Statistical Machine Learning and Kernel Methods
MADALGO Summer School, Aug 2014

Mikhail Belkin, Ohio State University,
Department of Computer Science and Engineering,
Department of Statistics

What is machine learning?

- Learning means finding a "pattern" in your experience.

Babies are not born with understanding of everyday objects, speech, writing, mathematics, …

- Machine learning is teaching machines to find patterns from "experience".

- Understand the nature of learning. Need:

1. mathematical formalism to describe
    a. the space of patterns
    b. the learning process – fitting pattern to experience

2. algorithms to implement learning in a computer

What is machine learning?

Interdisciplinary subject: CS, Statistics, Applied Mathematics, Engineering.

Also called pattern recognition.

Modern ML:

inference from large, high-dimensional, complex data.

Why statistical learning?

A way of dealing with uncertainty.

Powerful mathematical formalism.

Learning from examples

$$X = \begin{cases} \text{Pattern Space} \\ \text{Instance Space} \\ \text{Example Space} \end{cases} \quad \mathbb{R}^n, \mathcal{M}, \{-1,+1\}^n, \Sigma^*$$

$$Y = \begin{cases} \text{Label Space} \\ \text{Prediction Space} \\ \text{Response Space} \end{cases} \quad \mathbb{R}^n, \{-1,+1\}, \{1,\ldots,n\}$$

Examples (x, y)

Learning machine:
given a set of examples $(\mathrm{x_i}, \mathrm{y_i})$ construct (learn) a function $\phi: X \to Y$

We will talk primarily about supervised classification $Y = \{-1,1\}$ and regression $Y = \mathbb{R}$

Classification

Data (x_i, y_i)

x's are features, e.g. pixel values

y's are labels (+1, -1), e.g. faces or non-faces

Problem:

Construct an "optimal/reasonable" classifier function from X to $\{+1, -1\}$.

Need a definition of optimality. Start with a model. Note: cannot make predictions without having a model for the future.

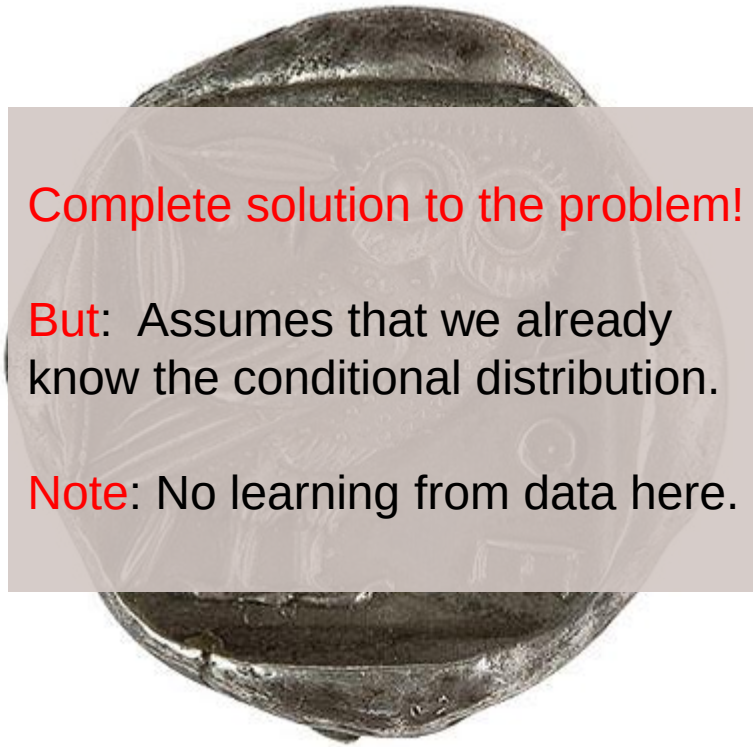
Standard statistical model for the data: a probability distribution p on $X \times Y$.

For classification equivalent to specifying conditional probability function $P(+1|x)$.

How to evaluate classifiers

The economics approach to classification:

Lose $1 for every wrong prediction.



Complete solution to the problem!

But: Assumes that we already know the conditional distribution.

Note: No learning from data here.

