Introduction to Machine Learning and Probabilistic Modeling

Piyush Rai

Probabilistic Machine Learning (CS772A)

Dec 30, 2015

Course Logistics

- **Course website:** `http://www.cse.iitk.ac.in/users/piyush/courses/pml_winter16/PML.html`
- **Instructor:** Piyush Rai (`http://www.cse.iitk.ac.in/users/piyush/`)
- **TAs:** Milan Someswar, Vinit Tiwari, Rahul Kumar Patidar
- **Discussion site:** `https://piazza.com/iitk.ac.in/secondsemester2016/cs772a/`
- **Background assumed:** basics of linear algebra, multivariate calculus, probability and statistics, optimization, programming (MATLAB, R, Python).

Course Logistics

- **Course website:** http://www.cse.iitk.ac.in/users/piyush/courses/pml_winter16/PML.html
- **Instructor:** Piyush Rai (http://www.cse.iitk.ac.in/users/piyush/)
- **TAs:** Milan Someswar, Vinit Tiwari, Rahul Kumar Patidar
- **Discussion site:** https://piazza.com/iitk.ac.in/secondsemester2016/cs772a/
- **Background assumed:** basics of linear algebra, multivariate calculus, probability and statistics, optimization, programming (MATLAB, R, Python).
- **Grading:**

Course Logistics

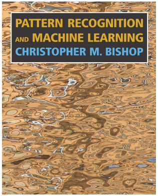
- **Course website:** http://www.cse.iitk.ac.in/users/piyush/courses/pml_winter16/PML.html
- **Instructor:** Piyush Rai (http://www.cse.iitk.ac.in/users/piyush/)
- **TAs:** Milan Someswar, Vinit Tiwari, Rahul Kumar Patidar
- **Discussion site:** https://piazza.com/iitk.ac.in/secondsemester2016/cs772a/
- **Background assumed:** basics of linear algebra, multivariate calculus, probability and statistics, optimization, programming (MATLAB, R, Python).
- **Grading:**
  - 3 homework assignments: 30%, Midterm exam: 20%, Final exam: 20%
  - Project: 30% (to be done in groups of 3 students)

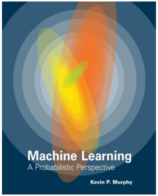
Course Logistics

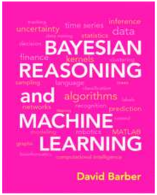
- **Course website:** http://www.cse.iitk.ac.in/users/piyush/courses/pml_winter16/PML.html
- **Instructor:** Piyush Rai (http://www.cse.iitk.ac.in/users/piyush/)
- **TAs:** Milan Someswar, Vinit Tiwari, Rahul Kumar Patidar
- **Discussion site:** https://piazza.com/iitk.ac.in/secondsemester2016/cs772a/
- **Background assumed:** basics of linear algebra, multivariate calculus, probability and statistics, optimization, programming (MATLAB, R, Python).
- **Grading:**
  - 3 homework assignments: 30%, Midterm exam: 20%, Final exam: 20%
  - Project: 30% (to be done in groups of 3 students)
  - **Note:** A really awesome project (e.g., publishable piece of work) may help you automatically get an A grade. You may propose your own project or talk to me for ideas. The project has to be (at least loosely) related to probabilistic ML. More details coming soon.

Books

Some books with a bent towards *probabilistic* machine learning:

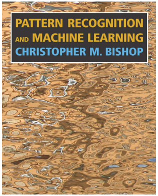


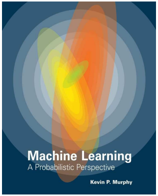


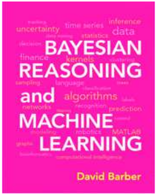


Books

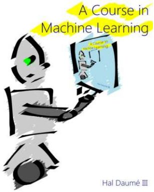
Some books with a bent towards *probabilistic* machine learning:

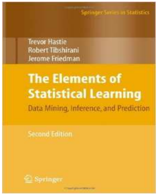


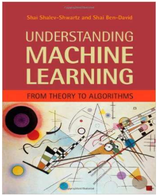




Some other books on machine learning:

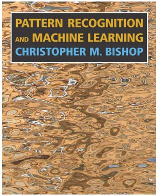


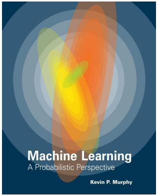


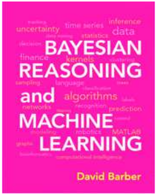


Books

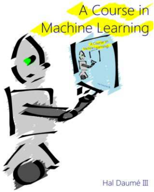
Some books with a bent towards *probabilistic* machine learning:

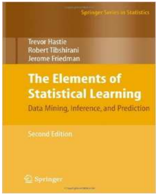


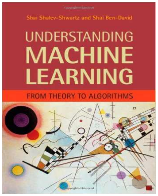




Some other books on machine learning:







Not shown: many excellent books on special topics (kernel methods, online learning, Bayesian learning, deep learning, etc.). Ask me if you want to know.

