CS6190 Probabilistic Modeling

Fall 2019

Instructor: Shandian Zhe
zhe@cs.utah.edu
School of Computing
Shandian Zhe: Probabilistic Machine Learning

Assistant Professor, School of Computing, University of Utah

zhe@cs.utah.edu

Research Topics:
1. Bayesian Nonparametrics
2. Bayesian Deep Learning
3. Probabilistic Graphical Models
4. Large-Scale Learning System
5. Tensor/Matrix Factorization
6. Embedding Learning

Applications:
• Collaborative Filtering
• Online Advertising
• Physical Simulation
• Brain Imaging Data Analysis

...
Outline

• What is probabilistic machine learning
• Why probabilistic/Bayesian machine learning
• Course requirements/policies (homework assignments, projects, final exams, etc.)
• Basic knowledge review
What is machine learning

“A computer program is said to learn from experience $E$ with respect to some class of tasks $T$ and performance measure $P$, if its performance at tasks in $T$, as measured by $P$, improves with experience $E$.”

Tom Mitchell (1999)
Machine learning is the driving force of AI
Alpha-Go!  A ML algorithm rather AI
Machine learning is everywhere!

And you are probably already using it
Machine learning is everywhere!

And you are probably already using it

• Is an email spam?
• Find all the people in this photo
• If I like these three movies, what should I watch next?
• Based on your purchase history, you might be interested in...
• Will a stock price go up or down tomorrow? By how much?
• Handwriting recognition
• What are the best ads to place on this website?
• I would like to read that Dutch website in English
• Ok Google, Drive this car for me. And, fly this helicopter for me.
• Does this genetic marker correspond to Alzheimer’s disease?
What is probabilistic (Bayesian) learning?

In a nutshell, probabilistic learning is branch of ML that uses **probabilistic (or Bayesian) principles** for model design and algorithm development.
Probabilistic Learning

Prior distribution

\[ p(\theta) \]

Data likelihood

\[ p(D|\theta) \]

Posterior distribution

\[ p(\theta|D) \]

Bayes’s Rule

\[
p(\theta|D) = \frac{p(\theta, D)}{p(D)} = \frac{p(\theta)p(D|\theta)}{\int p(\theta)p(D|\theta)d\theta}
\]
Advantage

- Unified, principled mathematical framework

\[ \theta \sim p(\theta) \quad \text{D}|\theta \sim p(\text{D}|\theta) \quad \Rightarrow \quad p(\theta|\text{D}) \]

- Uncertainty reasoning

Asthma: 60%
Heart disease: 30%
Healthy: 10%
Raining: 70%
Sunny: 30%
How important is the uncertainty?

Tesla death smash probe: Neither driver nor autopilot saw the truck
Challenges

• Modeling

Complex Knowledge/assumptions

• Calculation

$$p(\theta|D) = \frac{p(\theta)p(D|\theta)}{\int p(\theta)p(D|\theta) d\theta}$$

Valid Prior Distributions

- MCMC sampling
- Variational approximations
- Belief propagation

High dimensional integration

Rules
1. You can....
2. You can’t...
3. You can....
4. You can’t
In this course

- We will cover both the classical and state-of-the-art approaches to deal with these challenges.
Overview of this course

Syllabus

https://www.cs.utah.edu/~zhe/teach/cs6190.html
Warning

This course is \textit{math intensive} and requires \textit{nontrivial} programming (with Matlab, R or Python). Python components may require TensorFlow and/or PyTorch. The coding workload is not heavy, but requires \textit{mathematical derivations and careful debugging}. 
How will you learn?

