Foundations of Machine Learning
Convex Optimization

Mehryar Mohri
Courant Institute and Google Research
mohri@cims.nyu.edu

Convex Optimization

Convexity

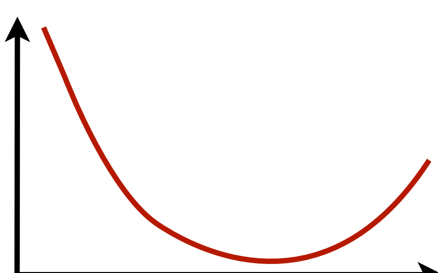
- Definition: $X \subseteq \mathbb{R}^N$ is said to be convex if for any two points $x, y \in X$ the segment $[x, y]$ lies in X:

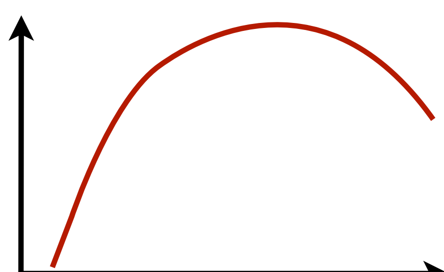
$$\{\alpha x + (1-\alpha)y, 0 \leq \alpha \leq 1\} \subseteq X.$$

- Definition: let X be a convex set. A function $f\colon X \to \mathbb{R}$ is said to be convex if for all $x, y \in X$ and $\alpha \in [0,1]$,

$$f(\alpha x + (1-\alpha)y) \leq \alpha f(x) + (1-\alpha)f(y).$$

With a strict inequality, f is said to be strictly convex. f is said to be concave when $-f$ is convex.

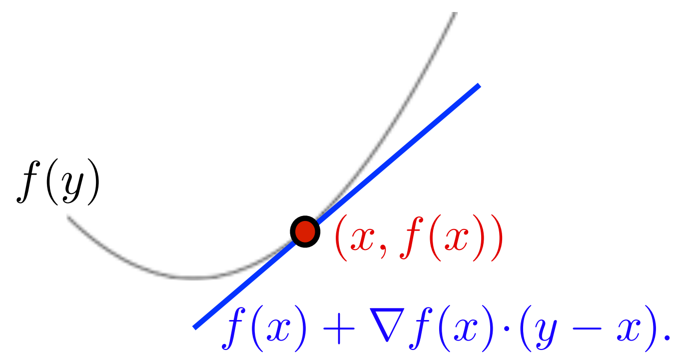




Properties of Convex Functions

- **Theorem**: let f be a differentiable function. Then, f is convex iff $\mathrm{dom}(f)$ is convex and

$$\forall x, y \in \mathrm{dom}(f), \;\; f(y) - f(x) \geq \nabla f(x) \cdot (y - x).$$



- **Theorem**: let f be a twice differentiable function. Then, f is convex iff its Hessian is positive semi-definite:

$$\forall x \in \mathrm{dom}(f), \;\; \nabla^2 f(x) \succeq 0.$$

Constrained Optimization Problem

- Problem: Let $X \subseteq \mathbb{R}^N$ and $f, g_i : X \to \mathbb{R}$, $i \in [1, m]$. A constrained optimization problem has the form:

$$\min_{\mathbf{x} \in X} \quad f(\mathbf{x})$$
$$\text{subject to:} \quad g_i(\mathbf{x}) \leq 0, i \in [1, m].$$

- Definition: The Lagrange function or Lagrangian associated to this problem is the function defined by:

$$\forall \mathbf{x} \in X, \forall \boldsymbol{\alpha} \geq 0, L(\mathbf{x}, \boldsymbol{\alpha}) = f(\mathbf{x}) + \sum_{i=1}^{m} \alpha_i g_i(x).$$

α_is are called Lagrange or dual variables.

Sufficient Condition

(Lagrange, 1797)

- **Theorem**: Let P be a constrained optimization problem over $X = \mathbb{R}^N$. If $(\mathbf{x}^*, \boldsymbol{\alpha}^*)$ is a saddle point, that is $\forall \mathbf{x} \in \mathbb{R}^N, \forall \boldsymbol{\alpha} \geq 0,\ L(\mathbf{x}^*, \boldsymbol{\alpha}) \leq L(\mathbf{x}^*, \boldsymbol{\alpha}^*) \leq L(\mathbf{x}, \boldsymbol{\alpha}^*)$, then it is a solution of P.

- **Proof**: By the first inequality,

$$\forall \boldsymbol{\alpha} \geq 0, L(\mathbf{x}^*, \boldsymbol{\alpha}) \leq L(\mathbf{x}^*, \boldsymbol{\alpha}^*) \Rightarrow \forall \boldsymbol{\alpha} \geq 0, \boldsymbol{\alpha} \cdot g(\mathbf{x}^*) \leq \boldsymbol{\alpha}^* \cdot g(\mathbf{x}^*)$$

$$(\text{use } \alpha \to +\infty \text{ then } \alpha \to 0) \Rightarrow \boxed{g(\mathbf{x}^*) \leq 0 \wedge \boldsymbol{\alpha}^* \cdot g(\mathbf{x}^*) = 0.}$$

  - In view of that, the second inequality gives

$$\forall \mathbf{x}, L(\mathbf{x}^*, \boldsymbol{\alpha}^*) \leq L(\mathbf{x}, \boldsymbol{\alpha}^*) \Rightarrow \forall \mathbf{x}, f(\mathbf{x}^*) \leq f(\mathbf{x}) + \boldsymbol{\alpha}^* \cdot g(\mathbf{x}).$$

Thus, for all x such that $g(x) \leq 0$, $\boxed{f(\mathbf{x}^*) \leq f(\mathbf{x}).}$

Constraint Qualification

- Definition: Assume that $\mathbf{int}\, X \neq \emptyset$. Then, the following is the strong constraint qualification or Slater's condition:

$$\exists\, \overline{\mathbf{x}} \in \mathbf{int}\, X\colon\ g(\overline{\mathbf{x}}) < 0.$$

- Definition: Assume that $\mathbf{int}\, X \neq \emptyset$. Then, the following is the weak constraint qualification or Slater's condition:

$$\exists\, \overline{\mathbf{x}} \in \mathbf{int}\, X\colon\ \forall i \in [1, m], \big(g_i(\overline{\mathbf{x}}) < 0\big) \vee \big(g_i(\overline{\mathbf{x}}) = 0 \wedge g_i \text{ affine}\big).$$

Necessary Conditions

- Theorem: Assume that f and g_i , $i \in [1, m]$, are convex functions and that Slater's condition holds. If $\mathbf{x}$ is a solution of the constrained optimization problem, then there exists $\alpha \geq 0$ such that $(\mathbf{x}, \boldsymbol{\alpha})$ is a saddle point of the Lagrangian.
- Theorem: Assume that f and g_i , $i \in [1, m]$, are convex differentiable functions and that the weak Slater's condition holds. If $\mathbf{x}$ is a solution of the constrained optimization problem, then there exists $\alpha \geq 0$ such that $(\mathbf{x}, \boldsymbol{\alpha})$ is a saddle point of the Lagrangian.

Kuhn-Tucker's Theorem

(Karush 1939; Kuhn-Tucker, 1951)

- **Theorem**: Assume that $f, g_i \colon X \to \mathbb{R}, i \in [1, m]$ are convex and differentiable and that the constraints are qualified. Then $\overline{\mathbf{x}}$ is a solution of the constrained program iff there exist $\overline{\boldsymbol{\alpha}} \geq 0$ such that:

$$\left.\begin{aligned}
&\nabla_{\mathbf{x}} L(\overline{\mathbf{x}}, \overline{\boldsymbol{\alpha}}) = \nabla_{\mathbf{x}} f(\overline{\mathbf{x}}) + \overline{\boldsymbol{\alpha}} \cdot \nabla_{\mathbf{x}} g(\overline{\mathbf{x}}) = 0 \\
&\nabla_{\boldsymbol{\alpha}} L(\overline{\mathbf{x}}, \overline{\boldsymbol{\alpha}}) = g(\overline{\mathbf{x}}) \leq 0 \\
&\overline{\boldsymbol{\alpha}} \cdot g(\overline{\mathbf{x}}) = \sum_{i=1}^{m} \overline{\boldsymbol{\alpha}}_i g_i(\overline{\mathbf{x}}) = 0 \,.
\end{aligned}\right\} \quad \text{KKT conditions}$$

- **Note**: Last two conditions equivalent to

$$\big(g(\overline{\mathbf{x}}) \leq 0\big) \wedge \big(\underbrace{\forall i \in [1, m], \bar{\alpha}_i g_i(\overline{\mathbf{x}}) = 0}_{\text{complementary conditions}}\big).$$

- Since the constraints are qualified, if $\overline{\mathbf{x}}$ is solution, then there exists $\overline{\boldsymbol{\alpha}}$ such that $(\overline{\mathbf{x}}, \overline{\boldsymbol{\alpha}})$ is a saddle point. In that case, the three conditions are verified (for the 3rd condition see proof of sufficient condition slide).
- Conversely, assume that the conditions are verified. Then, for any $\mathbf{x}$ such that $g(\mathbf{x}) < 0$,

$$
\begin{aligned}
f(\mathbf{x}) - f(\overline{\mathbf{x}}) &\geq \nabla_{\mathbf{x}} f(\overline{\mathbf{x}}) \cdot (\mathbf{x} - \overline{\mathbf{x}}) && \text{(convexity of } f\text{)} \\
&= -\sum_{i=1}^{m} \overline{\boldsymbol{\alpha}}_i \nabla_{\mathbf{x}} g_i(\overline{\mathbf{x}}) \cdot (\mathbf{x} - \overline{\mathbf{x}}) && \text{(first condition)} \\
&\geq -\sum_{i=1}^{m} \overline{\boldsymbol{\alpha}}_i [g_i(\mathbf{x}) - g_i(\overline{\mathbf{x}})] && \text{(convexity of } g_i\text{s)} \\
&= -\sum_{i=1}^{m} \overline{\boldsymbol{\alpha}}_i g_i(\mathbf{x}) \geq 0, && \text{(third condition)}
\end{aligned}
$$

Primal and Dual Problems

- Primal problem:

$$\min_{\mathbf{x} \in X} \quad f(\mathbf{x})$$
$$\text{subject to: } \quad g(\mathbf{x}) \leq 0.$$

- Dual problem:

$$\max_{\boldsymbol{\alpha}} \inf_{\mathbf{x} \in X} \quad L(\mathbf{x}, \boldsymbol{\alpha})$$
$$\text{subject to: } \quad \boldsymbol{\alpha} \geq 0.$$

Equivalent problems when constraints qualified.