Intro to Machine Learning

Machine Learning

- Creating programs that can automatically learn rules from data

  "Field of study that gives computers the ability to learn without being explicitly programmed" (Arthur Samuel, 1959)

Machine Learning

- Creating programs that can automatically learn rules from data

  "Field of study that gives computers the ability to learn without being explicitly programmed" (Arthur Samuel, 1959)

- Traditional way: Write programs using hard-coded (fixed) rules

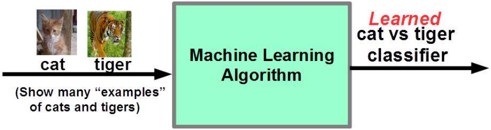
Machine Learning

- Creating programs that can automatically learn rules from data

  "Field of study that gives computers the ability to learn without being explicitly programmed" (Arthur Samuel, 1959)

- Traditional way: Write programs using hard-coded (fixed) rules

- Machine Learning (ML): Learn rules by looking at the data



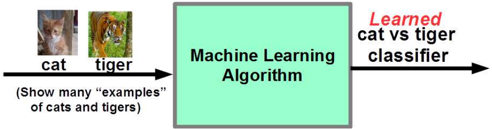
Machine Learning

- Creating programs that can automatically learn rules from data

  "Field of study that gives computers the ability to learn without being explicitly programmed" (Arthur Samuel, 1959)

- Traditional way: Write programs using hard-coded (fixed) rules

- Machine Learning (ML): Learn rules by looking at the data



- Learned rules must generalize (do well) on future "test" data (idea of generalization; more on this later)

Machine Learning in the real-world

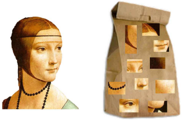
Broadly applicable in many domains (e.g., finance, robotics, bioinformatics, computer vision, NLP, databases, systems, etc.). Some applications:

- Information retrieval (text, visual, and multimedia searches)
- Machine Translation
- Question Answering
- Social networks
- Recommender systems (Amazon, Netflix, etc.)
- Speech/handwriting/object recognition
- Ad placement on websites
- Credit-card fraud detection
- Weather prediction
- Autonomous vehicles (self-driving cars)
- Healthcare and life-sciences
- .. and many more applications in sciences and engineering

Data and *Data Representation*..

Data Representation

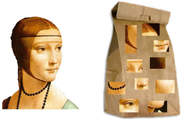
- ML algorithms work with data represented as a set of features/attributes
- One popular representation: **bag-of-features**



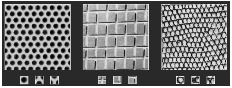
Picture courtesy: Svetlana Lazebnik

Data Representation

- ML algorithms work with data represented as a set of features/attributes
- One popular representation: **bag-of-features**



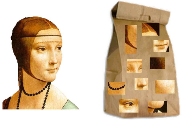
- The idea: Decide features to represent data (becomes our feature vocabulary)



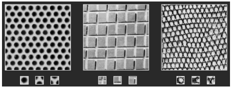
Picture courtesy: Svetlana Lazebnik

Data Representation

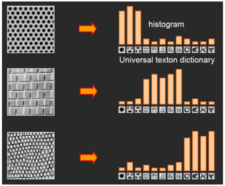
- ML algorithms work with data represented as a set of features/attributes
- One popular representation: **bag-of-features**



- The idea: Decide features to represent data (becomes our feature vocabulary)



- Now represent each example using the frequency of each feature



Picture courtesy: Svetlana Lazebnik

Data Representation

Another example: representing text data. Consider the following sentences:

- John likes to watch movies
- Mary likes movies too
- John also likes football

The feature vocabulary consists of 8 unique words

Data Representation

Another example: representing text data. Consider the following sentences:

- John likes to watch movies
- Mary likes movies too
- John also likes football

The feature vocabulary consists of 8 unique words

Here is the **bag-of-words** feature representation of these 3 sentences

$$
\begin{array}{l}
\\
\text{Sentence 1} \\
\text{Sentence 2} \\
\text{Sentence 3}
\end{array}
\begin{pmatrix}
\text{John} & \text{likes} & \text{to} & \text{watch} & \text{movies} & \text{Mary} & \text{too} & \text{also} & \text{football} \\
1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1
\end{pmatrix}
$$

Data Representation

Another example: representing text data. Consider the following sentences:

- John likes to watch movies
- Mary likes movies too
- John also likes football

The feature vocabulary consists of 8 unique words

Here is the **bag-of-words** feature representation of these 3 sentences

$$\begin{array}{l} \\ \text{Sentence 1} \\ \text{Sentence 2} \\ \text{Sentence 3} \end{array} \begin{pmatrix} \text{John} & \text{likes} & \text{to} & \text{watch} & \text{movies} & \text{Mary} & \text{too} & \text{also} & \text{football} \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

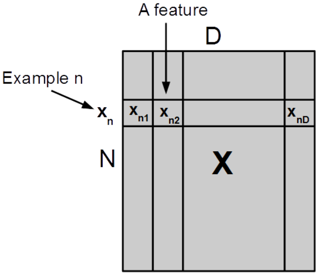
Note: Not necessarily the most optimal/most expressive feature representation

Feature representation learning is a very active area of research in ML (there is even a dedicated conference on this topic: ICLR)

Data Representation

We will (usually) assume that:

- $\mathbf{X}$ denotes data in form of an $N \times D$ feature matrix
- N examples, D features to represent each example
- Each row is an example, each column is a feature
- $\boldsymbol{x}_n$ denotes the n-th example (a vector of length D)

