

# COGS 185 — Advanced Machine Learning Methods

Notes taken by Nolan Chai

Spring 2023

# Contents

<b>0</b>	<b>Introduction</b>	<b>4</b>
<b>1</b>	<b>A Review of Supervised Learning</b>	<b>5</b>
1.1	Structure . . . . .	5
1.2	Trends in AI . . . . .	5
1.3	The three components of learning algorithms . . . . .	5
1.4	Structural Risk Minimization . . . . .	6
1.5	Context . . . . .	6
<b>2</b>	<b>Multi-Class Classification</b>	<b>8</b>
<b>3</b>	<b>Support Vector Machines</b>	<b>9</b>
<b>4</b>	<b>Softmax function</b>	<b>10</b>
<b>5</b>	<b>Structured Prediction</b>	<b>11</b>
<b>6</b>	<b>Random Fields</b>	<b>12</b>
<b>7</b>	<b>Auto-Context</b>	<b>13</b>
<b>8</b>	<b>Auto-Context (Cont.)</b>	<b>14</b>
<b>9</b>	<b>Recurrent Neural Networks</b>	<b>15</b>
<b>10</b>	<b>Recurrent Neural Networks (Cont.)</b>	<b>16</b>
<b>11</b>	<b>Attention based models</b>	<b>17</b>
<b>12</b>	<b>Transformers</b>	<b>18</b>
<b>13</b>	<b>Large Language Models</b>	<b>19</b>
<b>14</b>	<b>Compressive Sensing</b>	<b>20</b>
<b>15</b>	<b>Weakly-Supervised Learning</b>	<b>21</b>
<b>16</b>	<b>Self-Supervised Learning</b>	<b>22</b>
<b>17</b>	<b>Vision Transformers</b>	<b>23</b>
<b>18</b>	<b>Generative Adversarial Networks</b>	<b>24</b>

## Preface

These are a collection of notes personally taken by me, specifically for readings and allotted content for UCSD's COGS 185 Advanced Machine Learning Methods taken in Spring 2023. These notes are not endorsed by the lecturers nor staff, and I have modified them (often significantly) over random periods of time. They may become nowhere near accurate representations of what was actually lectured, or written in the books, and are simply to aid in my own understanding. In particular, all errors are almost surely mine.

*Notes are taken real time, and will be reviewed, updated, and revised within 48 hours of each lecture.*

My other notes are available **here**.

## 0 Introduction

*The course site is available [\*\*here\*\*](#).*

The main thing we'll be going over in this course is **context** - the most important thing in modern machine learning. In geeneral, when we say context - we mean **two things**: context within an instance such as yourself as a person (reasons why you decide on things, such as why you chose UCSD, or a particular subject), and context across instances (across a population - similarities with other people).

We'll proceed incrementally, beginning with a review of supervised learning into multiclass/multi-label classification, structural predictions, sequence modeling, semi-supervised and unsupervised learning, self-supervised learning, sparse coding, and reinforcement learning.

## Logistics

The course is structured as a hybrid course, with lectures available both online and in-person. The grade breakdown will be as follows:

- Assignments (4 total): 50%
- Midterm: 25%
- Final Project: 25%
- Bonus Points: 3% (Piazza, Final Project)

The midterm will be **Thursday of week 4**. Attendance is also not mandatory.

# 1 A Review of Supervised Learning

To begin, what exactly is a *pattern*? Is it repetitive? Not really. You can refer to a plank of wood, or a wooden tabletop, that contains a 'pattern' but it may not necessarily be repetitive. However, they can contain *subjective, explicit, and implicit descriptions*. They do not necessarily have to contain common features.

Then what is not a pattern? No "pattern" is also a pattern. Unpredictability can be a pattern. In essence, everything is a pattern.

## 1.1 Structure

But also, what is structure, then? When we look at a small segment of images, it's very difficult to understand the underlying structure without *context*. Supervised learning is basically a massive encoder for context.

The main scheme for structures is as follows:

- Structure within a data sample (*supervised*)
- Structure between data samples (*semi-supervised*)
- Structure within a sample (*unsupervised*)

## 1.2 Trends in AI

We've started from logical and hard-coded artificial intelligence (traditional AI) to statistical analyses to machine learning, and now, finally artificial intelligence (modern AI/ML). There is emphasis on two things: simplicity of capability / structure, and scalability.