Foundations of Machine Learning
Introduction to ML

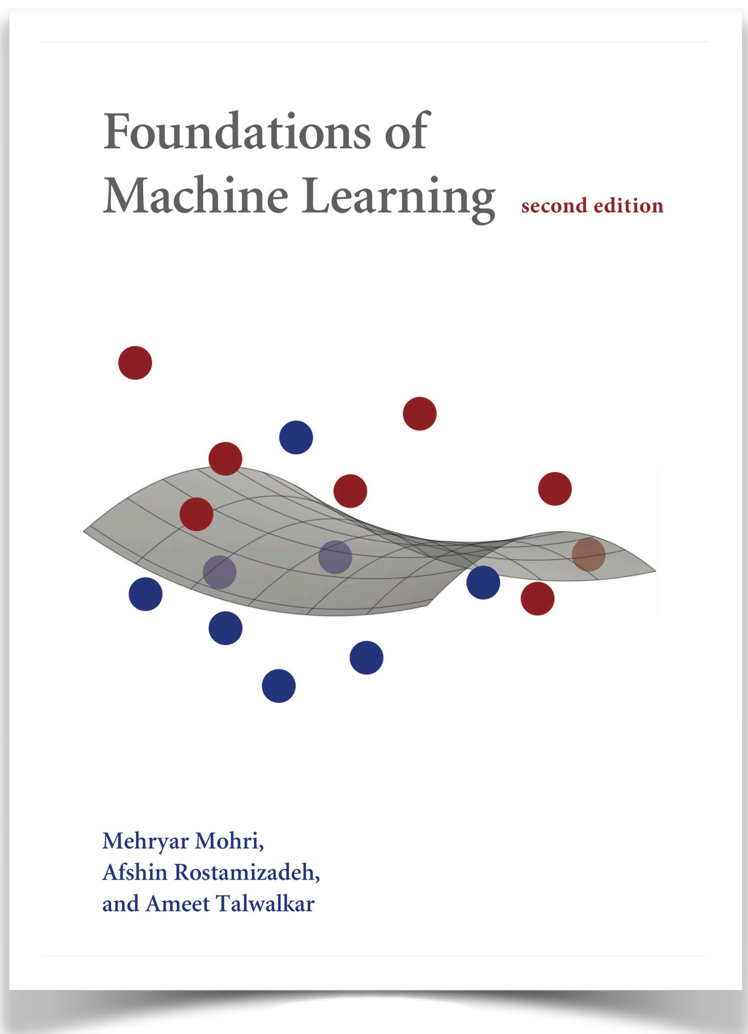
Mehryar Mohri
Courant Institute and Google Research
mohri@cims.nyu.edu

Logistics

- Prerequisites: basics in linear algebra, probability, and analysis of algorithms.
- Workload: about 3-4 homework assignments + project.
- Mailing list: join as soon as possible.

Course Material

- Textbook



- Slides: course web page.

  https://cs.nyu.edu/~mohri/ml24/

This Lecture

- Basic definitions and concepts.
- Introduction to the problem of learning.
- Probability tools.

Machine Learning

- Definition: computational methods using experience to improve performance.
- Experience: → data-driven task, thus statistics, probability, and optimization.
- Computer science: learning algorithms, analysis of complexity, theoretical guarantees.
- Example: use document word counts to predict its topic.

Examples of Learning Tasks

- Text: document classification, spam detection.
- Language: NLP tasks (e.g., morphological analysis, POS tagging, context-free parsing, dependency parsing).
- Speech: recognition, synthesis, verification.
- Image: annotation, face recognition, OCR, handwriting recognition.
- Games (e.g., chess, backgammon, go).
- Unassisted control of vehicles (robots, car).
- Medical diagnosis, fraud detection, network intrusion.

Some Broad ML Tasks

- Classification: assign a category to each item (e.g., document classification).
- Regression: predict a real value for each item (prediction of stock values, economic variables).
- Ranking: order items according to some criterion (relevant web pages returned by a search engine).
- Clustering: partition data into ‘homogenous’ regions (analysis of very large data sets).
- Dimensionality reduction: find lower-dimensional manifold preserving some properties of the data.

General Objectives of ML

- Theoretical questions:
  - what can be learned, under what conditions?
  - are there learning guarantees?
  - analysis of learning algorithms.
- Algorithms:
  - more efficient and more accurate algorithms.
  - deal with large-scale problems.
  - handle a variety of different learning problems.

This Course

- Theoretical foundations:
  - learning guarantees.
  - analysis of algorithms.
- Algorithms:
  - main mathematically well-studied algorithms.
  - discussion of their extensions.
- Applications:
  - illustration of their use.

Topics

- Probability tools, concentration inequalities.
- PAC learning model, Rademacher complexity, VC-dimension, generalization bounds.
- Support vector machines (SVMs), margin bounds, kernel methods.
- Ensemble methods, boosting.
- Logistic regression and conditional maximum entropy models.
- On-line learning, weighted majority algorithm, Perceptron algorithm, mistake bounds.
- Regression, generalization, algorithms.
- Ranking, generalization, algorithms.
- Reinforcement learning, MDPs, bandit problems and algorithm.

Definitions and Terminology

- Example: item, instance of the data used.
- Features: attributes associated to an item, often represented as a vector (e.g., word counts).
- Labels: category (classification) or real value (regression) associated to an item.
- Data:
  - training data (typically labeled).
  - test data (labeled but labels not seen).
  - validation data (labeled, for tuning parameters).

General Learning Scenarios

- Settings:
  - batch: learner receives full (training) sample, which he uses to make predictions for unseen points.
  - on-line: learner receives one sample at a time and makes a prediction for that sample.
- Queries:
  - active: the learner can request the label of a point.
  - passive: the learner receives labeled points.

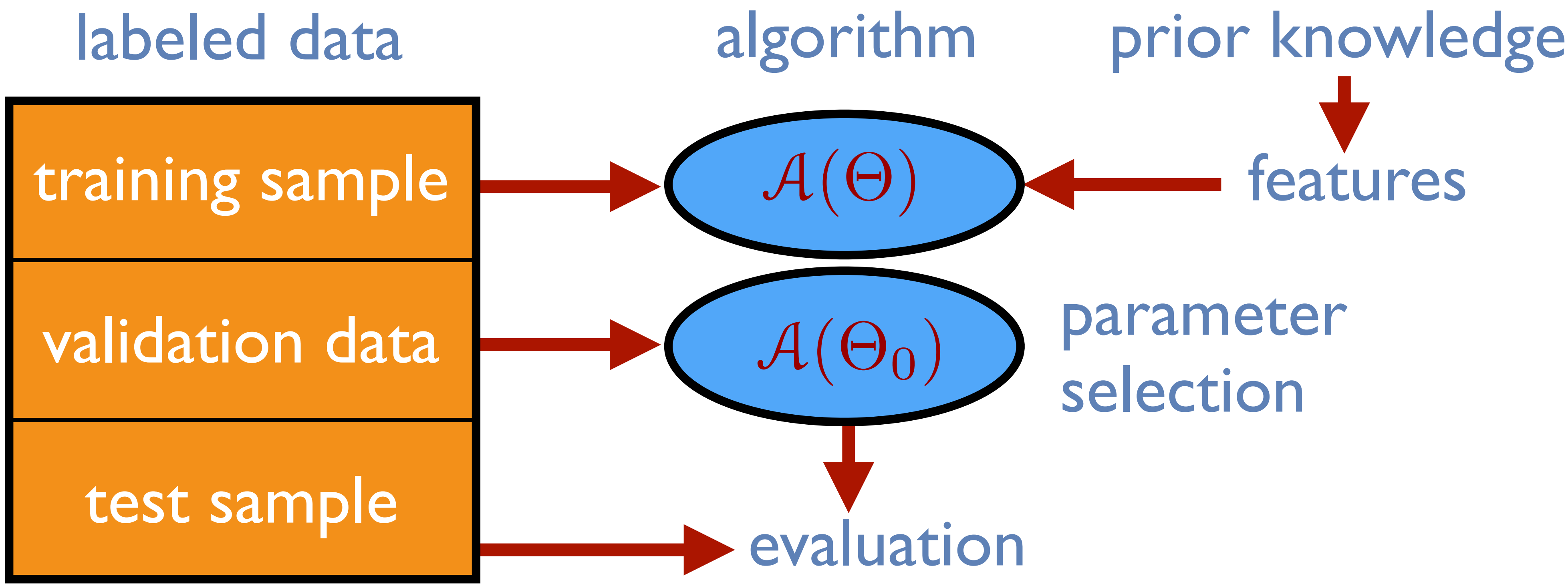
Standard Batch Scenarios

- Unsupervised learning: no labeled data.
- Supervised learning: uses labeled data for prediction on unseen points.
- Semi-supervised learning: uses labeled and unlabeled data for prediction on unseen points.
- Transduction: uses labeled and unlabeled data for prediction on seen points.

Example - SPAM Detection

- Problem: classify each e-mail message as SPAM or non-SPAM (binary classification problem).
- Potential data: large collection of SPAM and non-SPAM messages (labeled examples).

Learning Stages



This Lecture

- Basic definitions and concepts.
- Introduction to the problem of learning.
- Probability tools.

Definitions

- Spaces: input space X, output space Y.
- Loss function: $L\colon Y \times Y \to \mathbb{R}$.
  - $L(\widehat{y}, y)$: cost of predicting $\widehat{y}$ instead of y.
  - binary classification: 0-1 loss, $L(y, y') = 1_{y \neq y'}$.
  - regression: $Y \subseteq \mathbb{R}$, $l(y, y') = (y' - y)^2$.
- Hypothesis set: $H \subseteq Y^X$, subset of functions out of which the learner selects his hypothesis.
  - depends on features.
  - represents prior knowledge about task.

Supervised Learning Set-Up

- Training data: sample S of size m drawn i.i.d. from $X \times Y$ according to distribution D:

$$S = ((x_1, y_1), \ldots, (x_m, y_m)).$$

- Problem: find hypothesis $h \in H$ with small generalization error.
  - deterministic case: output label deterministic function of input, $y = f(x)$.
  - stochastic case: output probabilistic function of input.

Errors ⇒ They are essentially probabilities

- Generalization error: for $h \in H$, it is defined by

$$R(h) = \mathop{\mathrm{E}}_{(x,y)\sim D}[L(h(x), y)].$$ true error

(we don't have access to D)

- Empirical error: for $h \in H$ and sample S, it is

$$\widehat{R}(h) = \frac{1}{m} \sum_{i=1}^{m} L(h(x_i), y_i).$$

- Bayes error:

$$R^\star = \inf_{\substack{h \\ h \text{ measurable}}} R(h).$$ The absolute best error

  - in deterministic case, $R^\star = 0$.