In the statistical framework classifier $c(x)$ will be expected to lose

$$L(c) = E_{X \times Y} 1_{c(x) \neq y}.$$

Best classifier? Bayes optimal classifier:

$$c_{opt}(x) = \begin{cases} 1, & \text{if } P(1|x) > 1/2 \\ -1, & \text{if } P(1|x) \leq 1/2 \end{cases}$$

Bayes optimal classifier minimizes the expected loss!

Empirical risk minimization (ERM)

Bayes optimal minimizes expected loss $L(c) = E_{X \times Y}\, 1_{c(x) \neq y}$.

Cannot be evaluated directly.

Need something that uses data $(x_i, y_i), i = 1 \ldots n$.

Idea: Find a function $c(x)$ with small empirical loss

$$L_{emp}(c) = \frac{1}{n} \sum 1_{c(x_i) \neq y_i}.$$

Two issues:

- Structural: Ill-posed problem. What do we actually want?
- Algorithmic: how do we find it efficiently?

ERM: learning from data

Ill-posed problem: infinitely many functions c(x), s.t.

$$L_{emp}(c) := \frac{1}{n}\sum 1_{c(x_i)\neq y_i} = 0.$$

Is zero empirical loss even desirable?

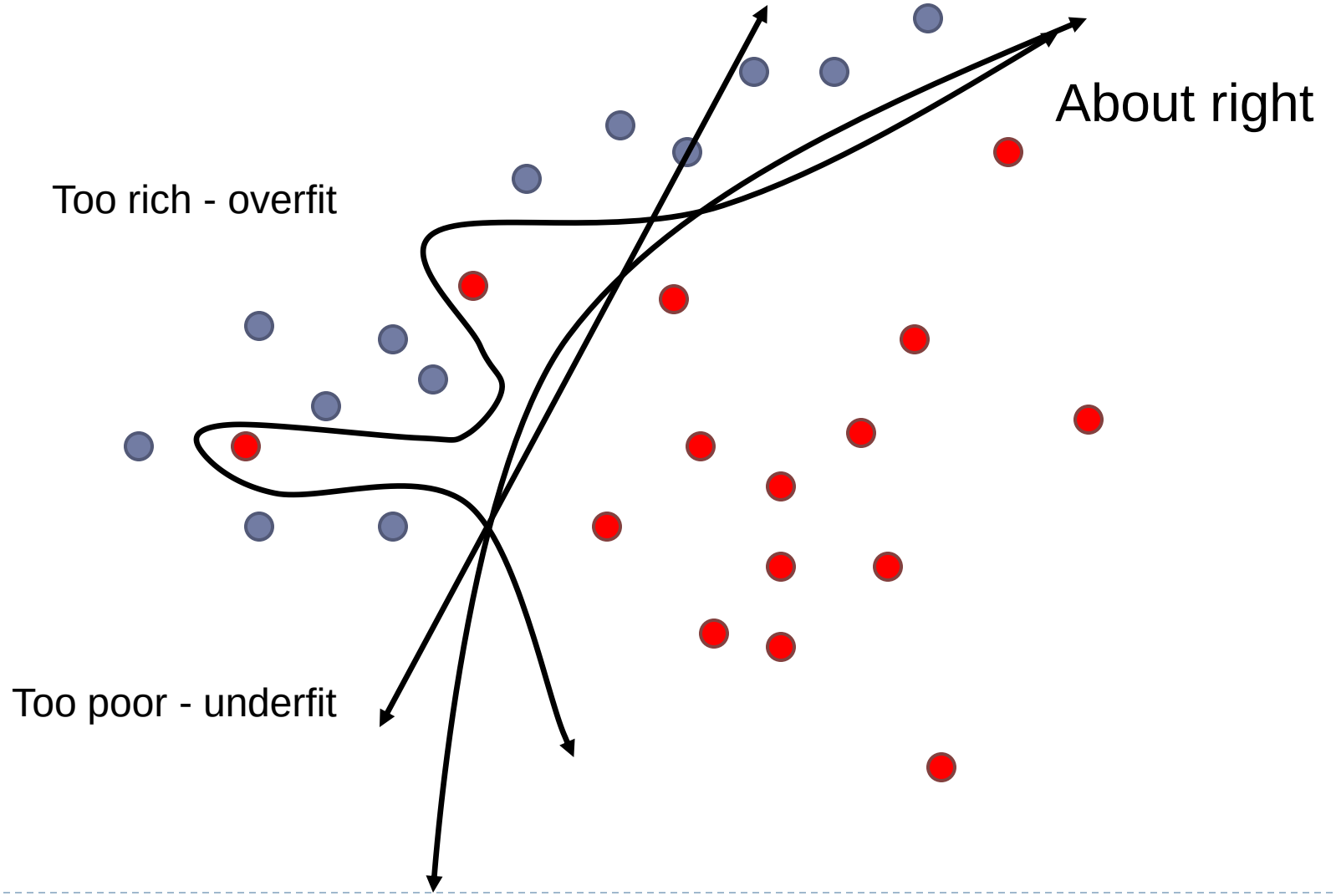
Remember -- we want small expected loss $L(c) = E_{X\times Y}\, 1_{c(x)\neq y}$.