AutoML was a hot topic for a while until the release of ChatGPT, where the importance of searching for best parameters was trivialized. Success in artificial intelligence and machine learning revolves around the availability of large amounts of training data (e.g., ImageNet), the access of modern computing infrastructures (e.g., Nvidia GPUs), and new developments in neural networks with deep structures (e.g., AlexNet).

AlexNet was one of the most fundamental models, as it displayed the importance of neural network architectures; and now, ChatGPT is the second major development in terms of waking up the world to how important such landmarks are.

## 1.3 The three components of learning algorithms

We classify the main three components of learning algorithms under representation, evaluation, and optimization. These are all typically very important, but in today's world, having good data is **far** more important than optimizing well.

Overall, Learning = Representation + Evaluation + Optimization.

## 1.4 Structural Risk Minimization

















Let  $\phi(f)$  = the set of functions representable by  $f$ .

Suppose:  $\phi(f_1) \subset \phi(f_2) \subset \dots \phi(f_n)$

Then:  $h(f_1) \leq h(f_2) \leq \dots h(f_n)$

We are trying to decide which machine to use. We train each machine and make a table based on our standard optimization for generalization:

$$e_{testing} \leq e_{training} + \sqrt{\frac{h(\log(2n/h + 1) - \log(n/4))}{n}}$$

$i$	$f_i$	$e_{training}$	$\sqrt{\frac{h(\log(2n/h+1)-\log(n/4))}{n}}$ generalization	upper bound $e_{testing}$	choice
1	$f_1$				
2	$f_2$				
3	$f_3$				
4	$f_4$				
5	$f_5$				

The more complex the model is, the smaller the training error. However, the less generalization capability it will subsequently have. Or...at least this is the case with older models.

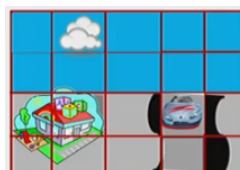
Suppose we increase the size and model complexity of ChatGPT's parameters even further than it is currently; let's say 100x. Although this would typically result in greater training error, it somehow doesn't decrease in generalization - rather, it gets better. This is the current state of LLMs, and why there is so much hype around it - we want to push the boundaries to see how far this will go.

When you begin to solve a machine learning problem, you want to focus on first understanding it by formulating it between an input and output state.

## 1.5 Context

Context comes from both *within-data* (parts/components) and *between-data* (configurations). For instance, some problems may exist where, from a small section of an image, you need context to understand the full image.

To formalize the problem, we can look at the image as a whole, then divide it into patches like so:

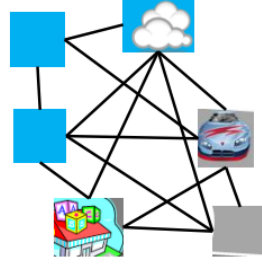


While subsequently labeling the images, which is a problem that we deal with - structural information:

$$\vec{x} = (\text{[blue square]}, \text{[cloud]}, \text{[blue square]}, \text{[house]}, \text{[car]}, \text{[road]})$$

$$\vec{y} = (\text{sky}, \text{cloud}, \text{sky}, \text{building}, \text{car}, \text{road})$$

Which we can then use to generate graph:



In today's world, we use dense graphs utilized by transformers. Before, we had primarily focused on sparse graphs due to the fear that we could not deal with dense graphs. Transformers are able to fully utilize dense graphs due to its adaptive attention mechanism and connect everything - this is also related to why transformers have such an incredible number of parameters.

This problem formulation can be represented by the following random fields:

- Markov Random Fields:

$$p(\vec{y} \mid \vec{x}) \propto p(\vec{y})p(\vec{x} \mid \vec{y})$$

$$\rightarrow \prod_{(i,j) \in \mathcal{N}} p(y_i, y_j) \prod p(x_i \mid y_i)$$

- Conditional Random Fields:

$$p(\vec{y} \mid \vec{x})$$

$$\rightarrow \prod_{(i,j) \in \mathcal{N}} p(y_i, y_j \mid x_i, x_j) \prod p(y_i, x_i)$$

## **2 Multi-Class Classification**



## 3 Support Vector Machines

## 4 Softmax function

## 5 Structured Prediction

## 6 Random Fields

## 7 Auto-Context

## 8 Auto-Context (Cont.)

## 9 Recurrent Neural Networks

## 10 Recurrent Neural Networks (Cont.)



## 11 Attention based models

## 12 Transformers

## 13 Large Language Models

## 14 Compressive Sensing

## 15 Weakly-Supervised Learning

## 16 Self-Supervised Learning

## 17 Vision Transformers

## 18 Generative Adversarial Networks