Noise

- Noise:
  - in binary classification, for any $x \in X$,

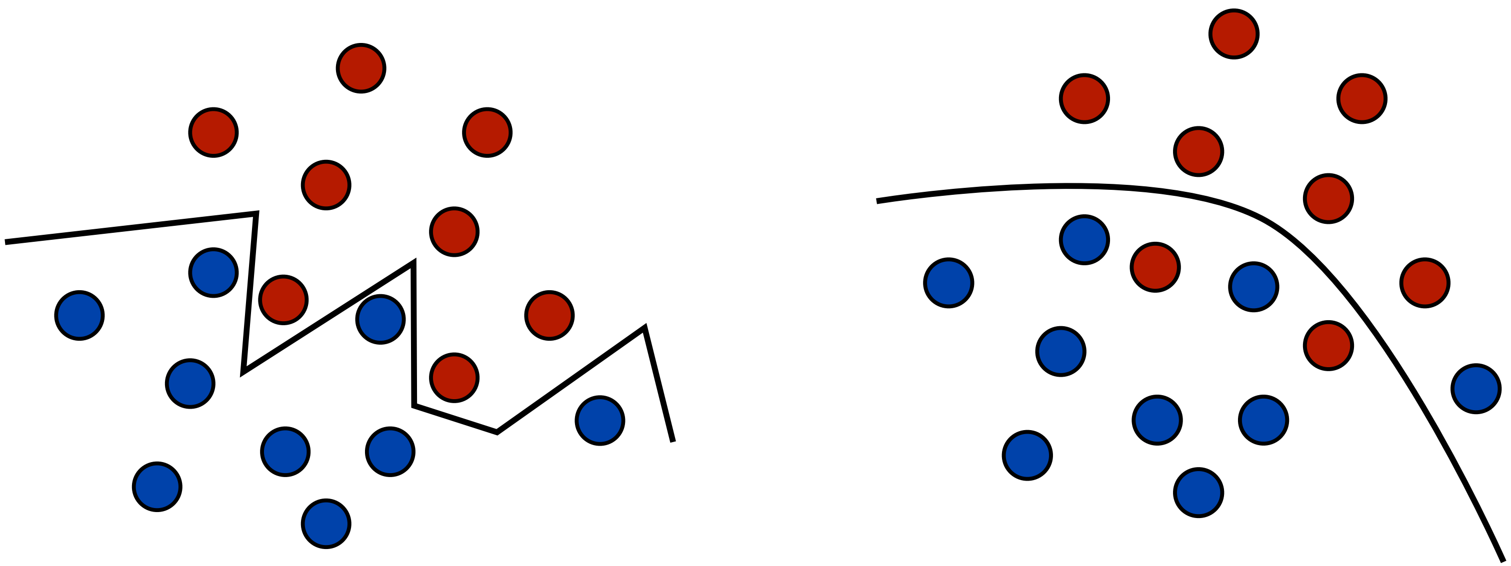
prob.

$$\text{noise}(x) = \min\{\Pr[1|x], \Pr[0|x]\}.$$

  - observe that $\text{E}[\text{noise}(x)] = R^*$.

what you suffer anyway
as you will pick the max

Learning ≠ Fitting



Notion of simplicity/complexity.

→ How do we define complexity?

Generalization

(Heart)

- Observations:
  - the best hypothesis on the sample may not be the best overall.
  - generalization is not memorization.
  - complex rules (very complex separation surfaces) can be poor predictors.
  - trade-off: complexity of hypothesis set vs sample size (underfitting/overfitting).

Model Selection

- General equality: for any $h \in H$,

best in class H

$$R(h) - R^* = \underbrace{[R(h) \overset{\geqslant}{-} R(h^*)]}_{\text{estimation}} + \underbrace{[R(h^*) - R^*]}_{\text{approximation}}.$$

- Approximation: not a random variable, only depends on H.
- Estimation: only term we can hope to bound.
- How should we choose H?

Empirical Risk Minimization

- Select hypothesis set H.
- Find hypothesis $h \in H$ minimizing empirical error:

$$h = \operatorname*{argmin}_{h \in H} \widehat{R}(h).$$

  - but H may be too complex.
  - the sample size may not be large enough.

Generalization Bounds

- Definition: upper bound on $\Pr\left[\sup_{h\in H} |R(h) - \widehat{R}(h)| > \epsilon\right]$.
- Bound on estimation error for hypothesis h_0 given by ERM:

$$\begin{aligned} R(h_0) - R(h^*) &= R(h_0) - \widehat{R}(h_0) + \widehat{R}(h_0) - R(h^*) \\ &\leq R(h_0) - \widehat{R}(h_0) + \widehat{R}(h^*) - R(h^*) \\ &\leq 2 \sup_{h\in H} |R(h) - \widehat{R}(h)|. \end{aligned}$$

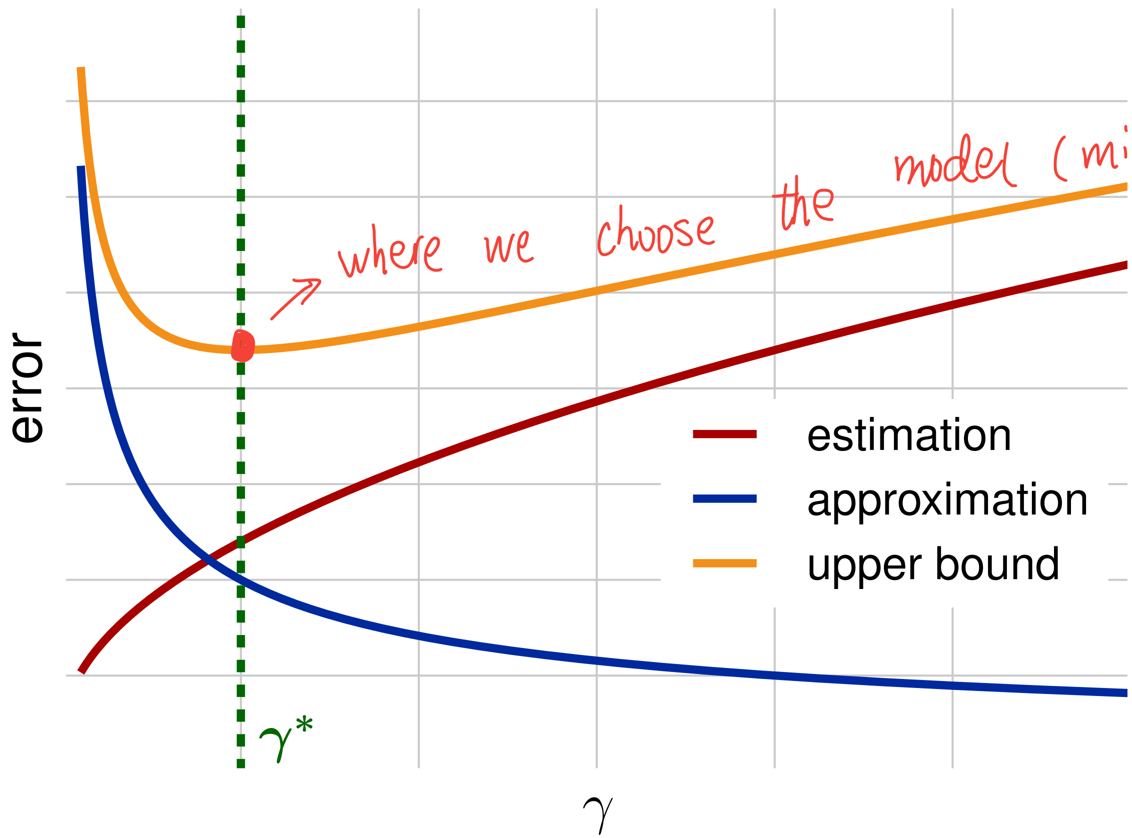
h_0 is best for $\widehat{R}$

h^* is best for R

for infinite dataset, h_0 could be h^*

→ How should we choose H? (model selection problem)

Model Selection



where we choose the model (min of orange, which is esti. + approx.)

$$\mathcal{H} = \bigcup_{\gamma \in \Gamma} \mathcal{H}_\gamma.$$

how complex $\mathcal{H}_\gamma$ is (e.g. degree)

Structural Risk Minimization

(Vapnik, 1995)

- Principle: consider an infinite sequence of hypothesis sets ordered for inclusion,

$$H_1 \subset H_2 \subset \cdots \subset H_n \subset \cdots$$

sample size

$$h = \operatorname*{argmin}_{h \in H_n, n \in \mathbb{N}} \widehat{R}(h) + \textbf{penalty}(H_n, m).$$

in particular, regularization

- strong theoretical guarantees.
- typically computationally hard.

General Algorithm Families

- Empirical risk minimization (ERM):

$$h = \operatorname*{argmin}_{h \in H} \widehat{R}(h).$$

- Structural risk minimization (SRM): $H_n \subseteq H_{n+1}$,

$$h = \operatorname*{argmin}_{h \in H_n, n \in \mathbb{N}} \widehat{R}(h) + \textbf{penalty}(H_n, m).$$

→ penalizing the complexity

- Regularization-based algorithms: $\lambda \geq 0$,

$$h = \operatorname*{argmin}_{h \in H} \widehat{R}(h) + \lambda \|h\|^2.$$

(can be viewed as a smooth version of SRM)

This Lecture

- Basic definitions and concepts.
- Introduction to the problem of learning.
- Probability tools.

Basic Properties

- Union bound: $\Pr[A \vee B] \leq \Pr[A] + \Pr[B]$.

??

- Inversion: if $\Pr[X \geq \epsilon] \leq f(\epsilon)$, then, for any $\delta > 0$, with probability at least $1 - \delta$, $X \leq f^{-1}(\delta)$.

- Jensen's inequality: if f is convex, $f(\mathrm{E}[X]) \leq \mathrm{E}[f(X)]$.

- Expectation: if $X \geq 0$, $\mathrm{E}[X] = \int_0^{+\infty} \Pr[X > t]\, dt$.

