UNIVERSITY AT ALBANY
STATE UNIVERSITY OF NEW YORK

CSI 436/536 (Fall 2024)

Machine Learning

Lecture 21: Course Review

Chong Liu

Assistant Professor of Computer Science

Nov 26, 2024

Topics in lectures

Lecture 1	Introduction to Machine Learning [slides]
Lecture 2	Review of Linear Algebra [slides]
Lecture 3	Review of Calculus and Optimization [slides]
Lecture 4	Review of Probability and Statistics [slides]

- Math review (L2-4)
  - Linear algebra, calculus and optimization, probability and statistics
- Supervised learning (L5-17)
  - Evaluation (L5)
  - (Discriminative) Linear model (L6-11)
  - (Generative) Probabilistic model (L12-13)
  - Advanced techniques (L14-17)
- Unsupervised learning (L18-19)
  - Clustering (L18)
  - Dimension reduction (L19)

Lecture 5	Elements of Machine Learning [slides]
Lecture 6	Evaluation Criteria [slides]
Lecture 7	Linear Classifier [slides]
Lecture 8	Loss and gradient descent [slides]
Lecture 9	Linear Regression [slides]
Lecture 10	Regularization [slides]
Lecture 11	Support Vector Machines [slides]
Lecture 12	Max-Likelihood Estimation [slides]
Lecture 13	Naïve Bayes Models [slides]

Lecture 14	Error Decomposition [slides]
Lecture 15	Decision Tree and Boosting [slides]
Lecture 16	Kernel Methods [slides]
Lecture 17	Neural Networks and Deep Learning [slides]
Lecture 18	Clustering [slides]
Lecture 19	Dimension Reduction [slides]

What are you expected to know (L2-4)?

- Basic mathematical tools
  - Linear algebra, calculus and optimization, probability and statistics

Lecture 1	Introduction to Machine Learning [slides]
Lecture 2	Review of Linear Algebra [slides]
Lecture 3	Review of Calculus and Optimization [slides]
Lecture 4	Review of Probability and Statistics [slides]

What are you expected to know (L5-9)?

- Basic concepts of machine learning
  - Classification and regression
  - Input space (feature space), output space (label space), hypothesis class
  - Confusion matrix of binary classification
  - Accuracy
  - Holdout / cross validation / hyperparameter
  - Problem of overfitting
  - Loss function
  - Linear model

Lecture 5	Elements of Machine Learning [slides]
Lecture 6	Evaluation Criteria [slides]
Lecture 7	Linear Classifier [slides]
Lecture 8	Loss and gradient descent [slides]
Lecture 9	Linear Regression [slides]
Lecture 10	Regularization [slides]
Lecture 11	Support Vector Machines [slides]
Lecture 12	Max-Likelihood Estimation [slides]
Lecture 13	Naïve Bayes Models [slides]

What are you expected to know (L7-13)?

- Understanding how machine learning algorithms work
  - Why do we need surrogate loss in classification?
  - Why do we need SGD? Drawback of GD?
  - How to define a linear classifier / linear regression?
  - Why do we need SVM? Difference between linear classifier and SVM.
  - Why do we need regularization? How to apply it?
  - Key idea of maximum likelihood estimation.
  - Key assumption of Naïve Bayes models.

Lecture 5	Elements of Machine Learning [slides]
Lecture 6	Evaluation Criteria [slides]
Lecture 7	Linear Classifier [slides]
Lecture 8	Loss and gradient descent [slides]
Lecture 9	Linear Regression [slides]
Lecture 10	Regularization [slides]
Lecture 11	Support Vector Machines [slides]
Lecture 12	Max-Likelihood Estimation [slides]
Lecture 13	Naïve Bayes Models [slides]

What are you expected to know (L14-19)?

- Main focus of final exam

- Understanding how machine learning algorithms work?
  - Advanced techniques to improve linear models
    - Error decomposition to understand ML
    - Ensemble methods
    - Kernel methods / feature transformation
    - Neural networks
  - Unsupervised learning
    - Clustering
    - Dimension reduction

Lecture 14	Error Decomposition [slides]
Lecture 15	Decision Tree and Boosting [slides]
Lecture 16	Kernel Methods [slides]
Lecture 17	Neural Networks and Deep Learning [slides]
Lecture 18	Clustering [slides]
Lecture 19	Dimension Reduction [slides]