• Take classes to follow the math, understand the models and algorithms
• **Derive the math details by yourself!**
• Finish the homework assignments to deepen your understanding
• **Implement and debug the models and algorithms by yourself!**
• Course project to enlarge your vision and practice your capability
This course

Focuses on the **mathematic foundations, modeling and algorithmic ideas** in probabilistic learning

This course is **not** about

• Using a specific machine learning tools, e.g., scikit-learn
• How to use Python, R or Matlab
We assume that

• You are not scared of math, statistics, calculus and calculations
• Your are comfortable with matrices operations
• You can pick-up Matlab/Python/R very quickly (even if you have never used them before)
• You enjoy debugging, step in, step out, print, etc.
• You can quickly learn how to use TensorFlow and PyTorch by following the documentation and searching for the online examples
If you feel NOT right about these assumptions

• Seriously consider whether to take this course
Course information

• The course website contains all the detailed information
• The course website is linked to my homepage

My home page  http://www.cs.utah.edu/~zhe/

Course website  https://www.cs.utah.edu/~zhe/teach/cs6190.html
Basic Knowledge Review for Convex Functions and Matrix Derivatives
Basic Knowledge Review

- Convex region/set
Basic Knowledge Review

- Convex function \( f: X \rightarrow R \)

- The input domain \( X \) is a convex region/set

\[
\forall x_1, x_2 \in X, \forall t \in [0, 1] : \quad f(tx_1 + (1-t)x_2) \leq tf(x_1) + (1-t)f(x_2).
\]
Basic Knowledge Review

• Examples of convex functions

<table>
<thead>
<tr>
<th>Single variable</th>
<th>multivariable</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f(x) = e^x$</td>
<td>$f(x) = \mathbf{a}^\top \mathbf{x} + b$</td>
</tr>
<tr>
<td>$f(x) = -\log(x)$</td>
<td>$f(x) = \frac{1}{2} \mathbf{x}^\top \mathbf{x}$</td>
</tr>
</tbody>
</table>

• How to determine a convex function?

When differentiable

\[ f(x) \geq f(y) + \nabla f(y)^\top (x - y) \]

When twice differentiable

\[ \nabla^2 f(x) \succeq 0 \]
Basic Knowledge Review

• Jensen’s inequality (for convex function)

When $X$ is random variable

$$f\left( E(X) \right) \leq E\left( f(X) \right)$$

$$f\left( E(g(X)) \right) \leq E\left( f(g(X)) \right)$$
Basic Knowledge Review

• Convex conjugate (Fenchel Conjugate)

for an arbitrary convex function \( f(\cdot) \), there exists a duality function \( g(\cdot) \)

\[
\begin{align*}
  f(x) &= \max_{\lambda} \lambda x - g(\lambda) \\
  g(\lambda) &= \max_{x} \lambda x - f(x)
\end{align*}
\]

Jensen’s equality and convex conjugate plays the key role in approximate inference
Basic Knowledge Review

• Matrix derivative: everything comes from the definition of the differential

\[ y(x + dx) = y(x) + A \, dx + \text{(high order terms)} \]

Derivative, a.k.a Jacobian

In general

\[ y : m \times 1 \quad x : n \times 1 \quad A : m \times n \]

\[ \partial y = A \, dx \quad \frac{\partial y}{\partial x} = A \]

Some books use \( A^\top \) instead, because they perform the chain rule from right to left
Basic Knowledge Review

• Commonly used results

\[
\begin{align*}
\partial A &= 0 \quad (A \text{ is a constant}) \\
\partial (\alpha X) &= \alpha \partial X \\
\partial (X + Y) &= \partial X + \partial Y \\
\partial (\text{Tr}(X)) &= \text{Tr}(\partial X) \\
\partial (XY) &= (\partial X)Y + X(\partial Y) \\
\partial (X \circ Y) &= (\partial X) \circ Y + X \circ (\partial Y) \\
\partial (X \otimes Y) &= (\partial X) \otimes Y + X \otimes (\partial Y) \\
\partial (X^{-1}) &= -X^{-1}(\partial X)X^{-1} \\
\partial (\det(X)) &= \det(X)\text{Tr}(X^{-1}\partial X) \\
\partial (\ln(\det(X))) &= \text{Tr}(X^{-1}\partial X) \\
\partial X^T &= (\partial X)^T \\
\partial X^H &= (\partial X)^H
\end{align*}
\]

An excellent note: [Old and New Matrix Algebra Useful for Statistics](https://people.reed.edu/~minka/notes.pdf), By Tom Minka, 2001

Another excellent reference: [Matrix Cookbook](https://matrixcookbook.com)

Hint: Decompose the derivatives into simpler terms, and look up the results from the above references.