and Optimization
- The course assumes familiarity to material from Intro to ML (CS771)
    - This will be important to fully appreciate how the probabilistic view of ML is connected to the non-probabilistic view, and what are its strengths/weaknesses

Background Expected (Important)

- You should be comfortable with concepts from probability and probability distributions
- Please brush-up the basics if you are rusty on these concepts (refer to probability refresher slides and relevant book chapters from PRML/MLAPP)
- You should also be comfortable with basics of Linear Algebra and Optimization
- The course assumes familiarity to material from Intro to ML (CS771)
  - This will be important to fully appreciate how the probabilistic view of ML is connected to the non-probabilistic view, and what are its strengths/weaknesses
  - Note that CS772 will NOT talk about non-probabilistic approaches such as Support Vector Machines or various distance based methods, but a knowledge of such methods will help you in the course

Background Expected (Important)

- You should be comfortable with concepts from probability and probability distributions
- Please brush-up the basics if you are rusty on these concepts (refer to probability refresher slides and relevant book chapters from PRML/MLAPP)
- You should also be comfortable with basics of Linear Algebra and Optimization
- The course assumes familiarity to material from Intro to ML (CS771)
  - This will be important to fully appreciate how the probabilistic view of ML is connected to the non-probabilistic view, and what are its strengths/weaknesses
  - Note that CS772 will NOT talk about non-probabilistic approaches such as Support Vector Machines or various distance based methods, but a knowledge of such methods will help you in the course
  - For reference, you may refer to material from last year's offering of CS771: `https://goo.gl/ViR1QW`

Collaboration vs Cheating

- Collaboration is encouraged. Cheating will lead to strict punishments.

Collaboration vs Cheating

- Collaboration is encouraged. Cheating will lead to strict punishments.
- Feel free to discuss homework assignments with your classmates.

Collaboration vs Cheating

- Collaboration is encouraged. Cheating will lead to strict punishments.
- Feel free to discuss homework assignments with your classmates.
- However, you are supposed to write your own solution in your own words (same goes for coding assignments)

Collaboration vs Cheating

- Collaboration is encouraged. Cheating will lead to strict punishments.
- Feel free to discuss homework assignments with your classmates.
- However, you are supposed to write your own solution in your own words (same goes for coding assignments)
- Plagiarism from other sources (for assignments/project) will also lead to strict punishment unless you duly credit the original source(s)

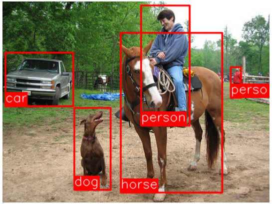
Collaboration vs Cheating

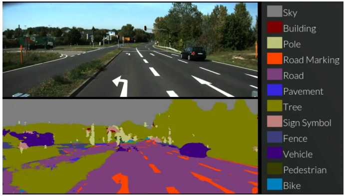
- Collaboration is encouraged. Cheating will lead to strict punishments.
- Feel free to discuss homework assignments with your classmates.
- However, you are supposed to write your own solution in your own words (same goes for coding assignments)
- Plagiarism from other sources (for assignments/project) will also lead to strict punishment unless you duly credit the original source(s)
- Other things that will lead to punishment
  - Use of unfair means in the exams
  - Fabricating experimental results in assignments/project

Machine Learning 101

What is Machine Learning?

- Machines trying to make sense of data by "looking" at the data (and making decisions)

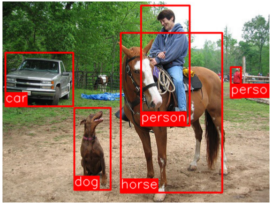


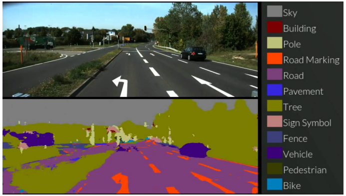


- The rules behind these decisions are not hard-coded but "inferred" from the (training) data

What is Machine Learning?

- Machines trying to make sense of data by "looking" at the data (and making decisions)



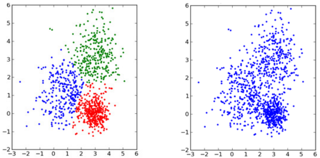


- The rules behind these decisions are not hard-coded but "inferred" from the (training) data
- Important: The rules must "generalize" to future (unseen) data

(Pic credits: kdnuggets.com, Girshick et al, CVPR 2014)

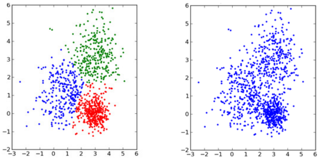
How Machines Learn?

- Primarily via one of the two ways
  - Learning with Supervision (*a.k.a.* "Supervised" Learning)
  - Learning without Supervision (*a.k.a.* "Unsupervised" Learning)



How Machines Learn?

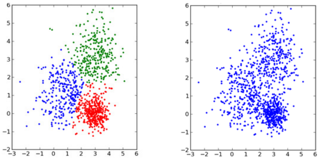
- Primarily via one of the two ways
  - Learning with Supervision (*a.k.a.* "Supervised" Learning)
  - Learning without Supervision (*a.k.a.* "Unsupervised" Learning)



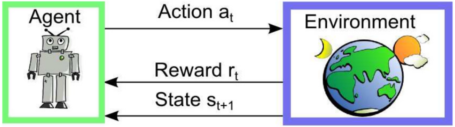
- Note: Often, supervision may only be available for part of the data ("Semi-supervised" Learning)

How Machines Learn?