(More on that later).

Key idea 1: control the space of functions $\mathcal{H}$ from which c is chosen.

$$c^*(x) = \operatorname{argmin}_{c\in\mathcal{H}} L_{emp}(c)$$

The “right” space of functions



Algorithmic issues

Suppose we found a "good" space $\mathcal{H}$ (more on that later).

The algorithmic issue remains:

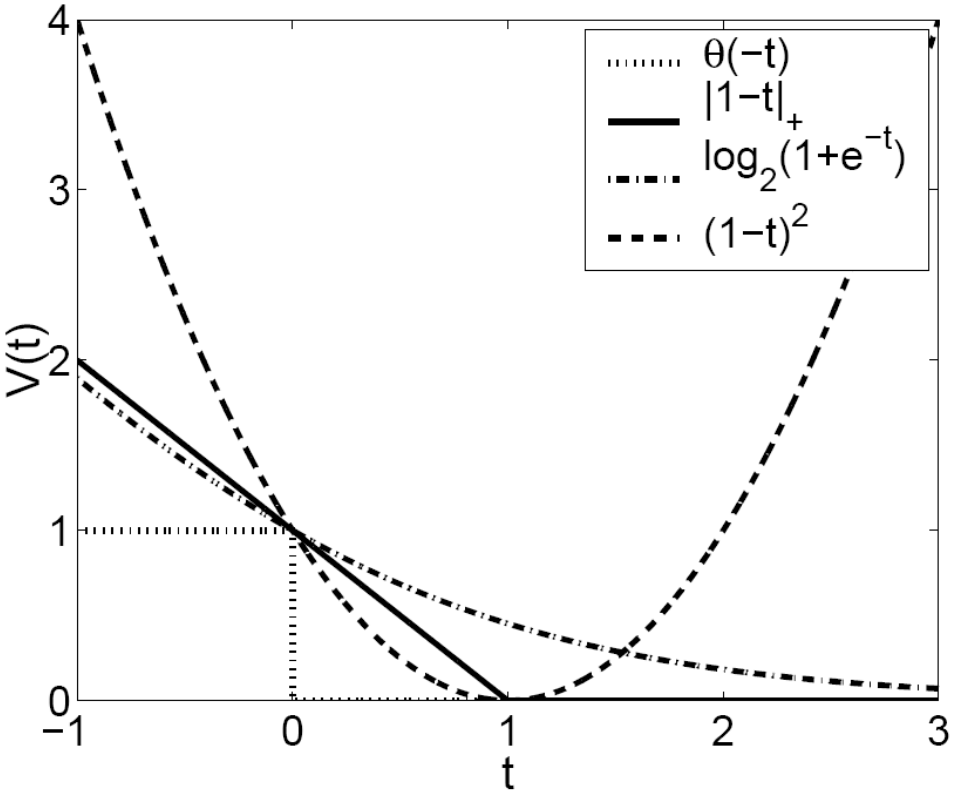
How do we actually find $c^*(x) = \operatorname{argmin}_{c \in \mathcal{H}} \frac{1}{n} \sum 1_{c(x_i) \neq y_i}$?

Generally hard discrete optimization problem (even for linear functions).

Key idea 2: make the loss function "nice".