Examples of supervised learning problems

	Binary classification	Multi-class classification	Regression
Feature space	$\mathbb{R}^d$	$\mathbb{R}^d$	$\mathbb{R}^d$
Label space	{-1, 1}	{1,2,3,…,K}	$\mathbb{R}$
Popular performance metric	Classification error (0-1 loss) for test data	Classification error (0-1 loss) for test data	Mean Square Error (MSE) vs ground truth
Popular surrogate loss (for training)	Logistic loss Square loss Hinge loss	Multiclass logistic loss aka. Cross-Entropy loss	Square loss

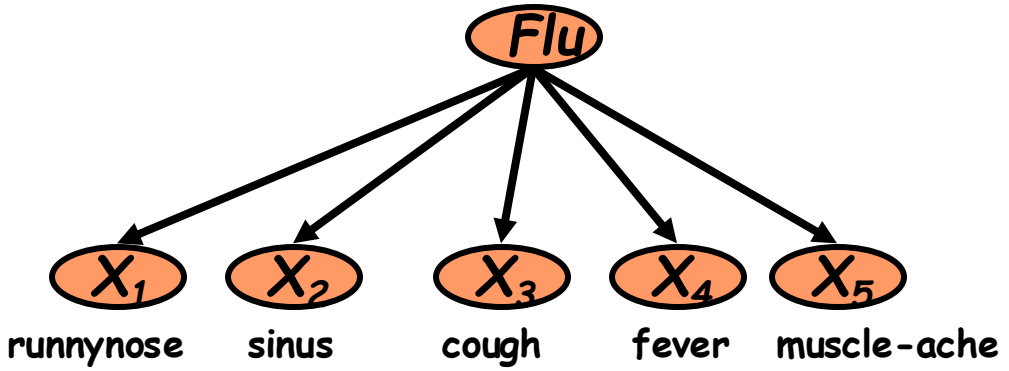
Regularization and SVM

- Regularization is used to avoid overfitting
- SVM is developed based on max-margin idea
- Statistically, regularization == max-margin

Maximum Likelihood Estimation

- MLE defines an optimization problem to solve for estimating the parameters given data.
  - Find the parameter that maximizes the likelihood
  - Find the **distribution** within **a set of distributions** that maximizes the probability (likelihood) of observing the data

Naïve Bayes Model --- a simple example of a generative model



- **Conditional Independence Assumption:** features are independent of each other given the class (label):

$$P(X_1,\ldots,X_5 \mid C) = P(X_1 \mid C) \bullet P(X_2 \mid C) \bullet \cdots \bullet P(X_5 \mid C)$$

- **Interpretation:** given that you have Flu, the event that you experience each of the five symptoms are independent.
- **The task of classification:** Predict disease using symptoms

Problem setup for machine learning problems

- Loss function

$$\ell(h, (x, y))$$

- Empirical Risk function

$$\hat{R}(h, \text{Data}) = \frac{1}{n} \sum_{i=1}^{n} \ell(h, (x_i, y_i))$$

- (Population) Risk function

$$R(h, \mathcal{D}) = \mathbb{E}_{\mathcal{D}}[\ell(h, (x_i, y_i))]$$

Risk Decomposition

$$\mathbb{E}[R(\hat{h})] - R(h_{\text{Bayes}})$$

$$\leq \underbrace{\mathbb{E}[\hat{R}(\hat{h}) - \hat{R}(h_{\text{ERM}})]}_{\text{Optimization Error}} + \underbrace{R(h^*) - R(h_{\text{Bayes}})}_{\text{Approximation Error}} + \underbrace{\mathbb{E}[R(\hat{h}) - \hat{R}(\hat{h})]}_{\text{Generalization Error}}$$

Optimization Error

How close am I from minimizing the empirical risk?

Approximation Error

How much worse the best "representable" classifier is from the best classifier out there.

Generalization Error

How different the empirical risk of my classifier is from its population risk?

Machine learning can be viewed as a collection of techniques in minimizing the three types of errors

	Optimization error	Generalization Error	Approximation Error
Definition	$\hat{R}(\hat{h}) - \hat{R}(h_{\mathrm{ERM}})$	$R(\hat{h}) - \hat{R}(\hat{h})$	$R(h^*) - R(h_{\mathrm{Bayes}})$
Challenges	• Finding ERM for some loss functions is NP-Hard. • Efficiency isn't enough. Need to be scalable.	• We do not observe Risk! • Don't have infinite data. • Large generalization error ⇔ Overfitting	• Don't know data distribution. • No knowledge of Bayes optimal classifier. • Large approx. error ⇔ Underfitting!
What we have learned to address these challenges?	"Just-relax" Surrogate loss, Gradient Descent, SGD Other more specialized solutions to optimization problems	Holdout, Cross-Validation Regularization Statistical learning theory *(advanced topic)*	Better features More flexible decision boundaries Better probabilistic models **Ensemble learning: Boosting, Bagging** **Feature expansion/ Kernels** **Neural Networks / Representation Learning**

