Summer School 2014
Learning at Scale
Machine Learning is...

...the branch of engineering that develops technology for automated inference

--- Cosma Shalizi

It combines...
Machine Learning is...

...the branch of engineering that develops technology for automated inference

--- Cosma Shalizi

It combines...

ALGORITHMS
Machine Learning is...

...the branch of engineering that develops technology for automated inference

--- Cosma Shalizi

It combines...

![Diagram showing overlap of algorithms and statistics](image-url)
Machine Learning is...

*...the branch of engineering that develops technology for automated inference*

--- Cosma Shalizi

It combines...

ALGORITHMS  [Statistics]  Optimization
Machine Learning is...

...the branch of engineering that develops technology for automated inference

--- Cosma Shalizi

It combines...

ALGORITHMS

Statistics

Optimization

Geometry
Flavors of Learning
Supervised Learning

(Binary) classification

Given: \[ \{(x_1, y_1), (x_2, y_2), \ldots \} \] drawn from some source

\[ x_i \in \mathbb{R}^d \quad y_i \in \{+1, -1\} \]

Find a function \( f : \mathbb{R}^d \mapsto \{+1, -1\} \) such that

\[ f \text{ captures the relationship between } x \text{ and } y \]

\[ \forall i, f(x_i) = y_i \]
Supervised Learning

(Binary) classification

Given: \( \{ (x_1, y_1), (x_2, y_2), \ldots \} \)

Find a function \( f : \mathbb{R}^d \rightarrow \{+1, -1\} \) such that \( f \) captures the relationship between \( x_i \) and \( y_i \).
(Binary) classification

Given: \[ \{(x_1, y_1), (x_2, y_2), \ldots\} \]

Find a function \( f : \mathbb{R}^d \rightarrow \{\pm 1\} \) such that \( f \) captures the relationship between \( x \) and \( y \).

- Spam: Testing if email is spam or not
- Sentiment analysis: is a product review positive or negative
Supervised Learning

Regression

Given: \( \{(x_1, y_1), (x_2, y_2), \ldots \} \) drawn from some source

\[
x_i \in \mathbb{R}^d \quad y_i \in \mathbb{R}
\]

Find a function \( f : \mathbb{R}^d \mapsto \mathbb{R} \)

such that \( f \) captures the relationship between \( x \) and \( y \)

\[
\forall i, f(x_i) = y_i
\]
Supervised Learning

Regression

Given: \( \{(x_1, y_1), (x_2, y_2), \ldots\} \)

Find a function \( f(x_i) = y_i \) drawn from some source such that \( f \) captures the relationship between \( x \) and \( y \).
Supervised Learning

Regression

Given: \( \{(x_1, y_1), (x_2, y_2), \ldots\} \)

Find a function \( f(x_i) = y_i \) drawn from some source such that \( f \) captures the relationship between \( x \) and \( y \).

- Predictions: Stock market price as function of financial specs
- Relationship between dosage and effectiveness
Unsupervised Learning

Clustering

Given a collection of objects, find a way to "group" them into similar pieces

Learning a function \( f : \mathbb{R}^d \mapsto \{1, 2, \ldots, k\} \)

But we don't have any examples of the "correct" answer!
Clustering is closely related to classification with multiple classes
Unsupervised Learning

Clustering

Given a collection of objects, find a way to "group" them into similar pieces.

Learning a function \( f : \mathbb{R}^d \mapsto \{1, 2, \ldots, k\} \)

But we don't have any examples of the "correct" answer!

Clustering is closely related to classification with multiple classes.
Unsupervised Learning

Dimensionality Reduction (or Feature Learning)

Given objects in $\mathbb{R}^d$ find a mapping $A : \mathbb{R}^d \mapsto \mathbb{R}^k$, $k \ll d$

that preserves the "structure" of the objects

- Find "relevant" dimensions for a task
- Reduce dimensionality to manage complexity of algorithms
Mixing and Matching

Semi-supervised learning: labelled and unlabeled data

Find a classifier that separates the labeled points and separates the unlabeled points "well"

Often have lots of unlabeled data and only a little labeled data to guide efforts
Mixing and Matching

Supervised clustering = multiclass classification

Supervised dimensionality reduction = (linear) discriminant analysis
Understanding vs Predicting

\{ (x_1, y_1), (x_2, y_2), \ldots \} \text{ is drawn from distribution } p(X, Y)

Generative learning: Learn the distribution \( p(X, Y) \)

"What controls the rise and fall of the tides?"