Basic Inequalities

- Markov's inequality: if $X \geq 0$ and $\epsilon > 0$, then

$$\Pr[X \geq \epsilon] \leq \frac{\mathrm{E}[X]}{\epsilon}.$$

- Chebyshev's inequality: for any $\epsilon > 0$,

Var(X)

$$\Pr[|X - \mathrm{E}[X]| \geq \epsilon] \leq \frac{\sigma_X^2}{\epsilon^2}.$$

Hoeffding's Inequality

- Theorem: Let $X_1, \ldots, X_m$ be indep. rand. variables with the same expectation μ and $X_i \in [a, b]$, ($a < b$). Then, for any $\epsilon > 0$, the following inequalities hold:

$$\Pr\left[\mu - \frac{1}{m}\sum_{i=1}^{m} X_i > \epsilon\right] \leq \exp\left(-\frac{2m\epsilon^2}{(b-a)^2}\right)$$

$$\Pr\left[\frac{1}{m}\sum_{i=1}^{m} X_i - \mu > \epsilon\right] \leq \exp\left(-\frac{2m\epsilon^2}{(b-a)^2}\right).$$

Hope: as long as the same size m is large enough,
we can estimate the expectation μ.

McDiarmid's Inequality

(McDiarmid, 1989)

- **Theorem**: let $X_1, \ldots, X_m$ be independent random variables taking values in U and $f: U^m \to \mathbb{R}$ a function verifying for all $i \in [1, m]$,

$$\sup_{x_1,\ldots,x_m,x_i'} |f(x_1,\ldots,x_i,\ldots,x_m) - f(x_1,\ldots,x_i',\ldots,x_m)| \leq c_i.$$

best: $c_i \in (\frac{1}{m^2}, \frac{1}{m})$

"Lipschitz condition"

we would like it to be dependent on m

if it's $\frac{1}{m}$, it's Hoffding's thm.

Then, for all $\epsilon > 0$,

$$\Pr\Big[\big|f(X_1,\ldots,X_m) - \mathrm{E}[f(X_1,\ldots,X_m)]\big| > \epsilon\Big] \leq 2\exp\left(-\frac{2\epsilon^2}{\sum_{i=1}^m c_i^2}\right).$$

Rmk. Hoffding's thm. is the special example for McDiamid's Ineq. by taking f to be the average function

Appendix

Markov's Inequality

- Theorem: let X be a non-negative random variable with $\mathrm{E}[X] < \infty$, then, for all $t > 0$,

$$\Pr[X \geq t\mathrm{E}[X]] \leq \frac{1}{t}.$$

- Proof:

$$\begin{aligned}
\Pr[X \geq t\,\mathrm{E}[X]] &= \sum_{x \geq t\mathrm{E}[X]} \Pr[X = x] \\
&\leq \sum_{x \geq t\,\mathrm{E}[X]} \Pr[X = x] \frac{x}{t\,\mathrm{E}[X]} \\
&\leq \sum_{x} \Pr[X = x] \frac{x}{t\,\mathrm{E}[X]} \\
&= \mathrm{E}\left[\frac{X}{t\,\mathrm{E}[X]}\right] = \frac{1}{t}.
\end{aligned}$$

Chebyshev's Inequality

- Theorem: let X be a random variable with $\mathrm{Var}[X] < \infty$, then, for all $t > 0$,

$$\Pr[|X - \mathrm{E}[X]| \geq t\sigma_X] \leq \frac{1}{t^2}.$$

- Proof: Observe that

$$\Pr[|X - \mathrm{E}[X]| \geq t\sigma_X] = \Pr[(X - \mathrm{E}[X])^2 \geq t^2\sigma_X^2].$$

The result follows Markov's inequality.

Weak Law of Large Numbers

- Theorem: let $(X_n)_{n\in\mathbb{N}}$ be a sequence of independent random variables with the same mean μ and variance $\sigma^2 < \infty$ and let $\overline{X}_n = \frac{1}{n}\sum_{i=1}^{n} X_i$, then, for any $\epsilon > 0$,

$$\lim_{n\to\infty} \Pr[|\overline{X}_n - \mu| \geq \epsilon] = 0.$$

- Proof: Since the variables are independent,

$$\mathrm{Var}[\overline{X}_n] = \sum_{i=1}^{n} \mathrm{Var}\left[\frac{X_i}{n}\right] = \frac{n\sigma^2}{n^2} = \frac{\sigma^2}{n}.$$

- Thus, by Chebyshev's inequality,

$$\Pr[|\overline{X}_n - \mu| \geq \epsilon] \leq \frac{\sigma^2}{n\epsilon^2}.$$

Concentration Inequalities

- Some general tools for error analysis and bounds:
  - Hoeffding's inequality (additive).
  - Chernoff bounds (multiplicative).
  - McDiarmid's inequality (more general).

Hoeffding's Inequality

- Corollary: for any $\epsilon > 0$, any distribution D and any hypothesis $h \colon X \to \{0, 1\}$, the following inequalities hold:

$$\Pr[\widehat{R}(h) - R(h) \geq \epsilon] \leq e^{-2m\epsilon^2}$$
$$\Pr[\widehat{R}(h) - R(h) \leq -\epsilon] \leq e^{-2m\epsilon^2}.$$

- Proof: follows directly Hoeffding's theorem.

- Combining these one-sided inequalities yields

$$\Pr\left[\left|\widehat{R}(h) - R(h)\right| \geq \epsilon\right] \leq 2e^{-2m\epsilon^2}.$$

Chernoff's Inequality

- Theorem: for any $\epsilon > 0$, any distribution D and any hypothesis $h\colon X \to \{0,1\}$, the following inequalities hold:

$$\Pr[\widehat{R}(h) \geq (1+\epsilon)R(h)] \leq e^{-m\,R(h)\,\epsilon^2/3}$$

$$\Pr[\widehat{R}(h) \leq (1-\epsilon)R(h)] \leq e^{-m\,R(h)\,\epsilon^2/2}.$$

- Proof: proof based on Chernoff's bounding technique.

McDiarmid's Inequality

(McDiarmid, 1989)

- Theorem: let $X_1, \ldots, X_m$ be independent random variables taking values in U and $f\colon U^m \to \mathbb{R}$ a function verifying for all $i \in [1, m]$,

$$\sup_{x_1,\ldots,x_m,x'_i} \left|f(x_1,\ldots,x_i,\ldots,x_m) - f(x_1,\ldots,x'_i,\ldots,x_m)\right| \leq c_i.$$

Then, for all $\epsilon > 0$,

$$\Pr\Big[\big|f(X_1,\ldots,X_m) - \mathrm{E}[f(X_1,\ldots,X_m)]\big| > \epsilon\Big] \leq 2\exp\left(-\frac{2\epsilon^2}{\sum_{i=1}^m c_i^2}\right).$$

- Comments:
  - Proof: uses Hoeffding's lemma.
  - Hoeffding's inequality is a special case of McDiarmid's with

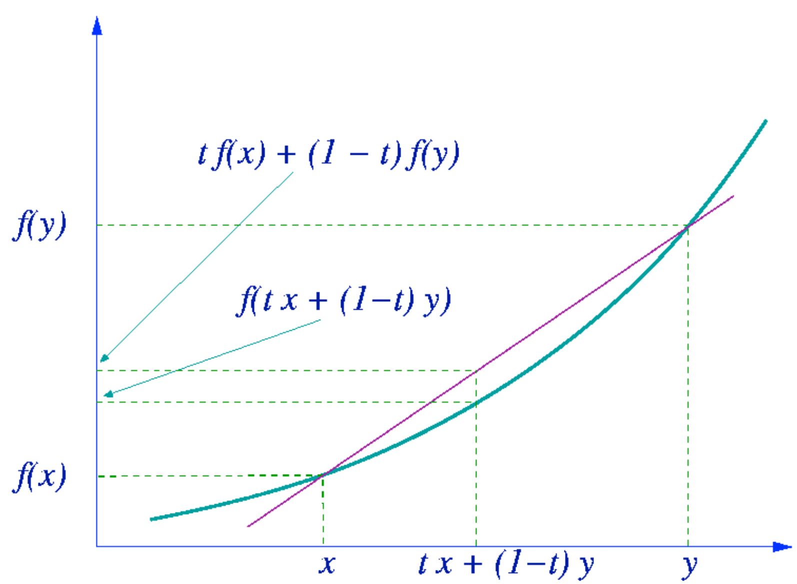
$$f(x_1, \ldots, x_m) = \frac{1}{m} \sum_{i=1}^{m} x_i \quad \text{and} \quad c_i = \frac{|b_i - a_i|}{m}.$$

Jensen's Inequality

- Theorem: let X be a random variable and f a measurable convex function. Then,

$$f(\mathrm{E}[X]) \leq \mathrm{E}[f(X)].$$

- Proof: definition of convexity, continuity of convex functions, and density of finite distributions.



Foundations of Machine Learning

Learning with Finite Hypothesis Sets

Mehryar Mohri
Courant Institute and Google Research
mohri@cims.nyu.edu

Motivation

- Some computational learning questions
  - What can be learned efficiently?
  - What is inherently hard to learn?
  - A general model of learning?
- Complexity
  - Computational complexity: time and space.
  - Sample complexity: amount of training data needed to learn successfully.
  - Mistake bounds: number of mistakes before learning successfully.

This lecture

- PAC Model
- Sample complexity, finite *H*, consistent case
- Sample complexity, finite *H*, inconsistent case

Definitions and Notation

- X: set of all possible instances or examples, e.g., the set of all men and women characterized by their height and weight.

→ we start with binary classification

- $c \colon X \to \{0, 1\}$: the target concept to learn; can be identified with its support $\{x \in X \colon c(x) = 1\}$.

(≠0)

- C: concept class, a set of target concepts c.
- D: target distribution, a fixed probability distribution over X. Training and test examples are drawn according to D.

Definitions and Notation

- S : training sample.
- H : set of concept hypotheses, e.g., the set of all linear classifiers.
- The learning algorithm receives sample S and selects a hypothesis h_S from H approximating c .

Errors

- True error or generalization error of h with respect to the target concept c and distribution D:

$$R(h) = \Pr_{x \sim D}[h(x) \neq c(x)] = \mathop{\mathrm{E}}_{x \sim D}[1_{h(x) \neq c(x)}].$$

- Empirical error: average error of h on the training sample S drawn according to distribution D,

$$\widehat{R}_S(h) = \Pr_{x \sim \widehat{D}}[h(x) \neq c(x)] = \mathop{\mathrm{E}}_{x \sim \widehat{D}}[1_{h(x) \neq c(x)}] = \frac{1}{m} \sum_{i=1}^{m} 1_{h(x_i) \neq c(x_i)}.$$

empirical distribution

- Note: $R(h) = \mathop{\mathrm{E}}_{S \sim D^m}\left[\widehat{R}_S(h)\right].$

PAC Model

(Valiant, 1984)

- PAC learning: Probably Approximately Correct learning.

  $\varepsilon \rightarrow$ error
  $\delta \rightarrow$ prob.