Key questions in modelling and tradeoffs

- How to come up with suitable hypothesis class?
  - We want the approximation error to be small
  - We want it to be (statistically) efficiently learnable

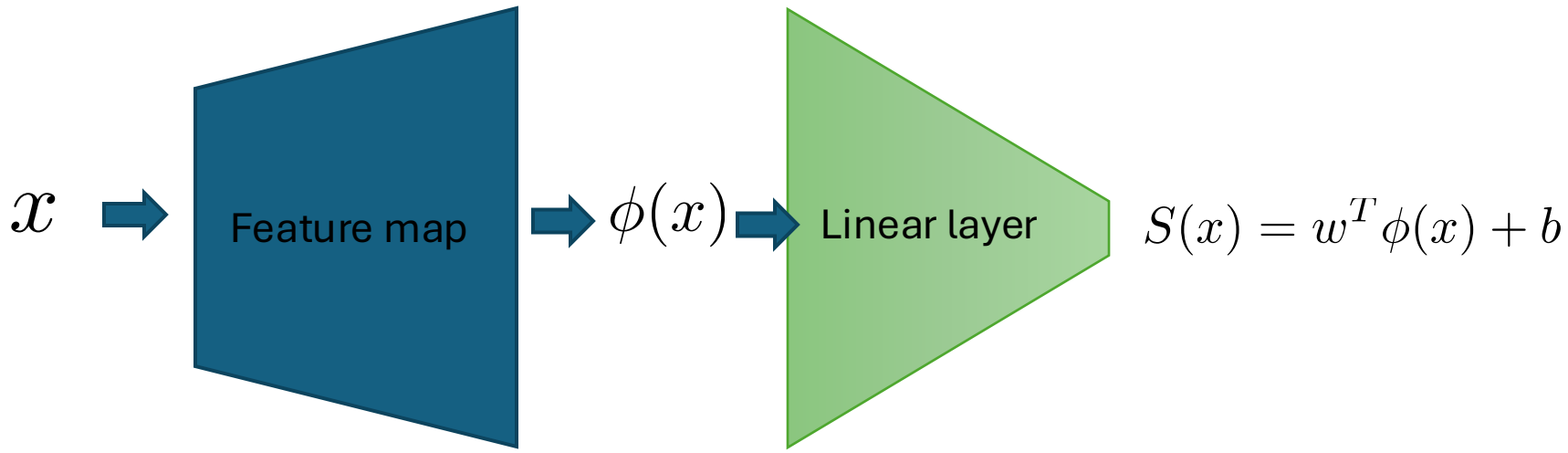
- How to come up with suitable loss functions?
  - We want the loss function to reflect that performance metric of interest.
  - We want the loss function to be efficiently optimizable

Two philosophy for answering the two questions

- Deterministic / Discriminative view:
  - Loss function is a surrogate the performance metric that we care about.
    - e.g. logistic loss, hinge-loss, etc. upper bounds the 0-1 loss
  - Geometric view: Hypothesis class specifies the shapes of the decision boundary.

- Probabilistic / Generative view:
  - Loss function can be derived from Max-Likelihood Principle.
  - Hypothesis class is specified by a probabilistic model of the data-generation process.

From kernels to neural networks



Neural networks: Example: AlexNet (2012)