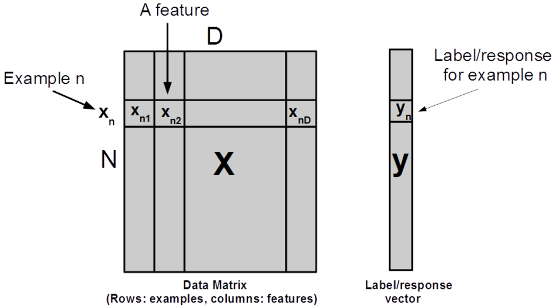


Data Matrix
(Rows: examples, columns: features)

Data Representation

We will (usually) assume that:

- $\mathbf{X}$ denotes data in form of an $N \times D$ feature matrix
- N examples, D features to represent each example
- Each row is an example, each column is a feature
- $\boldsymbol{x}_n$ denotes the n-th example (a vector of length D)



- $\boldsymbol{y}$ denotes labels/responses in form of an $N \times 1$ label/response vector
- y_n denotes label/response of the n-th example $\boldsymbol{x}_n$

Types of Machine Learning problems..

Supervised Learning

Supervised Learning

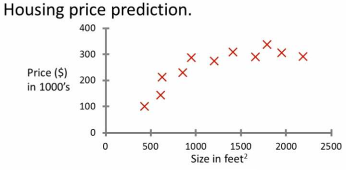
- Given: Training data as labeled examples $\{(x_1, y_1), \ldots, (x_N, y_N)\}$
- Goal: Learn a rule ("function" $f : x \rightarrow y$) to predict outputs y from inputs x

Supervised Learning

- Given: Training data as labeled examples $\{(x_1, y_1), \ldots, (x_N, y_N)\}$
- Goal: Learn a rule ("function" $f : x \to y$) to predict outputs y from inputs x
- Output y (label/response) can usually be:

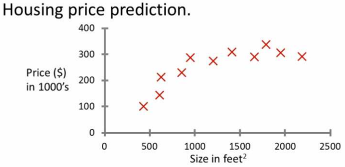
Supervised Learning

- Given: Training data as labeled examples $\{(x_1, y_1), \ldots, (x_N, y_N)\}$
- Goal: Learn a rule ("function" $f : x \rightarrow y$) to predict outputs y from inputs x
- Output y (label/response) can usually be:
  - **Continuous-/real-valued** (Regression problem). Example: when y is the price of a stock, price of a house, USD/rupee conversion rate, etc.

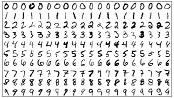


Supervised Learning

- Given: Training data as labeled examples $\{(x_1, y_1), \ldots, (x_N, y_N)\}$
- Goal: Learn a rule ("function" $f : x \to y$) to predict outputs y from inputs x
- Output y (label/response) can usually be:
  - **Continuous-/real-valued** (Regression problem). Example: when y is the price of a stock, price of a house, USD/rupee conversion rate, etc.

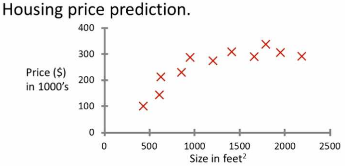


  - **Discrete-valued** (Classification problem): Example: when y is the binary 0/1 label (spam/normal) of an email, label (0-9) of a handwritten digit, etc.

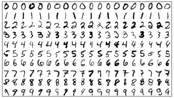


Supervised Learning

- Given: Training data as labeled examples $\{(x_1, y_1), \ldots, (x_N, y_N)\}$
- Goal: Learn a rule ("function" $f : x \to y$) to predict outputs y from inputs x
- Output y (label/response) can usually be:
  - **Continuous-/real-valued** (Regression problem). Example: when y is the price of a stock, price of a house, USD/rupee conversion rate, etc.



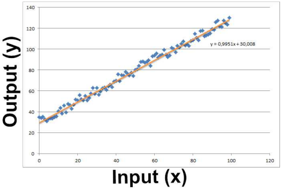
  - **Discrete-valued** (Classification problem): Example: when y is the binary 0/1 label (spam/normal) of an email, label (0-9) of a handwritten digit, etc.

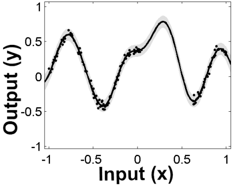


- Many other variants (structured-prediction, multi-label learning, ordinal regression, ranking, etc.), depending on the type of label y

Supervised Learning: Pictorially

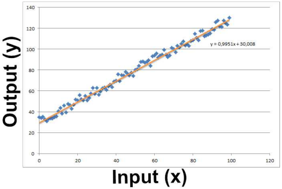
- Regression (linear/nonlinear): fitting a line/curve

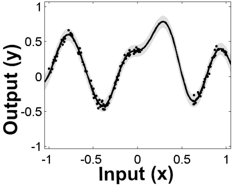




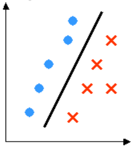
Supervised Learning: Pictorially

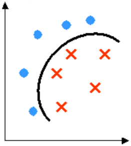
- Regression (linear/nonlinear): fitting a line/curve