Replace the indicator function $1_{c(x_i) \neq y_i}$ by a continuous/differentiable/convex function $l(c(x_i), y_i)$

Some popular loss functions



From Rosasco, et al, 03.

Square loss:

$$l(c(x), y) = (c(x) - y)^2$$

The "hinge loss":

$$l(c(x), y) = (1 - yc(x))_+$$

The logistic loss:

$$l(c(x), y) = \ln(1 + e^{-yc(x)})$$

Loss functions

How much do we lose by replacing the loss function?

Essentially equivalent to optimizing the binary loss.

Can be shown that in the infinite data setting minimizer of the square loss is the regression function $c^*(x) = E(y|x)$. $\text{sign}(c^*(x)) = c_{opt}(x)$ -- Bayes optimal.

Analogous results hold for other loss functions as well.

(But square loss also works for regression).

Ordinary least squares.

$\mathcal{H}$ is the set of linear functions $c(x)=<w,x>$

$$w^* = argmin_w \sum (y_i - < w, x_i >)^2$$

$$w^* = argmin_w \|y - Xw\|^2$$

Differentiating : $X^t X w^* = X^t y$

Ordinary least squares.

$\mathcal{H}$ is the set of linear functions $c(x)=<w,x>$

$$w^* = argmin_w \sum(y_i - < w, x_i >)^2$$

Differentiating : $X^t X w^* = X^t y$

What if $X^t X$ not invertible? Solution is not unique.

Idea: shrink the space of functions by controlling the norm $\|w\|$.

$\mathcal{H}$ is the set of linear functions $c(x)=<w,x>$, $\|w\| < t$

$$w^* = argmin_{\|w\|<t} \sum(y_i - < w, x_i >)^2$$

Equivalent but more convenient formulation (Tikhonov regularization).

$$w^* = argmin_w \sum (y_i - < w, x_i >)^2 + \lambda \|w\|^2$$

(Ridge regression, regularized least squares).

Solution: $(X^t X + \lambda I) w^* = X^t y$

Support Vector Machines

$$w^* = argmin_{||w||<t} \sum(1 - y_i < w, x_i >)_+$$

Equivalent formulation (Tikhonov regularization):

$$w^* = argmin_w \sum(1 - y_i < w, x_i >)_+ + \lambda \|w\|^2$$

This is what is typically called SVM. (You might have seen a formulations with "slack variables".)

A convex optimization problem.

Classically solved in the dual (introduced in that form by Vapnik in the 70's) but more recently direct (primal) optimization methods have become popular.

Beyond linear functions: choosing $\mathcal{H}$

Would like to have a much richer class of functions for inference.

Lots of candidates: polynomials, trigonometric functions, differentiable functions, Sobolev spaces …

How do we choose a good $\mathcal{H}$? What properties do we want?

Choosing $\mathcal{H}$

1. Should be a rich class with good approximation potential. But need a way of shrinking it as necessary.

2. Should have linear structure and inner products -- Hilbert Space.

Hilbert Space – vector space with inner products, complete.

Choosing $\mathcal{H}$: more technical

Functions which are close in $\mathcal{H}$ should make similar predictions.

$\|f - g\|$ is small means that $|f(x) - g(x)|$ is small.

More precisely

$$\forall_x \ |f(x) - g(x)| < C \ \|f - g\|$$

Evaluation functional $Eval_x(h) := h(x)$ is bounded.

Reproducing Kernel Hilbert Space

Reproducing Kernel Hilbert Space (RKHS).

Any Hilbert Space of function with bounded evaluation functionals.

Reproducing Kernel Hilbert Space

Sounds good but how do we get RKHS?

Mercer kernels:

Z is a compact metric space.

$K(x, y)$ is a continuous function $Z \times \mathrm{Z} \to \mathbb{R}$

1. $K(x, y) = K(y, x)$
2. $\forall\, x_1, \ldots, x_n$ the matrix $\mathrm{K_{i,j}} = K(x_i, x_j)$ is positive semi-definite (PSD). (No negative eigenvalues.)

Every Mercer kernel corresponds to a RKHS.