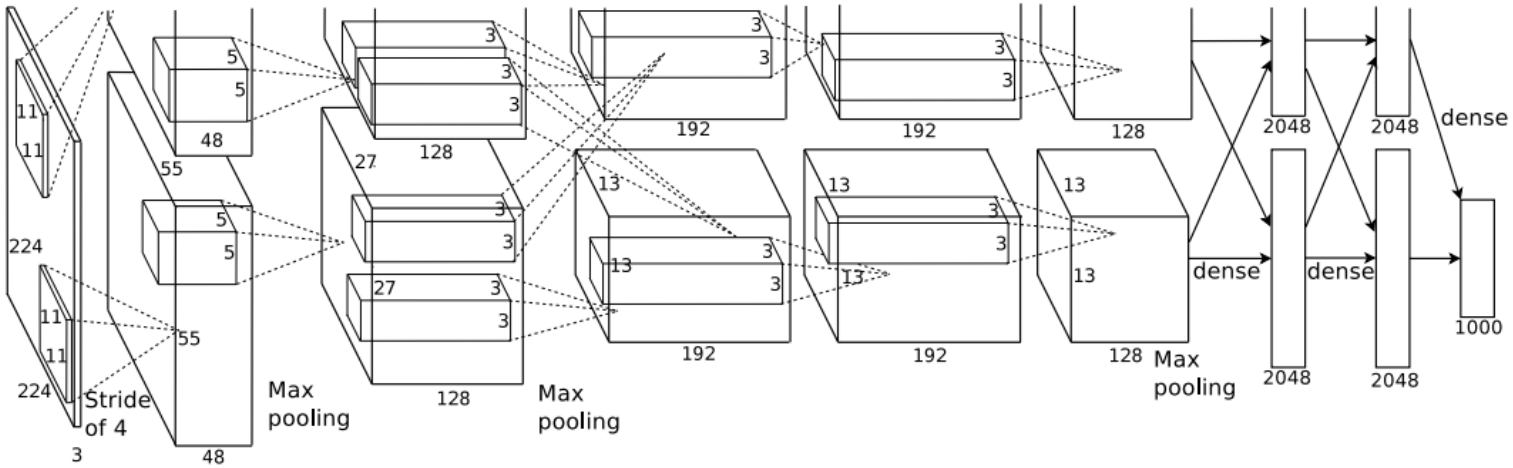


Figure 2: An illustration of the architecture of our CNN, explicitly showing the delineation of responsibilities between the two GPUs. One GPU runs the layer-parts at the top of the figure while the other runs the layer-parts at the bottom. The GPUs communicate only at certain layers. The network's input is 150,528-dimensional, and the number of neurons in the network's remaining layers is given by 253,440–186,624–64,896–64,896–43,264–4096–4096–1000.

Imagenet classification with **deep convolutional neural networks**

A Krizhevsky, I Sutskever… - Advances in **neural** …, 2012 - proceedings.neurips.cc

… a large, **deep convolutional neural network** to **classify** the 1.2 million high-resolution images in the **ImageNet** … The **neural network**, which has 60 million parameters and 650,000 neurons, …

☆ Save 99 Cite Cited by 122248 Related articles All 111 versions Import into BibTeX »

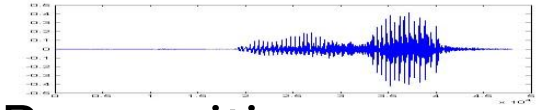
You can use neural network for all kinds of ML problems that we learned: classification, regression, clustering, dimension reduction etc..

- Neural networks provide a learnable function approximation

- Different kinds of NNs architecture (like LEGO blocks) are designed to address different challenges in different kind of problems:
  - Feedforward neural network
  - Recurrent neural network
  - Boltzmann machine
  - Convolutional neural network
  - Graph Neural Networks
  - Transformers
  - etc., etc.

Learning ≈ Configuring the Learnable Function so it behaves as instructed.

- Speech Recognition

$f($  $) =$ "Hello"

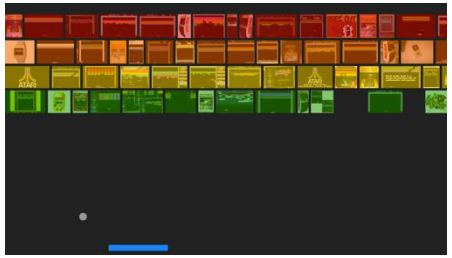
- Handwritten Recognition

$f($  $) =$ "2"

- Weather forecast

$f($ Thursday $) =$ "  Saturday"

- Play video games

$f($  $) =$ "move left"

Generally speaking, you need to make decisions about

- Which loss function to use
  - Regression, classification, clustering, dimension reduction, but also ranking, recommendation, and others…
- What type of neural network to use
  - Images
  - Text
  - Graphs (node and edges)
  - Time series
  - Decide on the hyperparameters: Depth, Width, number of hidden units, etc…
- How to train the neural network?
  - Initialization of weights: iid random? Recale or not?
  - Optimizer to use: SGD, ADAM, etc…
- How to collect, pre-process the data…