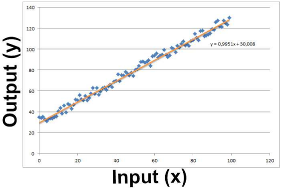
- Classification (linear/nonlinear): finding a separator

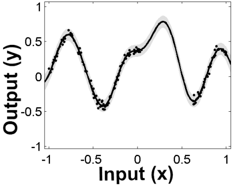




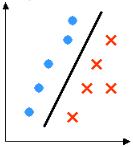
Supervised Learning: Pictorially

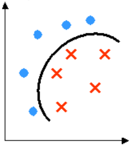
- Regression (linear/nonlinear): fitting a line/curve





- Classification (linear/nonlinear): finding a separator

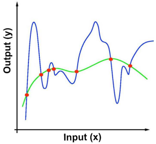




- Generalization is crucial (must do well on test data)

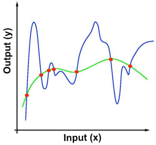
Generalization

- Simple hypotheses/rules are preferred over more complex ones

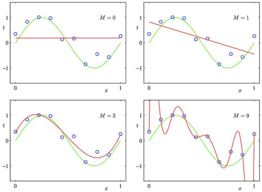


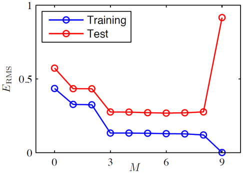
Generalization

- Simple hypotheses/rules are preferred over more complex ones



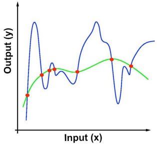
- Simple hypotheses/rules tend to generalize better



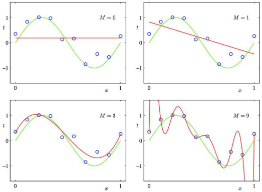


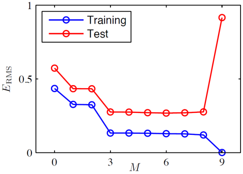
Generalization

- Simple hypotheses/rules are preferred over more complex ones



- Simple hypotheses/rules tend to generalize better





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

Unsupervised Learning

Unsupervised Learning

- Given: Training data in form of unlabeled examples $\{x_1, \ldots, x_N\}$
- Goal: Learn the instrinsic *structure* in the data.

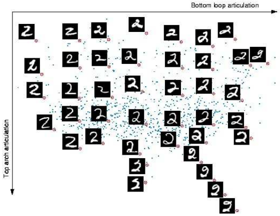
Unsupervised Learning

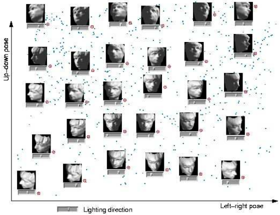
- Given: Training data in form of unlabeled examples $\{x_1, \ldots, x_N\}$
- Goal: Learn the instrinsic *structure* in the data. Examples:
    - Data clustering (grouping similar things together)

Unsupervised Learning

- Given: Training data in form of unlabeled examples $\{x_1, \ldots, x_N\}$
- Goal: Learn the instrinsic *structure* in the data. Examples:
    - Data clustering (grouping similar things together)

    - Dimensionality reduction, embedding, or manifold learning

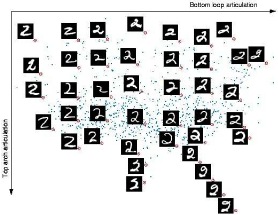


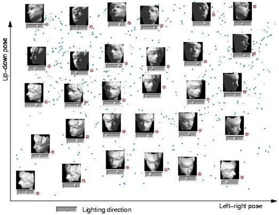


Unsupervised Learning

- Given: Training data in form of unlabeled examples $\{x_1, \ldots, x_N\}$
- Goal: Learn the instrinsic *structure* in the data. Examples:
  - Data clustering (grouping similar things together)

  - Dimensionality reduction, embedding, or manifold learning

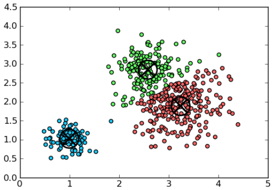




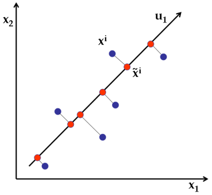
- Also very useful for summarizing/compressing data. Often also used as a preprocessing step for many supervised learning algorithms (e.g., to extract good features, to speed up the algorithms, etc.)

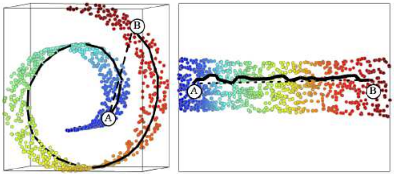
Unsupervised Learning: Pictorially

- Clustering: Find some "centers" and assign each data point to its "closest" center



- Dimensionality reduction: Find a lower-dimensional subspace that the data approximately lives on





Other popular Machine Learning paradigms

Semi-supervised Learning

- Learning with labeled+unlabeled data

Semi-supervised Learning

- Learning with labeled+unlabeled data
- Why is Semi-supervised Learning useful?