- Primarily via one of the two ways
  - Learning with Supervision (*a.k.a.* "Supervised" Learning)
  - Learning without Supervision (*a.k.a.* "Unsupervised" Learning)



- Note: Often, supervision may only be available for part of the data ("Semi-supervised" Learning)
- Note: Many other learning paradigms exist (e.g, Learning through interaction)



Supervised Learning

- Given: Training data $\mathcal{D}$ in form of N input-output pairs $\{(\boldsymbol{x}_1, \boldsymbol{y}_1), \ldots, (\boldsymbol{x}_N, \boldsymbol{y}_N)\}$
- Goal: Learn a function $f : \boldsymbol{x} \mapsto \boldsymbol{y}$ that can predict the output $\boldsymbol{y}_*$ for a new test input $\boldsymbol{x}_*$

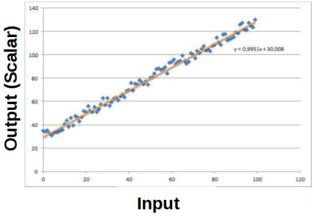
Supervised Learning

- Given: Training data $\mathcal{D}$ in form of N input-output pairs $\{(\boldsymbol{x}_1, \boldsymbol{y}_1), \ldots, (\boldsymbol{x}_N, \boldsymbol{y}_N)\}$
- Goal: Learn a function $f : \boldsymbol{x} \mapsto \boldsymbol{y}$ that can predict the output $\boldsymbol{y}_*$ for a new test input $\boldsymbol{x}_*$
- Many variants of this problem based on the nature of the output $\boldsymbol{y}$

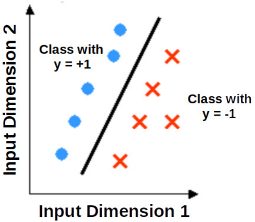
Supervised Learning

- Given: Training data $\mathcal{D}$ in form of N input-output pairs $\{(\boldsymbol{x}_1, \boldsymbol{y}_1), \ldots, (\boldsymbol{x}_N, \boldsymbol{y}_N)\}$
- Goal: Learn a function $f : \boldsymbol{x} \mapsto \boldsymbol{y}$ that can predict the output $\boldsymbol{y}_*$ for a new test input $\boldsymbol{x}_*$
- Many variants of this problem based on the nature of the output $\boldsymbol{y}$
- Some examples: Regression (scalar-valued $\boldsymbol{y}$), Classification (discrete-valued y)

(Linear) Regression



(Binary) Classification

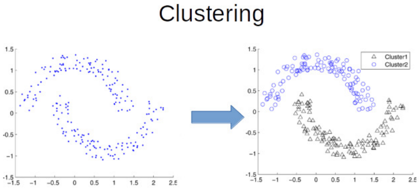


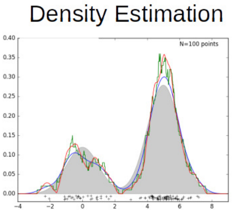
Unsupervised Learning

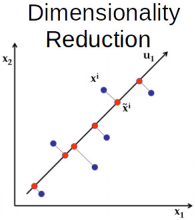
- Given: Training data $\mathcal{D}$ in form of N unlabeled inputs $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$
- Goal: Learn the underlying "latent" structure in the inputs

Unsupervised Learning

- Given: Training data $\mathcal{D}$ in form of N unlabeled inputs $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$
- Goal: Learn the underlying "latent" structure in the inputs
- Some examples: Clustering, Density Estimation, Dimensionality Reduction,



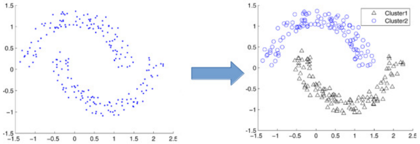




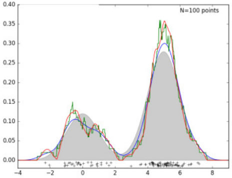
Unsupervised Learning

- Given: Training data $\mathcal{D}$ in form of N unlabeled inputs $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$
- Goal: Learn the underlying "latent" structure in the inputs
- Some examples: Clustering, Density Estimation, Dimensionality Reduction,

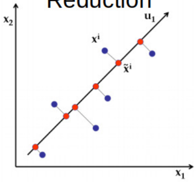
Clustering



Density Estimation



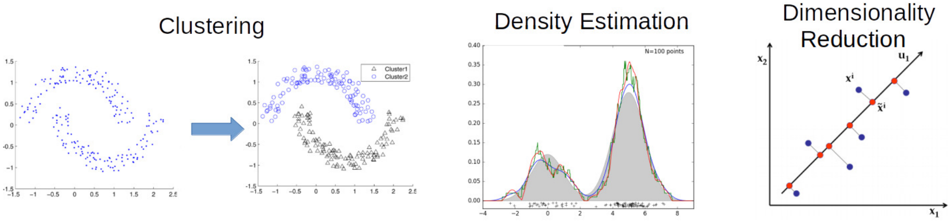
Dimensionality Reduction



- Note: Unsupervised Learning is also related to "Data Compression"

Unsupervised Learning

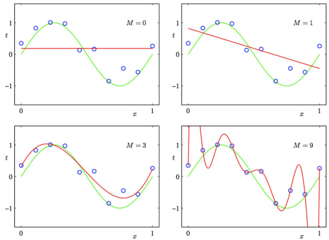
- Given: Training data $\mathcal{D}$ in form of N unlabeled inputs $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$
- Goal: Learn the underlying "latent" structure in the inputs
- Some examples: Clustering, Density Estimation, Dimensionality Reduction,

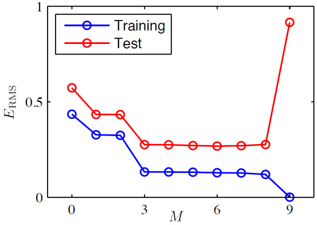


- Note: Unsupervised Learning is also related to "Data Compression"
- **Note:** Almost any ML problem (even supervised!) can be framed as a density estimation problem! As we will see, the probabilistic view of ML makes this connection even more explicit.