- Definition: concept class C is PAC-learnable if there exists a learning algorithm L such that:
  - for all $c \in C, \epsilon > 0, \delta > 0$, and all distributions D,

D is indep. of L

$$\Pr_{S \sim D^m}[R(h_S) \leq \epsilon] \geq 1 - \delta,$$

(very strict on L as D itself is indep.)

  - for samples S of size $m = poly(1/\epsilon, 1/\delta)$ for a fixed polynomial.

Remarks

- Concept class C is known to the algorithm.
- Distribution-free model: no assumption on D.
- Both training and test examples drawn $\sim D$.
  (may be not the same D)
- Probably: confidence $1-\delta$.
- Approximately correct: accuracy $1-\epsilon$.
- Efficient PAC-learning: L runs in time $poly(1/\epsilon, 1/\delta)$.
- What about the cost of the representation of $c \in C$?

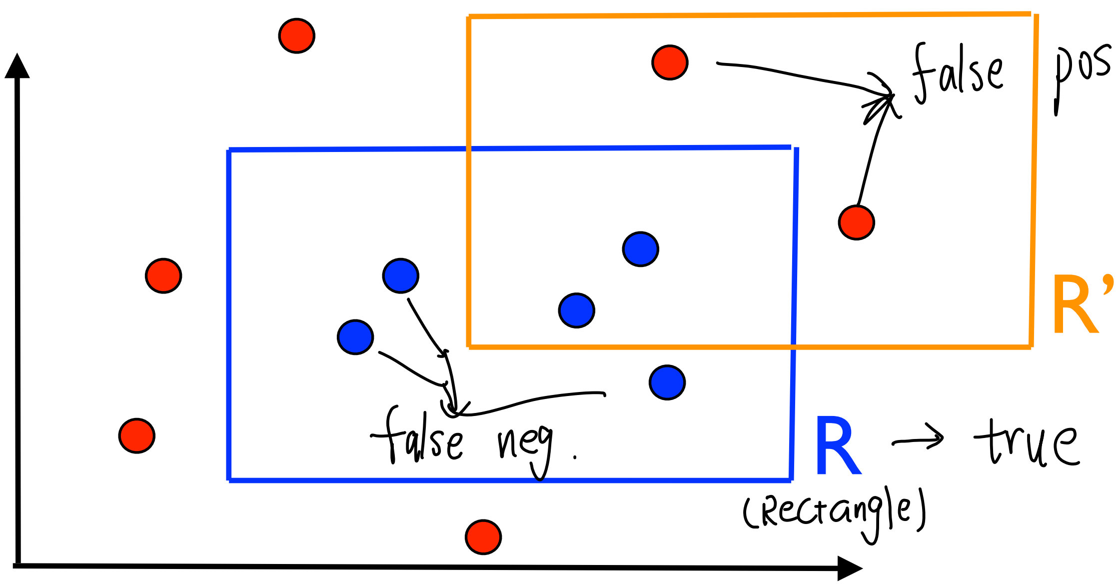
PAC Model - New Definition

- Computational representation:
  - cost for $x \in X$ in $O(n)$.
  - cost for $c \in C$ in $O(size(c))$.
- Extension: running time.

$$O(poly(1/\epsilon, 1/\delta)) \longrightarrow O(poly(1/\epsilon, 1/\delta, n, size(c))).$$

Example - Rectangle Learning

- Problem: learn unknown axis-aligned rectangle R using as small a labeled sample as possible.



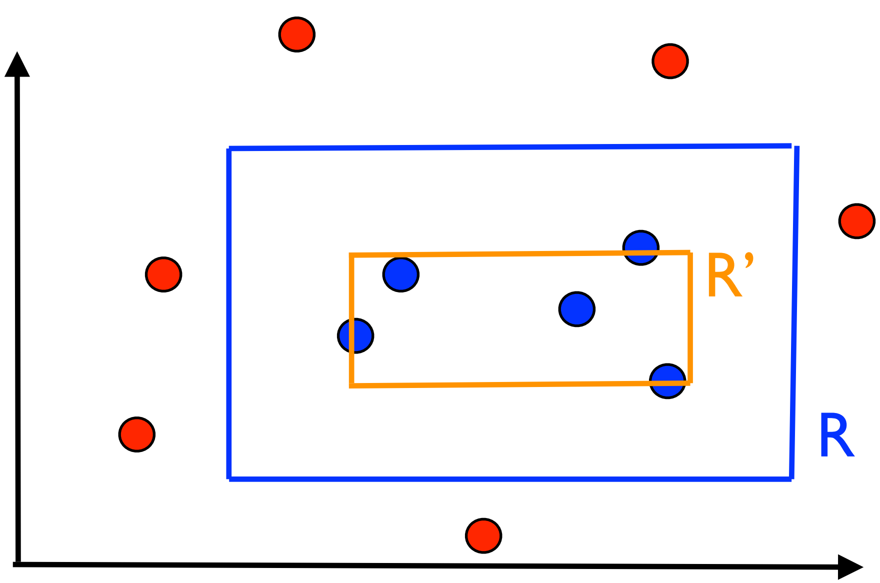
R' → concept that L selects out

R → true concept

- Hypothesis: rectangle R'. In general, there may be false positive and false negative points.

Example - Rectangle Learning

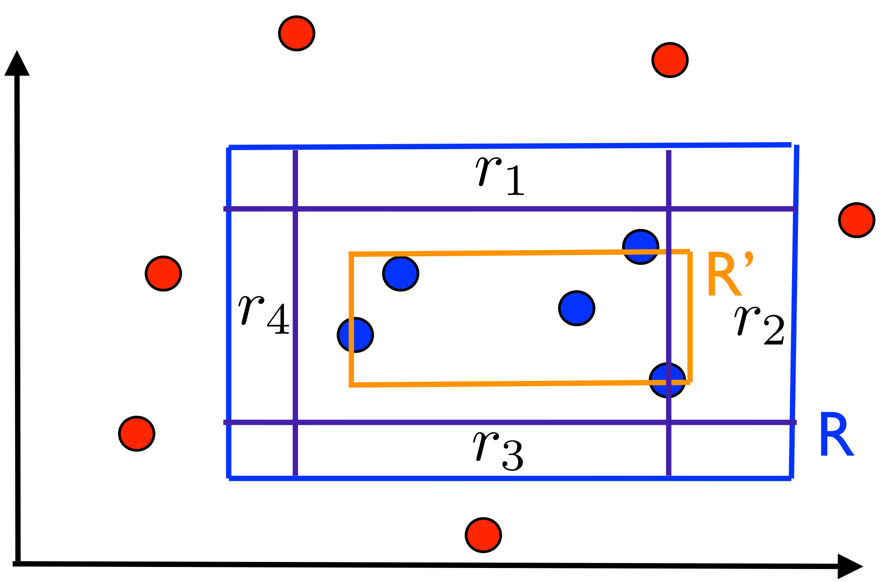
- Simple method: choose tightest consistent rectangle R' for a large enough sample. How large a sample? Is this class PAC-learnable?



- What is the probability that $R(\mathrm{R}') > \epsilon$?

Example - Rectangle Learning

- Fix $\epsilon > 0$ and assume $\Pr_D[\mathrm{R}] > \epsilon$ (otherwise the result is trivial).
- Let r_1, r_2, r_3, r_4 be four smallest rectangles along the sides of R such that $\Pr_D[r_i] \geq \frac{\epsilon}{4}$.



$$\mathrm{R} = [l, r] \times [b, t]$$
$$r_4 = [l, s_4] \times [b, t]$$
$$s_4 = \inf\{s \colon \Pr\left[[l, s] \times [b, t]\right] \geq \tfrac{\epsilon}{4}\}$$
$$\Pr_D\left[[l, s_4[\times [b, t]\right] < \tfrac{\epsilon}{4}$$

Example - Rectangle Learning

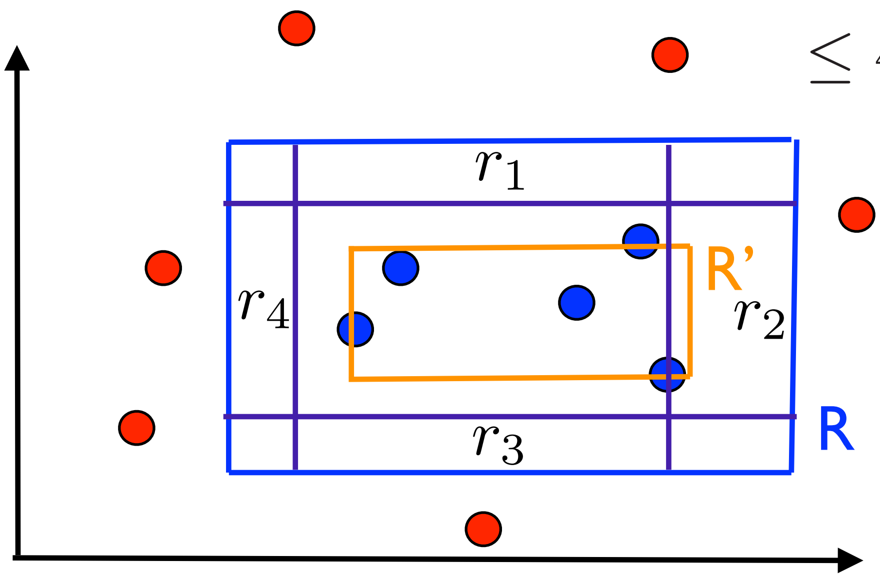
- Errors can only occur in $\mathsf{R}-\mathsf{R}'$. Thus (geometry),

  $R(\mathsf{R}') > \epsilon \Rightarrow \mathsf{R}'$ misses at least one region r_i.