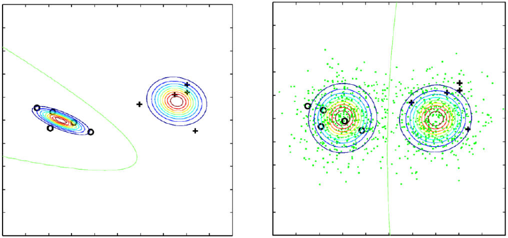
Semi-supervised Learning

- Learning with labeled+unlabeled data
- Why is Semi-supervised Learning useful?
  - Labeled data is expensive. Unlabeled data comes (almost) for free!

Semi-supervised Learning

- Learning with labeled+unlabeled data
- Why is Semi-supervised Learning useful?
  - Labeled data is expensive. Unlabeled data comes (almost) for free!
  - Unlabeled data can provide valuable information about the distribution of data (e.g., where might the low-density regions or the class separator lie)

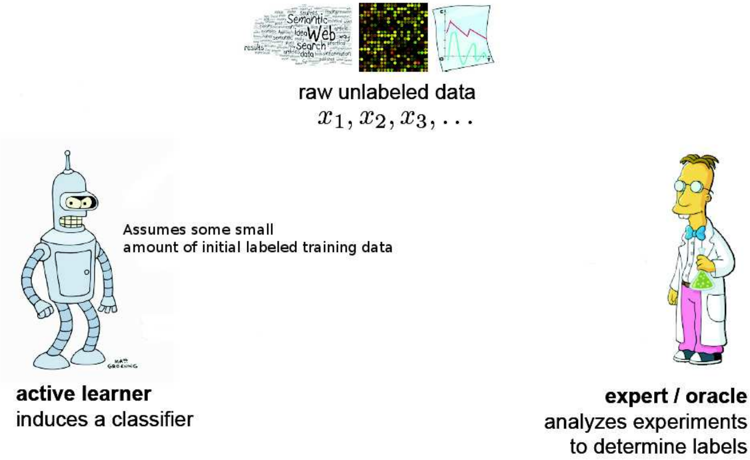
only labeled data with unlabeled data



from [Semi-Supervised Learning, ICML 2007 Tutorial; Xiaojin Zhu]

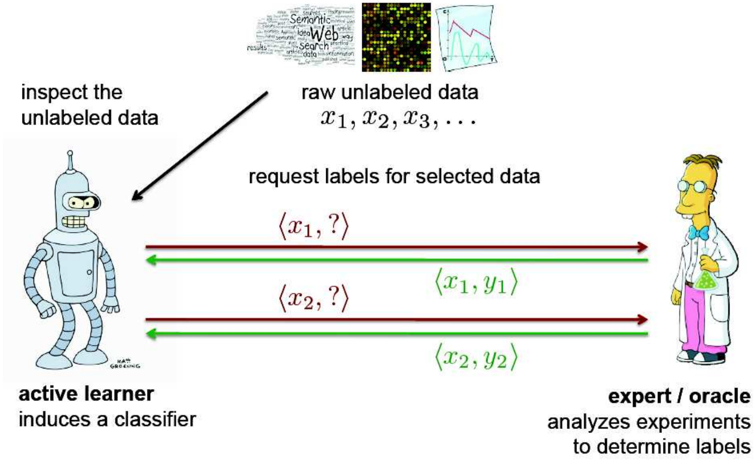
Active Learning

- The learner can interactively ask for labels of most informative examples



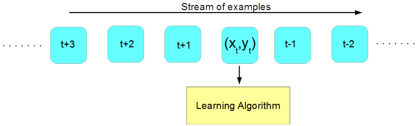
Active Learning

- The learner can interactively ask for labels of most informative examples



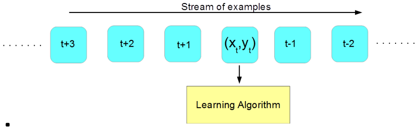
Some Other Learning Paradigms

- **Online Learning**
  - Learning with one example (or a small minibatch of examples) at a time

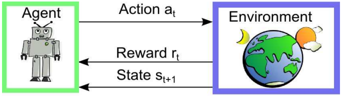


Some Other Learning Paradigms

- **Online Learning**
  - Learning with one example (or a small minibatch of examples) at a time



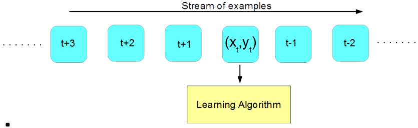
- **Reinforcement Learning**
  - Learning a "policy" by performing actions and getting rewards



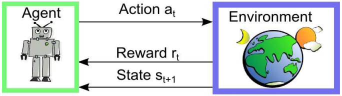
Reinforcement Learning Setup

Some Other Learning Paradigms

- **Online Learning**
  - Learning with one example (or a small minibatch of examples) at a time



- **Reinforcement Learning**
  - Learning a "policy" by performing actions and getting rewards



Reinforcement Learning Setup

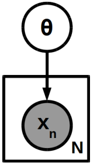
- **Transfer/Multitask Learning**
  - Leveraging knowledge of solving one problem to solve a new problem

On to Probabilistic Machine Learning..