Overfitting vs Generalization

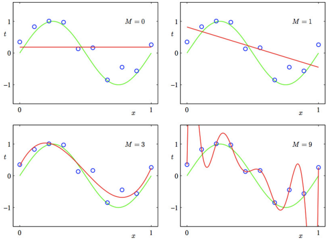
- Simple hypotheses/rules are preferred over more complex ones
- Reason: We care about future performance on test data, rather than the training data

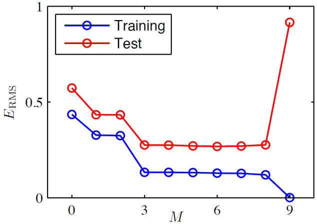




Overfitting vs Generalization

- Simple hypotheses/rules are preferred over more complex ones
- Reason: We care about future performance on test data, rather than the training data

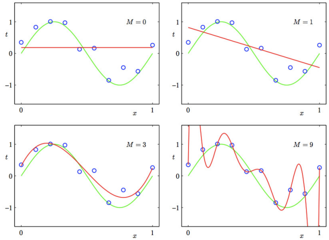


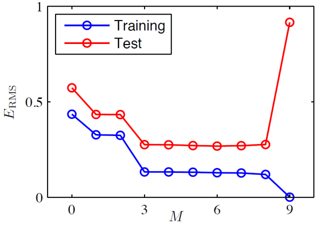


- Desired: hypotheses that are not too simple, not too complex

Overfitting vs Generalization

- Simple hypotheses/rules are preferred over more complex ones
- Reason: We care about future performance on test data, rather than the training data





- Desired: hypotheses that are not too simple, not too complex
- ML algorithms use *regularization* to achieve this

Probabilistic Machine Learning

Why a Probabilistic Approach?

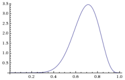
- Data could be imprecise/noisy. Need to model the noise/uncertainty explicitly.

Why a Probabilistic Approach?

- Data could be imprecise/noisy. Need to model the noise/uncertainty explicitly.
- Often we need probabilistic predictions (e.g., probability that a transaction is fraud)

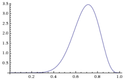
Why a Probabilistic Approach?

- Data could be imprecise/noisy. Need to model the noise/uncertainty explicitly.
- Often we need probabilistic predictions (e.g., probability that a transaction is fraud)
- There could also be uncertainty in the estimated model parameters

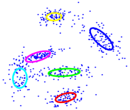


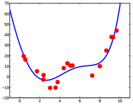
Why a Probabilistic Approach?

- Data could be imprecise/noisy. Need to model the noise/uncertainty explicitly.
- Often we need probabilistic predictions (e.g., probability that a transaction is fraud)
- There could also be uncertainty in the estimated model parameters



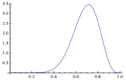
- There could be uncertainty in the learned model size/structure.



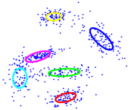


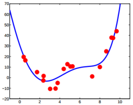
Why a Probabilistic Approach?

- Data could be imprecise/noisy. Need to model the noise/uncertainty explicitly.
- Often we need probabilistic predictions (e.g., probability that a transaction is fraud)
- There could also be uncertainty in the estimated model parameters

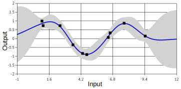


- There could be uncertainty in the learned model size/structure.





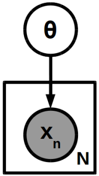
- Predictions made by the model may have uncertainty



Modeling Data Probabilistically

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

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

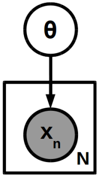


- The above picture denotes a simplistic "plate notation" graphical model

Modeling Data Probabilistically

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

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

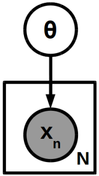


- The above picture denotes a simplistic "plate notation" graphical model
- Note: Shaded nodes = observed; unshaded nodes = unknown/unobserved

Modeling Data Probabilistically

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

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

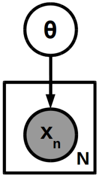


- The above picture denotes a simplistic "plate notation" graphical model
- Note: Shaded nodes = observed; unshaded nodes = unknown/unobserved
- Goal: To estimate the unknowns of the model (θ in this case), given the observed data $\mathbf{X}$

Modeling Data Probabilistically

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

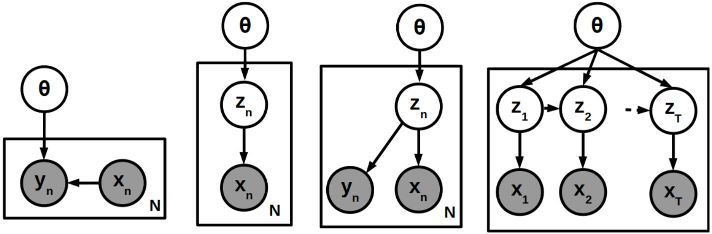
$$\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N \sim p(\boldsymbol{x}|\theta)$$



- The above picture denotes a simplistic "plate notation" graphical model
- Note: Shaded nodes = observed; unshaded nodes = unknown/unobserved
- Goal: To estimate the unknowns of the model (θ in this case), given the observed data $\mathbf{X}$
- Can use the learned model to make predictions
  - E.g., the probability $p(\boldsymbol{x}_*|\theta)$ of a new input $\boldsymbol{x}_*$ under this model