- Therefore, $\Pr[R(\mathsf{R}') > \epsilon] \leq \Pr[\cup_{i=1}^4 \{\mathsf{R}' \text{ misses } r_i\}]$

$$\leq \sum_{i=1}^4 \Pr[\{\mathsf{R}' \text{ misses } r_i\}]$$

$$\leq 4(1-\tfrac{\epsilon}{4})^m \leq 4e^{-\frac{m\epsilon}{4}}.$$



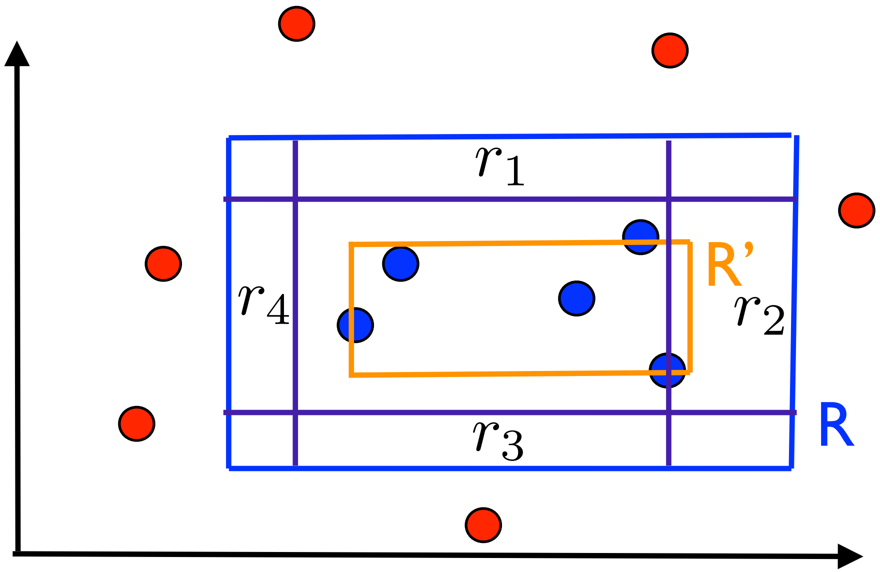
Example - Rectangle Learning

- Set $\delta > 0$ to match the upper bound:

$$4e^{-\frac{m\epsilon}{4}} \leq \delta \Leftrightarrow m \geq \frac{4}{\epsilon} \log \frac{4}{\delta}.$$

- Then, for $m \geq \frac{4}{\epsilon} \log \frac{4}{\delta}$, with probability at least $1 - \delta$,

$$R(\mathsf{R}') \leq \epsilon.$$



Notes

- Infinite hypothesis set, but simple proof.
- Does this proof readily apply to other similar concepts classes?
- Geometric properties:
  - key in this proof.
  - in general non-trivial to extend to other classes, e.g., non-concentric circles (see HW2, 2006).

→ Need for more general proof and results.

This lecture

- PAC Model
- Sample complexity, finite H, consistent case
- Sample complexity, finite H, inconsistent case

Learning Bound for Finite H - Consistent Case

- **Theorem**: let H be a finite set of functions from X to $\{0, 1\}$ and L an algorithm that for any target concept $c \in H$ and sample S returns a consistent hypothesis h_S: $\widehat{R}_S(h_S) = 0$. Then, for any $\delta > 0$, with probability at least $1 - \delta$,

Generalization Bounds

$$R(h_S) \leq \frac{1}{m}\left(\log |H| + \log \frac{1}{\delta}\right).$$

$\Updownarrow$ two identical forms of the ineq.

$\Leftrightarrow$ for $\forall \varepsilon, \delta > 0$, $\mathbb{P}_{S \sim D^m}[R(h_S) \leq \varepsilon] \geq 1 - \delta$ holds if

$m \geq \frac{1}{\varepsilon}\left(\log |H| + \log \frac{1}{\delta}\right)$ Sample Complexity Bounds

$\Updownarrow$

$\varepsilon \leq \frac{1}{m}\left(\log |H| + \log \frac{1}{\delta}\right)$

Learning Bound for Finite H - Consistent Case

as we consider $\mathcal{Y} = \{0, 1\}$

$$\mathbb{P}_{\underset{(\text{sum})}{x \sim D}}\big(\underbrace{h(x) \neq c(x)}_{\text{a condition } x \text{ should satisfy}}\big) > \varepsilon$$

$$= \mathbb{E}_{x \sim D} 1_{\{h(x) \neq c(x)\}}$$

- **Proof**: for any $\epsilon > 0$, define $H_\epsilon = \{h \in H : R(h) > \epsilon\}$. We want to prove that, with high probability, if h_S is consistent, then it has low error:

$$\underbrace{\mathbb{P}\Big[\widehat{R}_S(h_S) = 0 \overset{\text{the ERM algorithm}}{\Rightarrow} R(h_S) \leq \epsilon\Big] \geq 1 - \delta}_{\text{the inequality we want}} \Leftrightarrow \mathbb{P}\Big[\widehat{R}_S(h_S) = 0 \wedge R(h_S) > \epsilon\Big] \leq \delta$$

$$\Leftrightarrow \underbrace{\mathbb{P}\Big[\widehat{R}_S(h_S) = 0 \wedge h_S \in H_\epsilon\Big]}_{(*)} \leq \delta.$$

instead of finding specific h_S, we bound on $\forall h$ s.t. $\widehat{R}_S(h) = 0$

$$(*) \leq \mathbb{P}\Big[\exists h \in H : \widehat{R}_S(h) = 0 \wedge h \in H_\epsilon\Big]$$

$$= \mathbb{P}\Big[\exists h \in H_\epsilon : \widehat{R}_S(h) = 0\Big]$$

$$= \mathbb{P}\Big[\widehat{R}_S(h_1) = 0 \vee \ldots \vee \widehat{R}_S(h_{|H_\epsilon|}) = 0\Big]$$

$$\leq \sum_{h \in H_\epsilon} \mathbb{P}\Big[\widehat{R}_S(h) = 0\Big]$$

$$\leq \sum_{h \in H_\epsilon} (1 - \epsilon)^m \leq |H|(1 - \epsilon)^m \leq \underbrace{|H|e^{-m\epsilon}}_{\text{take it as } \delta}.$$

Rmk. $\overset{R(h) =}{\mathbb{P}_{x \sim D}}\big(h(x) \neq c(x)\big) > \varepsilon$ means the sum of prob. of pick $x \sim D$ which makes false prediction is greater than ε,

hence $\mathbb{P}_{h \in H_\varepsilon}[\underbrace{\widehat{R}_S(h) = 0}_{\text{taking consistent } m \text{ points from } D \text{ points}}] \leq (1-\varepsilon)^m$ (missing all inconsisten

where S is a sample of size m

Remarks

(we have a uniform bound actually)

(one way to choose h_S)

$\exists$ solution for ERM $= 0$

- The algorithm can be ERM if problem realizable.
- Error bound linear in $\frac{1}{m}$ and only logarithmic in $\frac{1}{\delta}$.
- $\log_2 |H|$ is the number of bits used for the representation of H.
- Bound is loose for large $|H|$.
- Uninformative for infinite $|H|$.

Conjunctions of Boolean Literals

- Example for $n=6$.
- Algorithm: start with $x_1 \wedge \overline{x}_1 \wedge \cdots \wedge x_n \wedge \overline{x}_n$ and rule out literals incompatible with positive examples.

0	1	1	0	1	1	+
0	1	1	1	1	1	+
0	0	1	1	0	1	-
0	1	1	1	1	1	+
1	0	0	1	1	0	-
0	1	0	0	1	1	+
0	1	?	?	1	1	

$\rightarrow \overline{x}_1 \wedge x_2 \wedge x_5 \wedge x_6.$

Intersections

Conjunctions of Boolean Literals

- **Problem**: learning class C_n of conjunctions of boolean literals with at most n variables (e.g., for $n=3$, $x_1 \wedge \overline{x_2} \wedge x_3$).

- **Algorithm**: choose h consistent with S.
  - Since $|H|=|C_n|=3^n$, sample complexity:

$$m \geq \frac{1}{\epsilon}((\log 3)\, n + \log \frac{1}{\delta}).$$

$\delta = .02, \epsilon = .1, n = 10, m \geq 149.$

  - Computational complexity: polynomial, since algorithmic cost per training example is in $O(n)$.

This lecture

① deterministic: $\exists! f: \mathcal{X} \to \mathcal{Y}$ (each x_i has prob 1 relating to label y_i)

② consistent: $\exists h \in H$ s.t. $\hat{R}(h) = 0$.

② is stricter than ① as even if deterministic, we still might not achieve consistency.

- PAC Model
- Sample complexity, finite *H*, consistent case
- Sample complexity, finite *H*, inconsistent case

Inconsistent Case

- No $h \in H$ is a consistent hypothesis.
- The typical case in practice: difficult problems, complex concept class.
- But, inconsistent hypotheses with a small number of errors on the training set can be useful.
- Need a more powerful tool: Hoeffding's inequality.

Hoeffding's Inequality

- Corollary: for any $\epsilon > 0$ and any hypothesis $h : X \to \{0, 1\}$ the following inequalities holds:

$$\Pr[R(h) - \widehat{R}(h) \geq \epsilon] \leq e^{-2m\epsilon^2}$$
$$\Pr[\widehat{R}(h) - R(h) \geq \epsilon] \leq e^{-2m\epsilon^2}.$$

- Combining these one-sided inequalities yields

$$\Pr[|R(h) - \widehat{R}(h)| \geq \epsilon] \leq 2e^{-2m\epsilon^2}.$$

Application to Learning Algorithm?

- Can we apply that bound to the hypothesis h_S returned by our learning algorithm when training on sample S?

- No, because h_S is not a fixed hypothesis, it depends on the training sample. Note also that $\underbrace{\mathrm{E}[\widehat{R}(h_S)]}_{\text{a R.V. depending on } S}$ is not a simple quantity such as $R(h_S)$.

- Instead, we need a bound that holds simultaneously for all hypotheses $h \in H$, a uniform convergence bound.

Generalization Bound - Finite H

- **Theorem**: let H be a finite hypothesis set, then, for any $\delta > 0$, with probability at least $1 - \delta$,

$$\forall h \in H,\ R(h) \leq \widehat{R}_S(h) + \sqrt{\frac{\log|H| + \log\frac{2}{\delta}}{2m}}.$$

- **Proof**: By the union bound,

We still derive a union bound.

$$\Pr\left[\max_{h\in H}\left|R(h) - \widehat{R}_S(h)\right| > \epsilon\right]$$
$$= \Pr\left[\left|R(h_1) - \widehat{R}_S(h_1)\right| > \epsilon \vee \ldots \vee \left|R(h_{|H|}) - \widehat{R}_S(h_{|H|})\right| > \epsilon\right]$$
$$\leq \sum_{h\in H} \Pr\left[\left|R(h) - \widehat{R}_S(h)\right| > \epsilon\right]$$
$$\leq 2|H|\exp(-2m\epsilon^2).$$

Remarks

- Thus, for a finite hypothesis set, whp,

$$\forall h \in H, R(h) \leq \widehat{R}_S(h) + O\left(\sqrt{\frac{\log |H|}{m}}\right).$$

- Error bound in $O(\frac{1}{\sqrt{m}})$ (quadratically worse).
- $\log_2 |H|$ can be interpreted as the number of bits needed to encode H.
- Occam's Razor principle (theologian William of Occam): "plurality should not be posited without necessity".