Discriminative learning: Learn the \textbf{conditional} distribution \( p(Y | X) \)

"Will there be a high tide tomorrow evening?"

\[ p(Y | X) = \frac{p(X, Y)}{p(X)} \]
Understanding vs Predicting

Discriminative clustering: predict the cluster of a new point

\[ p(\bullet|\bigcirc) = \exp(-\|\bigcirc - \bullet\|^2) \]
Understanding vs Predicting

Discriminative clustering: predict the cluster of a new point

\[ p(\bullet|\circ) = \exp(-\|\circ - \bullet\|^2) \]

Generative clustering: mixture density estimation
Parametric Learning:
- Define a space of models parametrized by fixed number of parameters
- Find model that best fits the data (by searching over parameters)

Parametric binary classification:
- Model: \((\mu_1, \Sigma_1, \mu_2, \Sigma_2)\)
  \[ p(x) \propto \exp\left(- (x - \mu)^\top \Sigma (x - \mu)\right) \]
- Maximize likelihood of any model
- \(d^2 + 2d\) parameters
Non-parametric Learning:
- Define a space of models that can grow in size with data.
- Find model that best fits the data
- "Non-parametric" means "Not-fixed", not "none"!

4 "support points" define the resulting classifier.
Bayesian Learning

Non-Bayesian (parametric) learning:

\[
\{\Theta\}, \quad \{(x_1, y_1), (x_2, y_2), \ldots\}
\]

Learner

\[\Theta^*\]
Bayesian Learning

Non-Bayesian (parametric) learning:

\[ \{ \Theta \} \]
\[ \{(x_1, y_1), (x_2, y_2), \ldots \} \]

Bayesian learning:

\[ p(\Theta) \]

Prior (belief about the world)

\[ \{(x_1, y_1), (x_2, y_2), \ldots \} \]

\[ \hat{\Theta} \]

Posterior (belief about the world)

\[ \hat{\Theta} \]

\( \Theta^* \) is a point estimate.
\( \hat{\Theta} \) is a distribution over possible worlds
Bayesian Learning

You know you’re talking to a Bayesian if...

What’s your prior?

Conjugate priors

Maximum a posteriori (MAP)
Many Learning Frameworks

- Online Learning: must make prediction as soon as item arrives
- Active Learning: I can get labels for data, but it's expensive.
- Multi-task Learning: I'm learning different tasks, but they're related so maybe the tasks can learn from each other.
- Transfer Learning: I can learn well in one domain: can I transfer this knowledge into a different domain?
- Ensemble Learning: I have bad learners, but together they're decent
The Mechanics of Learning
Loss Functions

Given: \( \{(x_1, y_1), (x_2, y_2), \ldots \} \) drawn from some source
\[
x_i \in \mathbb{R}^d \quad y_i \in \{+1, -1\}
\]

Find a function \( f : \mathbb{R}^d \mapsto \{+1, -1\} \)

such that \( f \) captures the relationship between \( x \) and \( y \)

Loss functions \( L(f((x), y)) \) measure the quality of \( f \):

0-1 loss:
\[
1_{f(x) \neq y}
\]

Hinge loss:
\[
\max(0, 1 - y \cdot f(x))
\]

Square loss:
\[
(y - f(x))^2
\]
Estimating Risk

Once we have a loss function, we can quantify how good a predictor is:

\[ R(f) = E_{x,y}[L(f(x), y)] = \int p(x, y)L(f(x), y) \]

and find a good predictor:

\[ f^* = \arg \min_{f \in \mathcal{F}} R(f) \]