Modeling Data Probabilistically

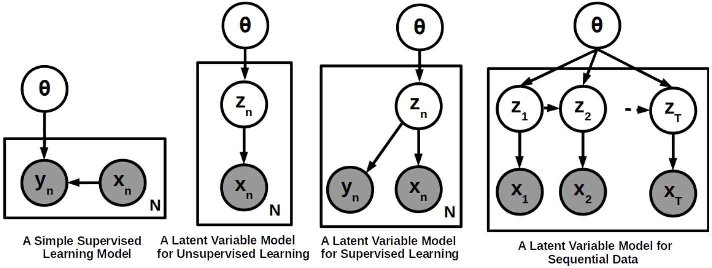
- This basic problem set-up can be generalized in various ways



A Simple Supervised Learning Model | A Latent Variable Model for Unsupervised Learning | A Latent Variable Model for Supervised Learning | A Latent Variable Model for Sequential Data

Modeling Data Probabilistically

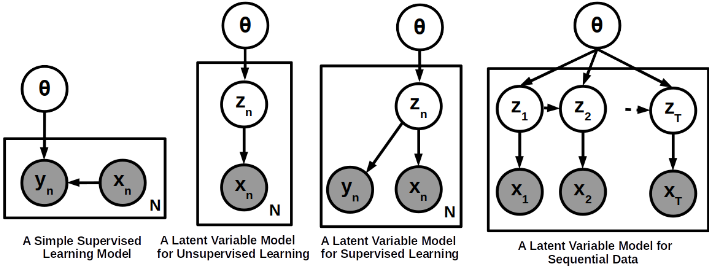
- This basic problem set-up can be generalized in various ways



- Any node (even if observed) that we are uncertain about is modeled by a probability distribution
  - These nodes become the random variables of the model

Modeling Data Probabilistically

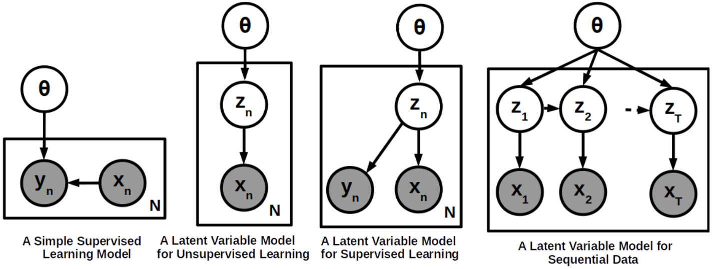
- This basic problem set-up can be generalized in various ways



- Any node (even if observed) that we are uncertain about is modeled by a probability distribution
  - These nodes become the random variables of the model
- The full model is specified via a joint prob. distribution over all random variables

Modeling Data Probabilistically

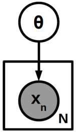
- This basic problem set-up can be generalized in various ways



- Any node (even if observed) that we are uncertain about is modeled by a probability distribution
  - These nodes become the random variables of the model
- The full model is specified via a joint prob. distribution over all random variables
- The goal is to infer the unknowns of the model, given the observed data

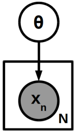
Modeling Data Probabilistically

- Specification of probabilistic models requires two key ingredients: Likelihood and prior



Modeling Data Probabilistically

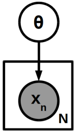
- Specification of probabilistic models requires two key ingredients: Likelihood and prior



- Likelihood function $p(\boldsymbol{x}|\theta)$ or the "observation model" specifies how data is generated
  - Measures data fit (or "loss") w.r.t. the given parameter θ

Modeling Data Probabilistically

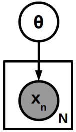
- Specification of probabilistic models requires two key ingredients: Likelihood and prior



- Likelihood function $p(\boldsymbol{x}|\theta)$ or the "observation model" specifies how data is generated
  - Measures data fit (or "loss") w.r.t. the given parameter θ
- Prior distribution $p(\theta)$ specifies how likely different parameter values are *a priori*
  - Also corresponds to imposing a "regularizer" over θ

Modeling Data Probabilistically

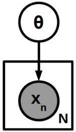
- Specification of probabilistic models requires two key ingredients: Likelihood and prior



- Likelihood function $p(\boldsymbol{x}|\theta)$ or the "observation model" specifies how data is generated
    - Measures data fit (or "loss") w.r.t. the given parameter θ
- Prior distribution $p(\theta)$ specifies how likely different parameter values are *a priori*
    - Also corresponds to imposing a "regularizer" over θ
- Domain knowledge can help in the specification of the likelihood and the prior

Parameter Estimation/Inference in Probabilistic Models

- Perhaps the simplest way is to find θ that makes the observed data most likely or most probable

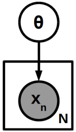


- Formally, find θ that maximizes the probability of the observed data

$$\hat{\theta} = \arg\max_{\theta} \log p(\mathbf{X}|\theta)$$

Parameter Estimation/Inference in Probabilistic Models

- Perhaps the simplest way is to find θ that makes the observed data most likely or most probable



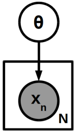
- Formally, find θ that maximizes the probability of the observed data

$$\hat{\theta} = \arg\max_{\theta} \log p(\mathbf{X}|\theta)$$

- However, this gives a single "point" estimate of θ. Doesn't tell us about the uncertainty in θ

Parameter Estimation/Inference in Probabilistic Models

- Perhaps the simplest way is to find θ that makes the observed data most likely or most probable