There is a trade of between reducing $\widehat{R}(h)$ and controlling m.

Occam's Razor

- Principle formulated by controversial theologian William of Occam: "plurality should not be posited without necessity", rephrased as "the simplest explanation is best";
  - invoked in a variety of contexts, e.g., syntax. Kolmogorov complexity can be viewed as the corresponding framework in information theory.
  - here, to minimize true error, choose the most parsimonious explanation (smallest $|H|$). choose simplest hypothsis set.
  - we will see later other applications of this principle.

Lecture Summary

- C is PAC-learnable if $\exists L, \forall c \in C, \forall \epsilon, \delta > 0, m = P\left(\frac{1}{\epsilon}, \frac{1}{\delta}\right)$,
$$\Pr_{S \sim D^m}[R(h_S) \leq \epsilon] \geq 1 - \delta.$$
- Learning bound, finite H consistent case:
$$R(h) \leq \frac{1}{m}\left(\log|H| + \log\frac{1}{\delta}\right).$$
- Learning bound, finite H inconsistent case:
$$R(h) \leq \widehat{R}_S(h) + \sqrt{\frac{\log|H| + \log\frac{2}{\delta}}{2m}}.$$
- How do we deal with infinite hypothesis sets?

References

- Anselm Blumer, A. Ehrenfeucht, David Haussler, and Manfred K. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *Journal of the ACM (JACM)*, Volume 36, Issue 4, 1989.

- Michael Kearns and Umesh Vazirani. *An Introduction to Computational Learning Theory*, MIT Press, 1994.

- Leslie G. Valiant. *A Theory of the Learnable*, Communications of the ACM 27(11):1134–1142 (1984).

Appendix

Universal Concept Class

- Problem: each $x \in X$ defined by n boolean features. Let C be the set of all subsets of X.
- Question: is C PAC-learnable?
- Sample complexity: H must contain C. Thus,

$$|H| \geq |C| = 2^{(2^n)}.$$

The bound gives $m = \frac{1}{\epsilon}((\log 2)\, 2^n + \log \frac{1}{\delta})$.

- It can be proved that C is not PAC-learnable, it requires an exponential sample size.

k-Term DNF Formulae

- Definition: expressions of the form $T_1 \vee \cdots \vee T_k$ with each term T_i conjunctions of boolean literals with at most n variables.

- Problem: learning *k*-term DNF formulae.

- Sample complexity: $|H| = |C| = 3^{nk}$. Thus, polynomial sample complexity $\frac{1}{\epsilon}\big((\log 3)\, nk + \log \frac{1}{\delta}\big)$.

- Time complexity: intractable if $RP \neq NP$: the class is then not efficiently PAC-learnable (proof by reduction from graph 3-coloring). But, a strictly larger class is!

k-CNF Expressions

- Definition: expressions $T_1 \wedge \cdots \wedge T_j$ of arbitrary length j with each term T_i a disjunction of at most k boolean attributes.

- Algorithm: reduce problem to that of learning conjunctions of boolean literals. $(2n)^k$ new variables:

$$(u_1, \ldots, u_k) \rightarrow Y_{u_1, \ldots, u_k}.$$

  - the transformation is a bijection;
  - effect of the transformation on the distribution is not an issue: PAC-learning allows any distribution D.

k-Term DNF Terms and *k*-CNF Expressions

- Observation: any *k*-term DNF formula can be written as a *k*-CNF expression. By associativity,

$$\bigvee_{i=1}^{k} u_{i,1} \wedge \cdots \wedge u_{i,n_i} = \bigwedge_{j_1 \in [1,n_1],\ldots,j_k \in [1,n_k]} u_{1,j_1} \vee \cdots \vee u_{k,j_k}.$$

  - Example: $(u_1 \wedge u_2 \wedge u_3) \vee (v_1 \wedge v_2 \wedge v_3) = \bigwedge_{i,j=1}^{3} (u_i \vee v_j)$.
  - But, in general converting a *k*-CNF (equiv. to a *k*-term DNF) to a *k*-term DNF is intractable.
- Key aspects of PAC-learning definition:
  - cost of representation of concept c.
  - choice of hypothesis set H.

Foundations of Machine Learning

Learning with Infinite Hypothesis Sets

Mehryar Mohri
Courant Institute and Google Research
mohri@cims.nyu.edu

Motivation

- With an infinite hypothesis set H, the error bounds of the previous lecture are not informative.
- Is efficient learning from a finite sample possible when H is infinite?
- Our example of axis-aligned rectangles shows that it is possible.
- Can we reduce the infinite case to a finite set? Project over finite samples?
- Are there useful measures of complexity for infinite hypothesis sets?

This lecture

- Rademacher complexity
- Growth Function
- VC dimension
- Lower bound

Empirical Rademacher Complexity

- Definition:
  - G family of functions mapping from set Z to $[a, b]$.
  - sample $S = (z_1, \ldots, z_m)$.
  - σ_is (Rademacher variables): independent uniform random variables taking values in $\{-1, +1\}$.

$$\widehat{\mathfrak{R}}_S(G) = \operatorname*{E}_{\boldsymbol{\sigma}}\left[\sup_{g \in G} \underbrace{\frac{1}{m} \begin{bmatrix} \sigma_1 \\ \vdots \\ \sigma_m \end{bmatrix} \cdot \begin{bmatrix} g(z_1) \\ \vdots \\ g(z_m) \end{bmatrix}}_{\text{correlation with random noise}}\right] = \operatorname*{E}_{\boldsymbol{\sigma}}\left[\sup_{g \in G} \frac{1}{m} \sum_{i=1}^{m} \sigma_i g(z_i)\right].$$

Rademacher Complexity

- Definitions: let G be a family of functions mapping from Z to $[a, b]$.
  - Empirical Rademacher complexity of G:

$$\widehat{\mathfrak{R}}_S(G) = \operatorname*{E}_{\sigma}\left[\sup_{g \in G} \frac{1}{m} \sum_{i=1}^{m} \sigma_i g(z_i)\right],$$

where σ_is are independent uniform random variables taking values in $\{-1, +1\}$ and $S = (z_1, \ldots, z_m)$.

  - Rademacher complexity of G:

$$\mathfrak{R}_m(G) = \operatorname*{E}_{S \sim D^m}[\widehat{\mathfrak{R}}_S(G)].$$

Rademacher Complexity Bound

(Koltchinskii and Panchenko, 2002)

- **Theorem**: Let G be a family of functions mapping from Z to $[0, 1]$. Then, for any $\delta > 0$, with probability at least $1-\delta$, the following holds for all $g \in G$:

$$\mathrm{E}[g(z)] \leq \frac{1}{m}\sum_{i=1}^{m} g(z_i) + 2\mathfrak{R}_m(G) + \sqrt{\frac{\log\frac{1}{\delta}}{2m}}.$$

$$\mathrm{E}[g(z)] \leq \frac{1}{m}\sum_{i=1}^{m} g(z_i) + 2\widehat{\mathfrak{R}}_S(G) + 3\sqrt{\frac{\log\frac{2}{\delta}}{2m}}.$$

- **Proof**: Apply McDiarmid's inequality to

$$\Phi(S) = \sup_{g \in G} \mathrm{E}[g] - \widehat{\mathrm{E}}_S[g].$$

- Changing one point of S changes $\Phi(S)$ by at most $\frac{1}{m}$.

$$\begin{aligned}\Phi(S') - \Phi(S) &= \sup_{g \in G}\{\mathrm{E}[g] - \widehat{\mathrm{E}}_{S'}[g]\} - \sup_{g \in G}\{\mathrm{E}[g] - \widehat{\mathrm{E}}_S[g]\} \\ &\leq \sup_{g \in G}\{\{\mathrm{E}[g] - \widehat{\mathrm{E}}_{S'}[g]\} - \{\mathrm{E}[g] - \widehat{\mathrm{E}}_S[g]\}\} \\ &= \sup_{g \in G}\{\widehat{\mathrm{E}}_S[g] - \widehat{\mathrm{E}}_{S'}[g]\} = \sup_{g \in G} \frac{1}{m}(g(z_m) - g(z'_m)) \leq \frac{1}{m}.\end{aligned}$$

- Thus, by McDiarmid's inequality, with probability at least $1 - \frac{\delta}{2}$

$$\Phi(S) \leq \mathop{\mathrm{E}}_S[\Phi(S)] + \sqrt{\frac{\log \frac{2}{\delta}}{2m}}.$$

- We are left with bounding the expectation.

- Series of observations:

$$\begin{aligned}
\mathop{\mathrm{E}}_{S}[\Phi(S)] &= \mathop{\mathrm{E}}_{S}\Big[\sup_{g\in G} \mathrm{E}[g] - \widehat{\mathrm{E}}_S(g)\Big] \\
&= \mathop{\mathrm{E}}_{S}\Big[\sup_{g\in G} \mathop{\mathrm{E}}_{S'}[\widehat{\mathrm{E}}_{S'}(g) - \widehat{\mathrm{E}}_S(g)]\Big] \\
\text{(sub-add. of sup)} &\leq \mathop{\mathrm{E}}_{S,S'}\Big[\sup_{g\in G} \widehat{\mathrm{E}}_{S'}(g) - \widehat{\mathrm{E}}_S(g)\Big] \\
&= \mathop{\mathrm{E}}_{S,S'}\Big[\sup_{g\in G} \frac{1}{m}\sum_{i=1}^{m}(g(z'_i) - g(z_i))\Big] \\
\text{(swap } z_i \text{ and } z'_i\text{)} &= \mathop{\mathrm{E}}_{\sigma,S,S'}\Big[\sup_{g\in G} \frac{1}{m}\sum_{i=1}^{m}\sigma_i(g(z'_i) - g(z_i))\Big] \\
\text{(sub-additiv. of sup)} &\leq \mathop{\mathrm{E}}_{\sigma,S'}\Big[\sup_{g\in G} \frac{1}{m}\sum_{i=1}^{m}\sigma_i g(z'_i)\Big] + \mathop{\mathrm{E}}_{\sigma,S}\Big[\sup_{g\in G} \frac{1}{m}\sum_{i=1}^{m}-\sigma_i g(z_i)\Big] \\
&= 2\mathop{\mathrm{E}}_{\sigma,S}\Big[\sup_{g\in G} \frac{1}{m}\sum_{i=1}^{m}\sigma_i g(z_i)\Big] = 2\mathfrak{R}_m(G).
\end{aligned}$$