Some examples of PSD kernels

$K(x,y) = e^{-\frac{\|x-y\|^2}{\sigma^2}}$ Gaussian kernel

$K(x,y) = (1 + \langle x,y \rangle)^d$ Polynomial kernel

$K(x,y) = e^{-\frac{\|x-y\|}{\sigma}}$ Laplace kernel

Many others… More on that later.

RKHS defined by a kernel

Given a PSD kernel $K(x, y)$ construct $\mathcal{H}_K$:

1. Start with $\mathcal{H}_0 = \text{span}\{K(x,\cdot)\}$
2. Put inner product structure $\langle K(x,\cdot), K(y,\cdot)\rangle_H := K(x, y)$
3. Take $\mathcal{H}_K$ to be the completion of $\mathcal{H}_0$, with respect to the inner product.

Theorem: $\mathcal{H}_K$ is a RKHS.

Linear kernel

$K(x, y) =< x, y >$ psd (the Gramm matrix is PSD)

A linear combination $\sum \alpha_i K(x_i, y)$ is a linear function.

$\mathcal{H}_K$ is the set of linear functions.

Prototypical kernel.

Reproducing property

Theorem (the Representer theorem):

Let $\phi: \mathcal{H} \to \mathbb{R}$ be a bounded linear functional.

Then the exists h_ϕ, s.t. $\phi(f) = \langle f, h_\phi \rangle_H$

h_ϕ "represents" ϕ.

In RKHS evaluation functional $Eval_x(f) \coloneqq f(x)$ is bounded.

$Eval_x(f) = \langle f, K(x,\cdot) \rangle_H$

By construction of RKHS from a PSD kernel, but can also be taken as a definition of the kernel.

The reproducing property

$f(x) = Eval_x(f) = \langle f, K(x,\cdot)\rangle_H$

Immediately follows that $\mathrm{K}(\mathrm{x},\mathrm{y}) = \langle K(y,\cdot), K(x,\cdot)\rangle_H$

(hence reproducing kernel)

Hence:

$$|f(x)| = |\langle f, K(x,\cdot)\rangle_H| \leq \|f\|\|K(x,\cdot)\| = \|f\|\sqrt{K(x,x)}$$

$$|f(x) - g(x)| < \max_x \sqrt{K(x,x)}\ \|f - g\|$$

Functions, which are close in $\mathcal{H}$ give predictions which are similar.

Practical implications

Functions, which are close in $\mathcal{H}$ should give predictions which are similar.

If that seems reasonable, need to deal with RKHS.

Nice theory, so what?

Gives rise to natural inference framework:

$$f^* = argmin_{f \in \mathcal{H}, \|f\| \leq t} \sum_i l(f(x_i), y_i)$$

t controls "richness" of the space of functions.

as before equivalent to:

$$f^* = argmin_{f \in \mathcal{H}} \sum_i l(f(x_i), y_i) + \lambda \|f\|^2$$

Practical implications

$$f^* = argmin_{f \in \mathcal{H}} \sum_i l(f(x_i), y_i) + \lambda \|f\|^2$$

Solution: $f^*(x) = \sum_i \alpha_i K(x_i, x)$

Known as the Representer theorem (follows from the Representer theorem.)

Proof: see board.

Practical implications

$$f^* = argmin_{f \in \mathcal{H}} \sum_i l(f(x_i), y_i) + \lambda \|f\|^2$$

Solution: $f^*(x) = \sum_i \alpha_i K(x_i, x)$

If l is convex becomes a finite-dimensional convex optimization problem over $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)^t$.

For least squares explicit solution: $\boldsymbol{\alpha} = (K + \lambda \mathrm{I})^{-1} \boldsymbol{y}$,where $\mathrm{K_{ij}} = K(x_i, x_j)$

Hinge loss – Kernel SVM.

Some references

- Aronszajn, **Theory of Reproducing Kernels.**
- Wahba, **Spline Models for Observational Data.**
- Vapnik, **Statistical Learning Theory.**
- Schölkopf, Smola, **Learning with Kernels**
- Evgeniou, Pontil, Poggio, **Regularization Networks and Support Vector Machines**
- Smale, **On the Mathematical Foundations of Learning**
- Christianini, Shawe-Taylor, **An Introduction to Support Vector Machines**