- Formally, find θ that maximizes the probability of the observed data

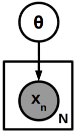
$$\hat{\theta} \quad = \quad \arg\max_{\theta} \; \log p(\mathbf{X}|\theta)$$

- However, this gives a single "point" estimate of θ. Doesn't tell us about the uncertainty in θ
- We can estimate the full posterior distribution over θ to get the uncertainty

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

Parameter Estimation/Inference in Probabilistic Models

- Perhaps the simplest way is to find θ that makes the observed data most likely or most probable



- Formally, find θ that maximizes the probability of the observed data

$$\hat{\theta} \quad = \quad \arg\max_{\theta} \; \log p(\mathbf{X}|\theta)$$

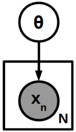
- However, this gives a single "point" estimate of θ. Doesn't tell us about the uncertainty in θ
- We can estimate the full posterior distribution over θ to get the uncertainty

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

- This is called **Bayesian inference**. The posterior distribution captures the uncertainty in θ

Parameter Estimation/Inference in Probabilistic Models

- Perhaps the simplest way is to find θ that makes the observed data most likely or most probable



- Formally, find θ that maximizes the probability of the observed data

$$\hat{\theta} \quad = \quad \arg\max_{\theta} \; \log p(\mathbf{X}|\theta)$$

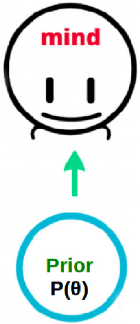
- However, this gives a single "point" estimate of θ. Doesn't tell us about the uncertainty in θ
- We can estimate the full posterior distribution over θ to get the uncertainty

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

- This is called **Bayesian inference**. The posterior distribution captures the uncertainty in θ
- We will study both point estimation and Bayesian inference methods (and hybrids!)