Machine Learning via Probabilistic Modeling

- Assume data $\mathbf{X} = \{\boldsymbol{x}_n\}_{n=1}^N$ generated from a probability distribution $p(\boldsymbol{x}|\theta)$, in an i.i.d. (independent and identically distributed) fashion

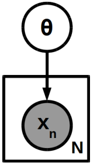
$$\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N \sim p(\boldsymbol{x}|\theta)$$



Machine Learning via Probabilistic Modeling

- Assume data $\mathbf{X} = \{\boldsymbol{x}_n\}_{n=1}^N$ generated from a probability distribution $p(\boldsymbol{x}|\theta)$, in an i.i.d. (independent and identically distributed) fashion

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

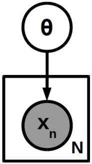


- The form of $p(\boldsymbol{x}|\theta)$ (also called likelihood) depends on the type of the data

Machine Learning via Probabilistic Modeling

- Assume data $\mathbf{X} = \{\boldsymbol{x}_n\}_{n=1}^N$ generated from a probability distribution $p(\boldsymbol{x}|\theta)$, in an i.i.d. (independent and identically distributed) fashion

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

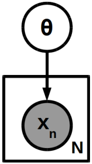


- The form of $p(\boldsymbol{x}|\theta)$ (also called likelihood) depends on the type of the data
- Assumptions about parameter θ can be encoded via a prior distribution $p(\theta)$
    - Also corresponds to imposing a regularizer over θ (helps in generalization)

Machine Learning via Probabilistic Modeling

- Assume data $\mathbf{X} = \{\boldsymbol{x}_n\}_{n=1}^{N}$ generated from a probability distribution $p(\boldsymbol{x}|\theta)$, in an i.i.d. (independent and identically distributed) fashion

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

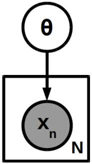


- The form of $p(\boldsymbol{x}|\theta)$ (also called likelihood) depends on the type of the data
- Assumptions about parameter θ can be encoded via a prior distribution $p(\theta)$
  - Also corresponds to imposing a regularizer over θ (helps in generalization)
- **Goal:** To estimate parameter θ, given data $\mathbf{X}$

Machine Learning via Probabilistic Modeling

- Assume data $\mathbf{X} = \{\boldsymbol{x}_n\}_{n=1}^N$ generated from a probability distribution $p(\boldsymbol{x}|\theta)$, in an i.i.d. (independent and identically distributed) fashion

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

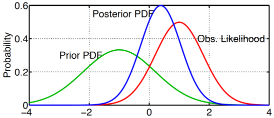


- The form of $p(\boldsymbol{x}|\theta)$ (also called likelihood) depends on the type of the data
- Assumptions about parameter θ can be encoded via a prior distribution $p(\theta)$
  - Also corresponds to imposing a regularizer over θ (helps in generalization)
- **Goal:** To estimate parameter θ, given data $\mathbf{X}$
- Variations of this general view subsume most machine learning problems
  - Regression, classification, clustering, dimensionality reduction, etc.

Parameter Estimation

- Can use Bayes rule to estimate the posterior **distribution** over parameters

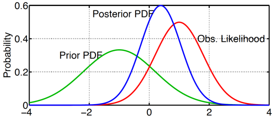
$$p(\theta|\mathbf{X}) = \frac{p(\mathbf{X}|\theta)p(\theta)}{p(\mathbf{X})} = \frac{\text{Likelihood} \times \text{Prior}}{\text{Marginal likelihood}}$$



Parameter Estimation

- Can use Bayes rule to estimate the posterior **distribution** over parameters

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

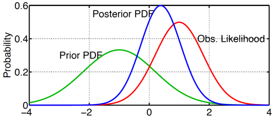


- .. or find the **single "best" estimate** of the parameters via optimization

Parameter Estimation

- Can use Bayes rule to estimate the posterior **distribution** over parameters

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



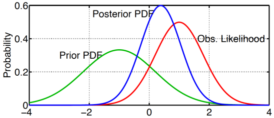
- .. or find the **single "best" estimate** of the parameters via optimization
  - Maximum likelihood estimation (MLE)

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

Parameter Estimation

- Can use Bayes rule to estimate the posterior **distribution** over parameters

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



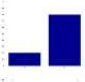
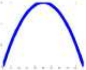
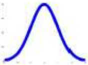
- .. or find the **single "best" estimate** of the parameters via optimization
  - Maximum likelihood estimation (MLE)

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

  - Maximum-a-Posteriori (MAP) estimation

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

Some common probability distributions

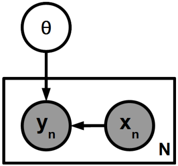
Distribution	Domain	Picture	Parametric Form
Binomial	Binary		$Bin(x \mid N, \theta) \propto \theta^{n} (1-\theta)^{N-n}$
Multinomial	K classes		$Mult(\bar{x} \mid \bar{\theta}) \propto \prod \theta_k^{x_k}$
Beta	[0,1]		$Beta(\theta \mid \alpha, \beta) \propto \theta^{\alpha-1} (1-\theta)^{\beta-1}$
Gamma	$[0, \infty)$		$Gam(x \mid a, b) \propto x^{-a-1} \exp(-bx)$
Dirichlet	Simplex		$Dir(\bar{\theta} \mid \bar{\alpha}) \propto \prod \theta_k^{\alpha_k - 1}$
Gaussian	Reals		$Nor(x \mid \mu, \sigma^2) \propto \exp\left((x-\mu)^2 / 2\sigma^2\right)$