But we don't usually know what the data distribution is, so we can't solve the minimization!
Empirical Risk Minimization

Assume the given data is drawn from the source distribution. Replace

\[ R(f) = \mathbb{E}_{x,y}[L(f(x), y)] = \int p(x,y)L(f(x), y) \]

by the empirical mean:

\[ \hat{R}(f) = \frac{1}{n} \sum_{i} L(f(x_i, y_i)) \]

with the hope that the estimate is unbiased and converges:

\[ E[\hat{R}(f)] = R(f), \quad \hat{R}(f) \to R(f) \]

But now we have a "normal" optimization:

\[ \min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i} L(f(x_i, y_i)) \]
Overfitting and Regularization

The problem with optimizing over the data is that you can over-fit to your samples. (low bias)

This is bad because then your predictive power goes down (high variance) and you can't generalize.

Complex models (with more parameters) can overfit. Penalize them!

$$\min_{f \in F} \frac{1}{n} \sum_{i} L(f(x_i, y_i)) + c(f)$$

This is called regularization.

model complexity term
Generalization

How many samples of the data do you need for the empirically optimized answer to get close to the true answer?

If your function space is "well behaved" (not too wiggly), then you don't need too many samples.

Well behaved:
• VC dimension is small
• Rademacher complexity is small
• Fat shattering dimension is small
• ... and others.

All of this assumes that you sample from the real distribution...
Overview... so far...

1) Choose a learning task (classification, clustering, regression, ...)
2) Pick a convenient loss function
3) Sample a sufficient number of points from a source
4) Build an optimization using the data, the loss function, and any regularizers
5) OPTIMIZE !!!
6) Use learned model on new data to predict.
7) (if you're doing online learning, repeat)

$ML = \text{Design choices} + \text{careful optimization}$
Representations

The Computational Geometry prayer:
Let $P$ be a set of points in the plane. Amen.
Representations

The Computational Geometry prayer:
Let $P$ be a set of points in the plane. Amen.

Let $G$ be a graph with $n$ vertices and $m$ edges
Representations

The Computational Geometry prayer:
*Let P be a set of points in the plane. Amen.*

Let G be a graph with n vertices and m edges

Let S be a set of elements drawn from a universe U
The Computational Geometry prayer:

Let $P$ be a set of points in the plane. Amen.

Let $G$ be a graph with $n$ vertices and $m$ edges

Let $S$ be a set of elements drawn from a universe $U$

Let $M$ be an $m \times n$ matrix of reals.
Representations

The Computational Geometry prayer:
*Let P be a set of points in the plane. Amen.*

Let G be a graph with n vertices and m edges

Let S be a set of elements drawn from a universe U

Let M be an m X n matrix of reals.

But in learning, we don't have a "natural" representation. We have to CHOOSE one.
Representations help algorithms

Learning a circle separating classes can be tricky
Representations help algorithms

Learning a circle separating classes can be tricky

If we change the representation

\[ \ell : (x, y) \mapsto (x, y, x^2 + y^2) \]
Representations help algorithms

Learning a circle separating classes can be tricky

If we change the representation

\[ \ell : (x, y) \mapsto (x, y, x^2 + y^2) \]

Circle separation becomes linear separation!
Constructing a representation

Supervision guides the representation

Unsupervised spectral representation

And many other kinds...
A new overview of learning

- Choose a learning task
- Build a model of data
- Shape your learner (loss, regularizer)
- Choose samples from a source
- Optimize!
- Predict!
A new overview of learning

Choose a learning task

Shape your learner (loss, regularizer)

Choose samples from a source

Optimize!

Predict!
A new overview of learning

Choose a learning task
Shape your learner (loss, regularizer)
Choose samples from a source
Optimize!
(Submodular) optimization
Provably efficient algorithms
Predict!
A new overview of learning

- Kernels
- Choose a learning task
- Shape your learner (loss, regularizer)
- Choose samples from a source

- Optimize!
- (Submodular) optimization
- Provably efficient algorithms
- Doing all of this at scale

Doing all of this at scale