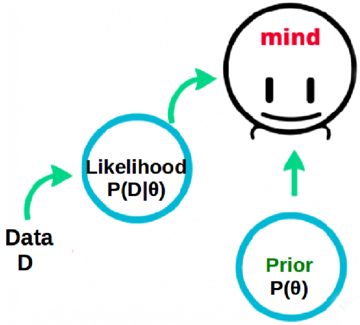
Bayesian Inference

- Bayesian inference fits naturally into an "online" learning setting



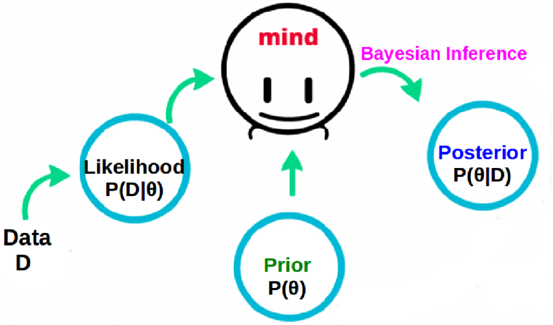
Bayesian Inference

- Bayesian inference fits naturally into an "online" learning setting



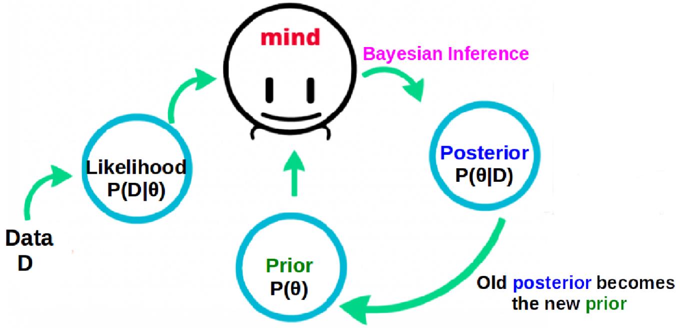
Bayesian Inference

- Bayesian inference fits naturally into an "online" learning setting



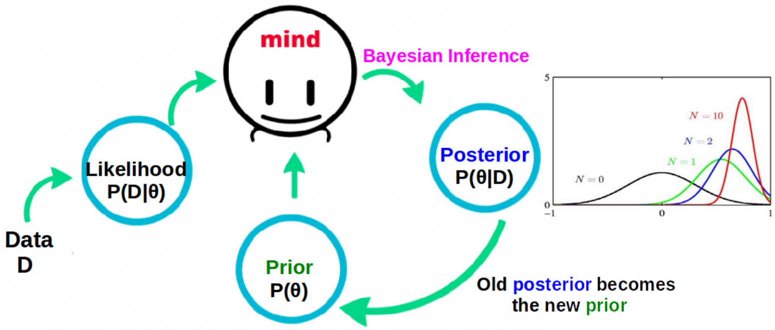
Bayesian Inference

- Bayesian inference fits naturally into an "online" learning setting



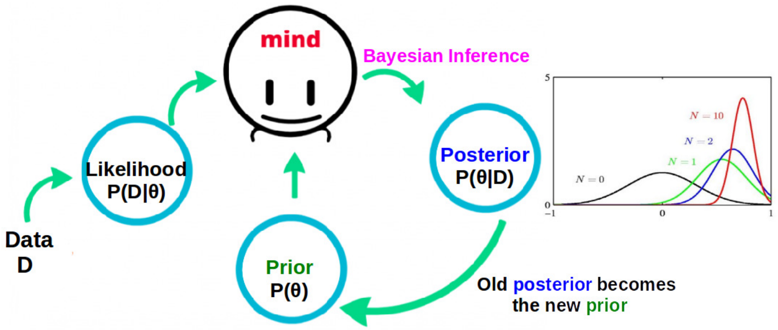
Bayesian Inference

- Bayesian inference fits naturally into an "online" learning setting



Bayesian Inference

- Bayesian inference fits naturally into an "online" learning setting

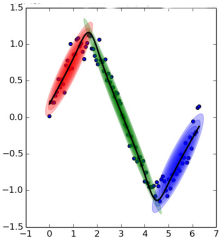


- Our belief about θ keeps getting updated as we see more and more data

Some Other Benefits of the Probabilistic Approach

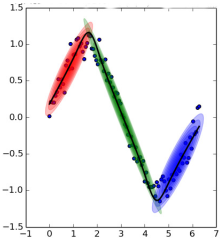
Modular Construction of Complex Models

- Can easily construct combinations of multiple simple probabilistic models to learn complex patterns
- An example: Can perform nonlinear classification using a mixture of linear classifiers
  - It is a simple yet powerful combination of two models - one that performs clustering of the data and the other that learns a linear classifier within each cluster (both learned jointly)



Modular Construction of Complex Models

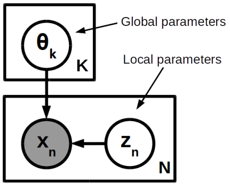
- Can easily construct combinations of multiple simple probabilistic models to learn complex patterns
- An example: Can perform nonlinear classification using a mixture of linear classifiers
  - It is a simple yet powerful combination of two models - one that performs clustering of the data and the other that learns a linear classifier within each cluster (both learned jointly)



- More generally, these are called "mixture of experts" models

Generative Models

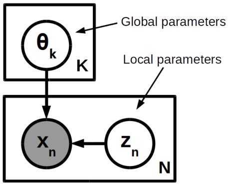
- Generative models of data can be naturally specified in a probabilistic framework



- Each data point $\boldsymbol{x}_n$ is associated with latent variables $\boldsymbol{z}_n$
- Latent variables can be used a compact representation or an "encoding" of the data
- Such models are used in many problems, especially unsupervised learning: Gaussian mixture model, probabilistic principal component analysis, topic models, deep generative models, etc.

Generative Models

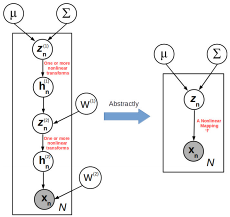
- Generative models of data can be naturally specified in a probabilistic framework



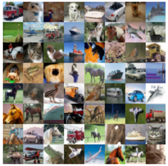
- Each data point $\boldsymbol{x}_n$ is associated with latent variables $\boldsymbol{z}_n$
- Latent variables can be used a compact representation or an "encoding" of the data
- Such models are used in many problems, especially unsupervised learning: Gaussian mixture model, probabilistic principal component analysis, topic models, deep generative models, etc.
- Can also use the latent variables to infer missing data or relevance of each data point

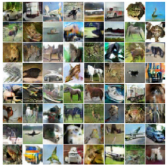
(Deep) Generative Models

- Deep Generative Models for extremely popular nowadays (e.g., Variational Auto-encoders and Generative Adversarial Networks)



- Once learned, these models can also synthesize realistic looking "new" data from random $\boldsymbol{z}$'s


Real images (CIFAR-10)


Generated images

Averaging Over Posterior Distribution

- Can use the posterior distribution over parameters to compute "averaged prediction", e.g.,

$$p(\boldsymbol{y}_* = 1|\boldsymbol{x}_*, \mathbf{X}, \boldsymbol{y}) = \int p(\boldsymbol{y}_* = 1|\boldsymbol{x}_*, \theta)p(\theta|\mathbf{X}, \boldsymbol{y})d\theta$$

- $p(\boldsymbol{y}_* = 1|\boldsymbol{x}_*, \mathbf{X}, \boldsymbol{y})$ with θ "integrated out" is called posterior predictive distribution

Averaging Over Posterior Distribution

- Can use the posterior distribution over parameters to compute "averaged prediction", e.g.,

$$p(\boldsymbol{y}_* = 1|\boldsymbol{x}_*, \mathbf{X}, \boldsymbol{y}) = \int p(\boldsymbol{y}_* = 1|\boldsymbol{x}_*, \theta)p(\theta|\mathbf{X}, \boldsymbol{y})d\theta$$

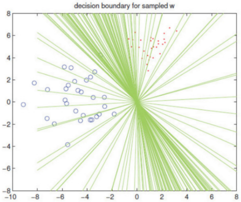
- $p(\boldsymbol{y}_* = 1|\boldsymbol{x}_*, \mathbf{X}, \boldsymbol{y})$ with θ "integrated out" is called posterior predictive distribution
- Without a posterior, we can only compute $p(\boldsymbol{y}_* = 1|\boldsymbol{x}_*, \theta)$ using a "single best" estimate of θ

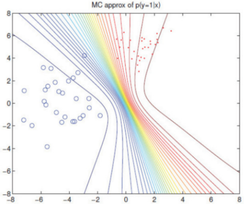
Averaging Over Posterior Distribution

- Can use the posterior distribution over parameters to compute "averaged prediction", e.g.,

$$p(\boldsymbol{y}_* = 1|\boldsymbol{x}_*, \mathbf{X}, \boldsymbol{y}) = \int p(\boldsymbol{y}_* = 1|\boldsymbol{x}_*, \theta)p(\theta|\mathbf{X}, \boldsymbol{y})d\theta$$

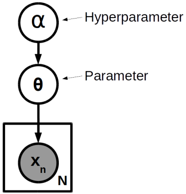
- $p(\boldsymbol{y}_* = 1|\boldsymbol{x}_*, \mathbf{X}, \boldsymbol{y})$ with θ "integrated out" is called posterior predictive distribution
- Without a posterior, we can only compute $p(\boldsymbol{y}_* = 1|\boldsymbol{x}_*, \theta)$ using a "single best" estimate of θ
- Averaging leads to more robust predictions (and prevents overfitting)





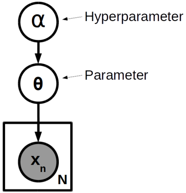
Hyperparameter Estimation

- Every model invariably has certain hyperparameters, e.g., regularization hyperparater in a linear regression model, or kernel hyperparameters in nonlinear regression of kernel SVM, etc.