Some Examples of Probabilistic Modeling in Machine Learning

Probabilistic Supervised Learning

- Consider regression/classification. Training data $\{(\boldsymbol{x}_1, \boldsymbol{y}_1), \ldots, (\boldsymbol{x}_N, \boldsymbol{y}_N)\}$
- Goal: Learn a function to predict outputs $\boldsymbol{y}$ from inputs $\boldsymbol{x}$
- Model the output/response/label as a probability distribution

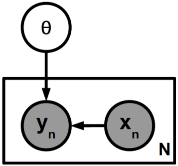
$$\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N \sim p(\boldsymbol{y}|\boldsymbol{x}, \theta)$$



Probabilistic Supervised Learning

- Consider regression/classification. Training data $\{(\boldsymbol{x}_1, \boldsymbol{y}_1), \ldots, (\boldsymbol{x}_N, \boldsymbol{y}_N)\}$
- Goal: Learn a function to predict outputs $\boldsymbol{y}$ from inputs $\boldsymbol{x}$
- Model the output/response/label as a probability distribution

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

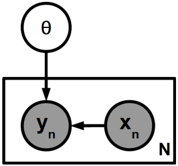


- Learning involves estimating the parameter θ given data $\{\boldsymbol{x}_n, \boldsymbol{y}_n\}_{n=1}^N$

Probabilistic Supervised Learning

- Consider regression/classification. Training data $\{(\boldsymbol{x}_1, \boldsymbol{y}_1), \ldots, (\boldsymbol{x}_N, \boldsymbol{y}_N)\}$
- Goal: Learn a function to predict outputs $\boldsymbol{y}$ from inputs $\boldsymbol{x}$
- Model the output/response/label as a probability distribution

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

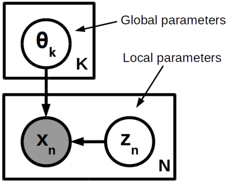


- Learning involves estimating the parameter θ given data $\{\boldsymbol{x}_n, \boldsymbol{y}_n\}_{n=1}^N$
- Can now make probabilistic predictions for new data $\boldsymbol{x}_*$ using θ

$$p(\boldsymbol{y}_*|\boldsymbol{x}_*, \theta) \quad \text{or} \quad p(\boldsymbol{y}_*|\boldsymbol{x}_*)$$

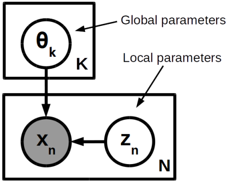
Probabilistic Unsupervised Learning

- Consider clustering or dimensionality reduction problems
- Each data point $\boldsymbol{x}_n$ assumed to be generated via some latent variable $\boldsymbol{z}_n$ and parameters θ



Probabilistic Unsupervised Learning

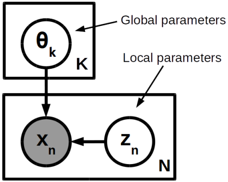
- Consider clustering or dimensionality reduction problems
- Each data point $\boldsymbol{x}_n$ assumed to be generated via some latent variable $\boldsymbol{z}_n$ and parameters θ



  - Clustering: $\boldsymbol{z}_n$ denotes which cluster $\boldsymbol{x}_n$ belongs to

Probabilistic Unsupervised Learning

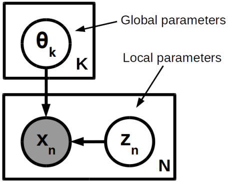
- Consider clustering or dimensionality reduction problems
- Each data point $\boldsymbol{x}_n$ assumed to be generated via some latent variable $\boldsymbol{z}_n$ and parameters θ



  - Clustering: $\boldsymbol{z}_n$ denotes which cluster $\boldsymbol{x}_n$ belongs to
  - Dimensionality Reduction: $\boldsymbol{z}_n$ represents the compressed representation of $\boldsymbol{x}_n$

Probabilistic Unsupervised Learning

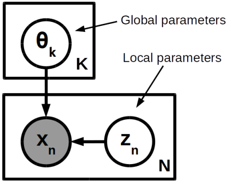
- Consider clustering or dimensionality reduction problems
- Each data point $\boldsymbol{x}_n$ assumed to be generated via some latent variable $\boldsymbol{z}_n$ and parameters θ



  - Clustering: $\boldsymbol{z}_n$ denotes which cluster $\boldsymbol{x}_n$ belongs to
  - Dimensionality Reduction: $\boldsymbol{z}_n$ represents the compressed representation of $\boldsymbol{x}_n$
  - Parameters $\theta = \{\theta_1, \ldots, \theta_K\}$ may denote parameters of cluster centers (clustering) or parameters of the subspace (dimensionality reduction)

Probabilistic Unsupervised Learning

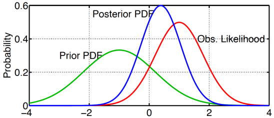
- Consider clustering or dimensionality reduction problems
- Each data point $\boldsymbol{x}_n$ assumed to be generated via some latent variable $\boldsymbol{z}_n$ and parameters θ