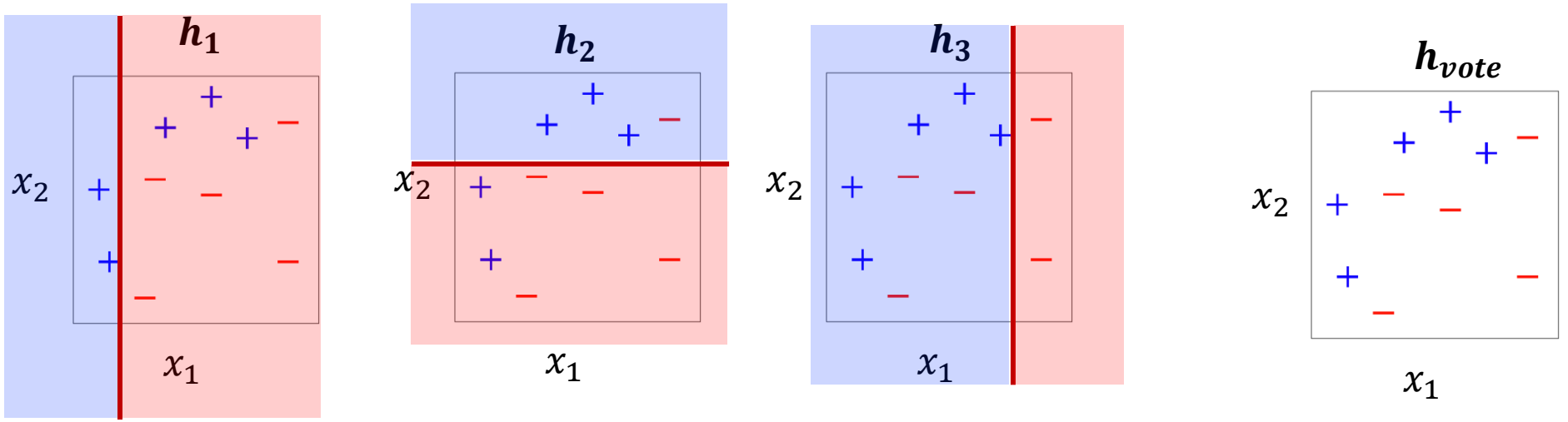
Classifiers that we have learned so far

- Linear classifier
- Decision Tree (Decision Stumps)
- Naïve Bayes classifier
- Voting classifiers <= Bagging, Boosting
- Feature-expanded linear classifiers <= Kernel methods
- Neural Networks <= Learning representation

Important learning goals:

1. What are the parameters
2. Fix a parameter, how does the classifier make predictions
3. Sketching the decision boundary

Decision stumps and voting classifiers



$$\alpha_1 = \alpha_2 = \alpha_3 = 1.0$$

$$h_{\text{vote}}(x) = \arg\max_{y \in \mathcal{Y}} \frac{1}{N} \sum_{i=1}^{N} \alpha_i \mathbf{1}(h_i(x) = y)$$

Two unsupervised learning problems

- K-means clustering
  - Assign hard labels to each data point
  - What does the learning process look like? Alternating between finding the cluster and calculating the clustering centers.

- Principal Component Analysis for dimension reduction
  - Linear dimension reduction
  - What's the key idea? Max variance or min construction error.
  - Algorithm is based on matrix eigen-decomposition.

Problems, Learning Algorithm, Inference Algorithm

Problems	Problems	Learning Algorithm	Inference
Unsupervised Learning	K-means clustering	K-means algorithm / SGD	Assigning data points to clusters: c ← x
	Principal component analysis	PCA algorithm (eigen-decomposition)	Reduce dimension
Supervised Learning	Linear classification	Perceptron / SGD with logistic loss	Prediction $y \leftarrow \text{sign}(x^T w + b)$
	Naïve Bayes Classification	Solve MLE (e.g., direct solver or SGD)	Prediction $p(y\|x)$
	Learning Voting Classifiers	Bagging, Boosting, AdaBoost	Prediction using voting classifiers
	Learning Neural Networks	SGD to learn ϕ, and w, b	Prediction by $y \leftarrow \text{sign}(\phi(x)^T w + b)$

Final exam

- What does the exam look like?
  - 90 min (8:30-10am) on Monday December 16 at LC 5
  - Please arrive earlier!
  - Closed-book exam
  - Given individually (not in groups!)
  - Counts 30% towards your final grades
  - Main focus is lectures after mid-term
- What to bring?
  - Your pen
- What not to bring?
  - Books, notes, lecture slide, draft papers, or cheat sheets

Announcements

- Only 2 instructor office hours left!
  - Today after class
  - Next Tue Dec 3
- UAlbany Course Evaluation form
  - 2 participation points for everyone if we get above 60%!
- Final presentation (Tue Dec 3)
  - 15 groups, each has 3 min
  - 13 points towards your final grades
- Final exam (Mon Dec 16)
  - 30 points towards final grades
  - Review homework 1-4, midterm exam, and lecture slides
  - Main focus is lectures after midterm