Hyperparameter Estimation

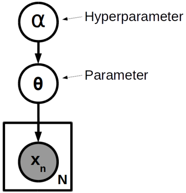
- Every model invariably has certain hyperparameters, e.g., regularization hyperparater in a linear regression model, or kernel hyperparameters in nonlinear regression of kernel SVM, etc.



- The probabilistic approach enables learning the hyperparam. from data (without cross-validation)

Hyperparameter Estimation

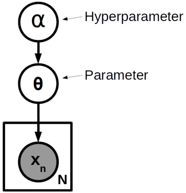
- Every model invariably has certain hyperparameters, e.g., regularization hyperparater in a linear regression model, or kernel hyperparameters in nonlinear regression of kernel SVM, etc.



- The probabilistic approach enables learning the hyperparam. from data (without cross-validation)
  - Can put priors on the hyperparameters and infer the posterior distribution

Hyperparameter Estimation

- Every model invariably has certain hyperparameters, e.g., regularization hyperparater in a linear regression model, or kernel hyperparameters in nonlinear regression of kernel SVM, etc.



- The probabilistic approach enables learning the hyperparam. from data (without cross-validation)
  - Can put priors on the hyperparameters and infer the posterior distribution
  - Can do point estimation for hyperparameters by maximizing the marginal likelihood

$$\hat{\alpha} = \arg\max_{\alpha} \log P(\mathbf{X}|\alpha)$$

Model Comparison

- Suppose we have a number of models to choose from
- Let's compute the posterior probability of each candidate model, again using Bayes rule

$$P(m|\mathbf{X}) = \frac{P(m)P(\mathbf{X}|m)}{P(\mathbf{X})}$$

Model Comparison

- Suppose we have a number of models to choose from
- Let's compute the posterior probability of each candidate model, again using Bayes rule

$$P(m|\mathbf{X}) = \frac{P(m)P(\mathbf{X}|m)}{P(\mathbf{X})}$$

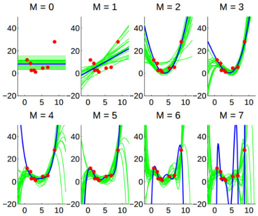
- Assuming each model is equally likey to be chosen *a priori*, we can ignore the prior $P(m)$

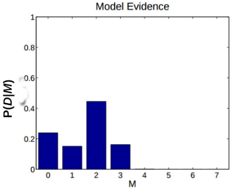
Model Comparison

- Suppose we have a number of models to choose from
- Let's compute the posterior probability of each candidate model, again using Bayes rule

$$P(m|\mathbf{X}) = \frac{P(m)P(\mathbf{X}|m)}{P(\mathbf{X})}$$

- Assuming each model is equally likey to be chosen *a priori*, we can ignore the prior $P(m)$
  - Just choose the model m that has the highest marginal likelihood $P(\mathbf{X}|m)$



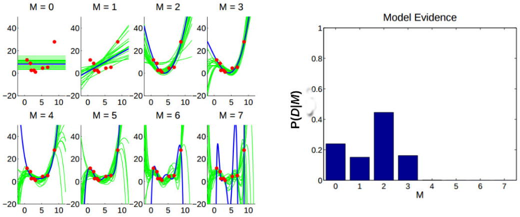


Model Comparison

- Suppose we have a number of models to choose from
- Let's compute the posterior probability of each candidate model, again using Bayes rule

$$P(m|\mathbf{X}) = \frac{P(m)P(\mathbf{X}|m)}{P(\mathbf{X})}$$

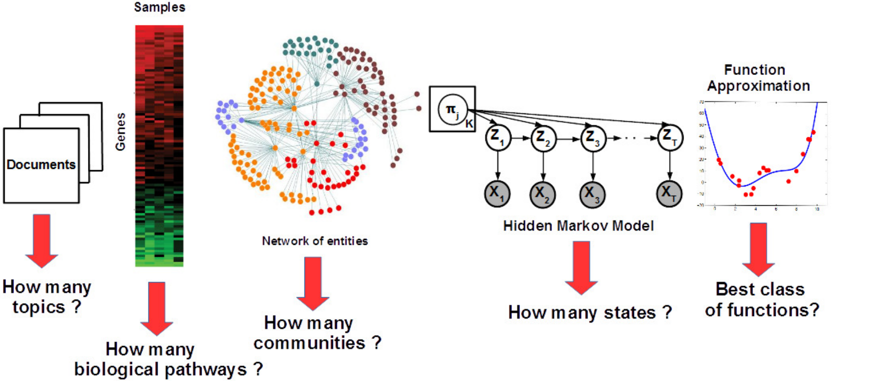
- Assuming each model is equally likey to be chosen *a priori*, we can ignore the prior $P(m)$
  - Just choose the model m that has the highest marginal likelihood $P(\mathbf{X}|m)$



- It doesn't require a cross-validation set (can be done even for unsupervised learning problems)