  - Clustering: $\boldsymbol{z}_n$ denotes which cluster $\boldsymbol{x}_n$ belongs to
  - Dimensionality Reduction: $\boldsymbol{z}_n$ represents the compressed representation of $\boldsymbol{x}_n$
  - Parameters $\theta = \{\theta_1, \ldots, \theta_K\}$ may denote parameters of cluster centers (clustering) or parameters of the subspace (dimensionality reduction)
- Learning involves estimating the parameters θ and latent variables $\{\boldsymbol{z}_n\}_{n=1}^N$ given data $\{\boldsymbol{x}_n\}_{n=1}^N$

Benefits of Probabilistic Modeling

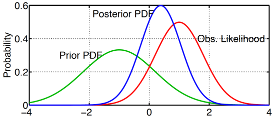
- Can get estimate of the the **uncertainty** in the parameter estimates via the posterior distribution



- Useful when we only have limited data for learning each parameter

Benefits of Probabilistic Modeling

- Can get estimate of the the **uncertainty** in the parameter estimates via the posterior distribution

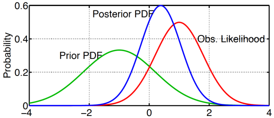


- Useful when we only have limited data for learning each parameter
- Can get estimate of the the **uncertainty in the model's predictions**
  - E.g., Instead of a single prediction $\boldsymbol{y}_*$, we get a distribution over possible predictions (useful for applications such as diagnosis, decision making, etc.)

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

Benefits of Probabilistic Modeling

- Can get estimate of the the **uncertainty** in the parameter estimates via the posterior distribution



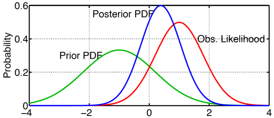
- Useful when we only have limited data for learning each parameter
- Can get estimate of the the **uncertainty in the model's predictions**
  - E.g., Instead of a single prediction $\boldsymbol{y}_*$, we get a distribution over possible predictions (useful for applications such as diagnosis, decision making, etc.)

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

- Can handle missing and noisy data in a principled way

Benefits of Probabilistic Modeling

- Can get estimate of the the **uncertainty** in the parameter estimates via the posterior distribution



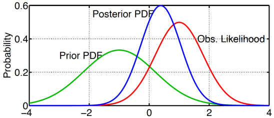
- Useful when we only have limited data for learning each parameter
- Can get estimate of the the **uncertainty in the model's predictions**
  - E.g., Instead of a single prediction $\boldsymbol{y}_*$, we get a distribution over possible predictions (useful for applications such as diagnosis, decision making, etc.)

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

- Can handle missing and noisy data in a principled way
- Easy/more natural to do semi-supervised learning, active learning, etc.

Benefits of Probabilistic Modeling

- Can get estimate of the the **uncertainty** in the parameter estimates via the posterior distribution



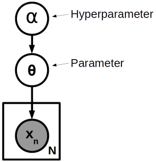
- Useful when we only have limited data for learning each parameter
- Can get estimate of the the **uncertainty in the model's predictions**
  - E.g., Instead of a single prediction $\boldsymbol{y}_*$, we get a distribution over possible predictions (useful for applications such as diagnosis, decision making, etc.)

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

- Can handle missing and noisy data in a principled way
- Easy/more natural to do semi-supervised learning, active learning, etc.
- Can **generate** (synthesize) data by simulating from the data distribution

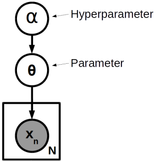
Benefits of Probabilistic Modeling

- Hyperparameters can be learned from data (need not be tuned)

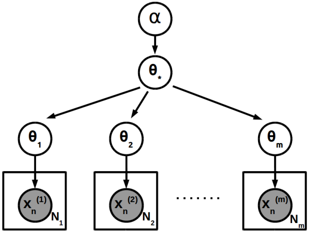


Benefits of Probabilistic Modeling

- Hyperparameters can be learned from data (need not be tuned)

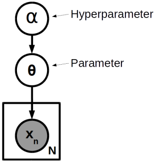


- Simple models can be neatly combined to solve complex problems

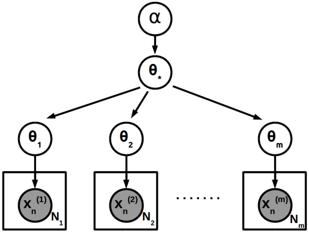


Benefits of Probabilistic Modeling

- Hyperparameters can be learned from data (need not be tuned)



- Simple models can be neatly combined to solve complex problems



- Many other benefits. Highly recommended to read this article from Nature: `http://www.cse.iitk.ac.in/users/piyush/courses/pml_winter16/nature14541.pdf`

Course Outline

- Basics of probabilistic modeling and inference
- Probabilistic models for:
    - Regression and classification
    - Clustering
    - Dimensionality reduction
    - Matrix factorization and matrix completion
    - Time-series data modeling
- Bayesian learning and approximate inference
- Deep Learning
- .. and possibly some other topics of common interest

Course Outline

- Basics of probabilistic modeling and inference
- Probabilistic models for:
  - Regression and classification
  - Clustering
  - Dimensionality reduction
  - Matrix factorization and matrix completion
  - Time-series data modeling
- Bayesian learning and approximate inference
- Deep Learning
- .. and possibly some other topics of common interest

Next class: Maths refresher. Common probability distributions and their properties