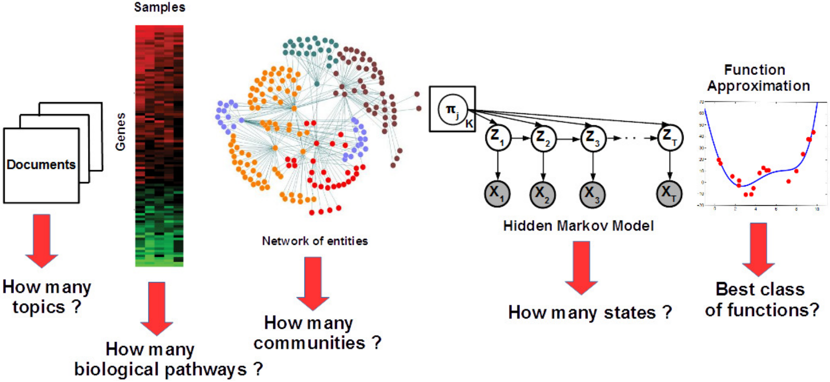
Nonparametric Bayesian Modeling

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



Nonparametric Bayesian Modeling

- **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)

Tentative List of Topics

- Basics of probabilistic modeling and estimation (in general, and through specific examples)
  - MLE and MAP Estimation
  - Fully Bayesian posterior inference
  - Exponential family distributions and their properties

Tentative List of Topics

- Basics of probabilistic modeling and estimation (in general, and through specific examples)
    - MLE and MAP Estimation
    - Fully Bayesian posterior inference
    - Exponential family distributions and their properties
- Probabilistic models for linear and nonlinear regression and classification
- Basics of probabilistic graphical models, structure learning and inference

Tentative List of Topics

- Basics of probabilistic modeling and estimation (in general, and through specific examples)
  - MLE and MAP Estimation
  - Fully Bayesian posterior inference
  - Exponential family distributions and their properties
- Probabilistic models for linear and nonlinear regression and classification
- Basics of probabilistic graphical models, structure learning and inference
- Generative latent variable models (LVMs) for data, and inference for LVMs

Tentative List of Topics

- Basics of probabilistic modeling and estimation (in general, and through specific examples)
  - MLE and MAP Estimation
  - Fully Bayesian posterior inference
  - Exponential family distributions and their properties
- Probabilistic models for linear and nonlinear regression and classification
- Basics of probabilistic graphical models, structure learning and inference
- Generative latent variable models (LVMs) for data, and inference for LVMs
- LVMs for unsupervised learning (clustering, dim. red.) and density estimation

Tentative List of Topics

- Basics of probabilistic modeling and estimation (in general, and through specific examples)
  - MLE and MAP Estimation
  - Fully Bayesian posterior inference
  - Exponential family distributions and their properties
- Probabilistic models for linear and nonlinear regression and classification
- Basics of probabilistic graphical models, structure learning and inference
- Generative latent variable models (LVMs) for data, and inference for LVMs
- LVMs for unsupervised learning (clustering, dim. red.) and density estimation
- LVMs for sequential and time-series data

Tentative List of Topics

- Basics of probabilistic modeling and estimation (in general, and through specific examples)
  - MLE and MAP Estimation
  - Fully Bayesian posterior inference
  - Exponential family distributions and their properties
- Probabilistic models for linear and nonlinear regression and classification
- Basics of probabilistic graphical models, structure learning and inference
- Generative latent variable models (LVMs) for data, and inference for LVMs
- LVMs for unsupervised learning (clustering, dim. red.) and density estimation
- LVMs for sequential and time-series data
- Introduction to approximate Bayesian inference (MCMC and variational methods)

Tentative List of Topics

- Basics of probabilistic modeling and estimation (in general, and through specific examples)
  - MLE and MAP Estimation
  - Fully Bayesian posterior inference
  - Exponential family distributions and their properties
- Probabilistic models for linear and nonlinear regression and classification
- Basics of probabilistic graphical models, structure learning and inference
- Generative latent variable models (LVMs) for data, and inference for LVMs
- LVMs for unsupervised learning (clustering, dim. red.) and density estimation
- LVMs for sequential and time-series data
- Introduction to approximate Bayesian inference (MCMC and variational methods)
- Deep Generative Models (Variational Auto-encoders and Generative Adversarial Networks)

Tentative List of Topics

- Basics of probabilistic modeling and estimation (in general, and through specific examples)
  - MLE and MAP Estimation
  - Fully Bayesian posterior inference
  - Exponential family distributions and their properties
- Probabilistic models for linear and nonlinear regression and classification
- Basics of probabilistic graphical models, structure learning and inference
- Generative latent variable models (LVMs) for data, and inference for LVMs
- LVMs for unsupervised learning (clustering, dim. red.) and density estimation
- LVMs for sequential and time-series data
- Introduction to approximate Bayesian inference (MCMC and variational methods)
- Deep Generative Models (Variational Auto-encoders and Generative Adversarial Networks)
- Other advanced topics based on students' interest

Course Goals

- Learn the basics of probabilistic modeling of data
- Learn how to formulate an ML problem as a probabilistic model
- Learn how to do inference for estimating the unknowns in the model
- Learn how to implement such models (both from scratch, as well as using support from some existing software frameworks)
- Use the course project to reinforce the topics you learn in the class, and explore other topics beyond what is covered in the class

Announcement

- Need to re-schedule (prepone) the Aug 8 lecture to Aug 5 (6pm RM-101)
- Need to re-schedule (postpone) the Aug 10 lecture to Aug 14 (6pm RM-101)
- Apologies for the re-schedule. Just a one-time thing.
- A probability refresher tutorial on Aug 8 by one of the TAs (6pm RM-101)