- Now, changing one point of S makes $\widehat{\mathfrak{R}}_S(G)$ vary by at most $\frac{1}{m}$. Thus, again by McDiarmid's inequality, with probability at least $1-\frac{\delta}{2}$,

$$\mathfrak{R}_m(G) \leq \widehat{\mathfrak{R}}_{\mathcal{S}}(G) + \sqrt{\frac{\log\frac{2}{\delta}}{2m}}.$$

- Thus, by the union bound, with probability at least $1-\delta$,

$$\Phi(S) \leq 2\widehat{\mathfrak{R}}_S(G) + 3\sqrt{\frac{\log\frac{2}{\delta}}{2m}}.$$

Loss Functions - Hypothesis Set

- Proposition: Let H be a family of functions taking values in $\{-1, +1\}$, G the family of zero-one loss functions of H: $G = \big\{(x, y) \mapsto 1_{h(x) \neq y} \colon h \in H\big\}$. Then,

$$\mathfrak{R}_m(G) = \frac{1}{2}\mathfrak{R}_m(H).$$

- Proof:

$$\begin{aligned}
\mathfrak{R}_m(G) &= \mathop{\mathrm{E}}_{S,\sigma}\Big[\sup_{h \in H} \frac{1}{m}\sum_{i=1}^m \sigma_i 1_{h(x_i) \neq y_i}\Big] \\
&= \mathop{\mathrm{E}}_{S,\sigma}\Big[\sup_{h \in H} \frac{1}{m}\sum_{i=1}^m \sigma_i \frac{1}{2}(1 - y_i h(x_i))\Big] \\
&= \underbrace{\frac{1}{2}\mathop{\mathrm{E}}_{S,\sigma}\Big[\sup_{h \in H} \frac{1}{m}\sum_{i=1}^m \sigma_i\Big]}_{=0} + \frac{1}{2}\mathop{\mathrm{E}}_{S,\sigma}\Big[\sup_{h \in H} \frac{1}{m}\sum_{i=1}^m -\sigma_i y_i h(x_i)\Big] \\
&= \frac{1}{2}\mathop{\mathrm{E}}_{S,\sigma}\Big[\sup_{h \in H} \frac{1}{m}\sum_{i=1}^m \sigma_i h(x_i)\Big].
\end{aligned}$$

Generalization Bounds - Rademacher

- Corollary: Let H be a family of functions taking values in $\{-1, +1\}$. Then, for any $\delta > 0$, with probability at least $1 - \delta$, for any $h \in H$,

$$R(h) \leq \widehat{R}(h) + \mathfrak{R}_m(H) + \sqrt{\frac{\log \frac{1}{\delta}}{2m}}.$$

$$R(h) \leq \widehat{R}(h) + \widehat{\mathfrak{R}}_S(H) + 3\sqrt{\frac{\log \frac{2}{\delta}}{2m}}.$$

Remarks

- First bound distribution-dependent, second data-dependent bound, which makes them attractive.
- But, how do we compute the empirical Rademacher complexity?
- Computing $\mathrm{E}_\sigma[\sup_{h\in H} \frac{1}{m}\sum_{i=1}^m \sigma_i h(x_i)]$ requires solving ERM problems, typically computationally hard.
- Relation with combinatorial measures easier to compute?

This lecture

- Rademacher complexity
- Growth Function
- VC dimension
- Lower bound

Growth Function

- Definition: the growth function $\Pi_H \colon \mathbb{N} \to \mathbb{N}$ for a hypothesis set H is defined by

$$\forall m \in \mathbb{N},\ \Pi_H(m) = \max_{\{x_1,\ldots,x_m\} \subseteq X} \Big| \big\{ \big(h(x_1), \ldots, h(x_m)\big) \colon h \in H \big\} \Big|.$$

- Thus, $\Pi_H(m)$ is the maximum number of ways m points can be classified using H.

Massart's Lemma

(Massart, 2000)

- **Theorem**: Let $A \subseteq \mathbb{R}^m$ be a finite set, with $R = \max_{x \in A} \|x\|_2$, then, the following holds:

$$\mathop{\mathrm{E}}_{\sigma}\left[\frac{1}{m}\sup_{x\in A}\sum_{i=1}^{m}\sigma_i x_i\right] \leq \frac{R\sqrt{2\log|A|}}{m}.$$

- **Proof**:

$$\begin{aligned}
\exp\left(t \mathop{\mathrm{E}}_{\sigma}\left[\sup_{x\in A}\sum_{i=1}^{m}\sigma_i x_i\right]\right) &\leq \mathop{\mathrm{E}}_{\sigma}\left(\exp\left[t\sup_{x\in A}\sum_{i=1}^{m}\sigma_i x_i\right]\right) \quad \text{(Jensen's ineq.)}\\
&= \mathop{\mathrm{E}}_{\sigma}\left(\sup_{x\in A}\exp\left[t\sum_{i=1}^{m}\sigma_i x_i\right]\right)\\
&\leq \sum_{x\in A}\mathop{\mathrm{E}}_{\sigma}\left(\exp\left[t\sum_{i=1}^{m}\sigma_i x_i\right]\right) = \sum_{x\in A}\prod_{i=1}^{m}\mathop{\mathrm{E}}_{\sigma}\left(\exp\left[t\sigma_i x_i\right]\right)\\
\text{(Hoeffding's ineq.)} &\leq \sum_{x\in A}\left(\exp\left[\frac{\sum_{i=1}^{m} t^2(2|x_i|)^2}{8}\right]\right) \leq |A|e^{\frac{t^2R^2}{2}}.
\end{aligned}$$

- Taking the log yields:

$$\mathop{\mathrm{E}}_{\sigma}\left[\sup_{x \in A} \sum_{i=1}^{m} \sigma_i x_i\right] \leq \frac{\log |A|}{t} + \frac{tR^2}{2}.$$

- Minimizing the bound by choosing $t = \frac{\sqrt{2 \log |A|}}{R}$ gives

$$\mathop{\mathrm{E}}_{\sigma}\left[\sup_{x \in A} \sum_{i=1}^{m} \sigma_i x_i\right] \leq R\sqrt{2 \log |A|}.$$

Growth Function Bound on Rad. Complexity

- **Corollary**: Let G be a family of functions taking values in $\{-1, +1\}$, then the following holds:

$$\mathfrak{R}_m(G) \leq \sqrt{\frac{2 \log \Pi_G(m)}{m}}.$$

- **Proof**:

$$\begin{aligned}
\widehat{\mathfrak{R}}_S(G) &= \mathop{\mathrm{E}}_{\sigma}\left[\sup_{g \in G} \frac{1}{m} \begin{bmatrix} \sigma_1 \\ \vdots \\ \sigma_m \end{bmatrix} \cdot \begin{bmatrix} g(z_1) \\ \vdots \\ g(z_m) \end{bmatrix}\right] \\
&\leq \frac{\sqrt{m}\sqrt{2 \log |\{(g(z_1), \ldots, g(z_m)) \colon g \in G\}|}}{m} \qquad \text{(Massart's Lemma)} \\
&\leq \frac{\sqrt{m}\sqrt{2 \log \Pi_G(m)}}{m} = \sqrt{\frac{2 \log \Pi_G(m)}{m}}.
\end{aligned}$$

Generalization Bound - Growth Function

- Corollary: Let H be a family of functions taking values in $\{-1,+1\}$. Then, for any $\delta>0$, with probability at least $1-\delta$, for any $h \in H$,

$$R(h) \leq \widehat{R}(h) + \sqrt{\frac{2\log \Pi_H(m)}{m}} + \sqrt{\frac{\log \frac{1}{\delta}}{2m}}.$$

- But, how do we compute the growth function? Relationship with the VC-dimension (Vapnik-Chervonenkis dimension).

This lecture

- Rademacher complexity
- Growth Function
- VC dimension
- Lower bound

VC Dimension

(Vapnik & Chervonenkis, 1968-1971; Vapnik, 1982, 1995, 1998)

- Definition: the VC-dimension of a hypothesis set H is defined by

$$\mathrm{VCdim}(H) = \max\{m \colon \Pi_H(m) = 2^m\}.$$

- Thus, the VC-dimension is the size of the largest set that can be fully shattered by H.

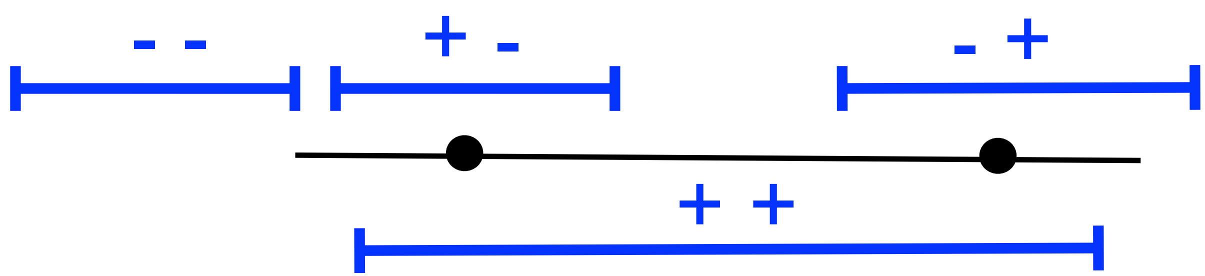
- Purely combinatorial notion.

Examples

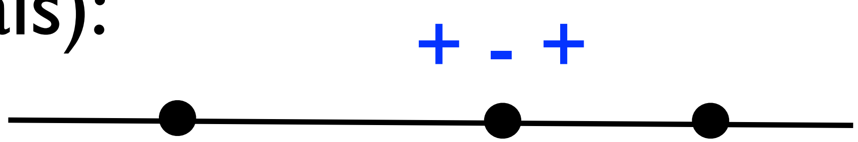
- In the following, we determine the VC dimension for several hypothesis sets.
- To give a lower bound d for $\mathrm{VCdim}(H)$, it suffices to show that a set S of cardinality d can be shattered by H.
- To give an upper bound, we need to prove that no set S of cardinality $d+1$ can be shattered by H, which is typically more difficult.

Intervals of The Real Line

- **Observations:**
  - Any set of two points can be shattered by four intervals



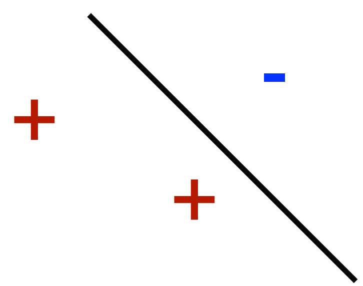
  - No set of three points can be shattered since the following dichotomy "+ - +" is not realizable (by definition of intervals):



  - Thus, $\mathrm{VCdim}(\text{intervals in } \mathbb{R}) = 2$.

Hyperplanes

- Observations:
  - Any three non-collinear points can be shattered:

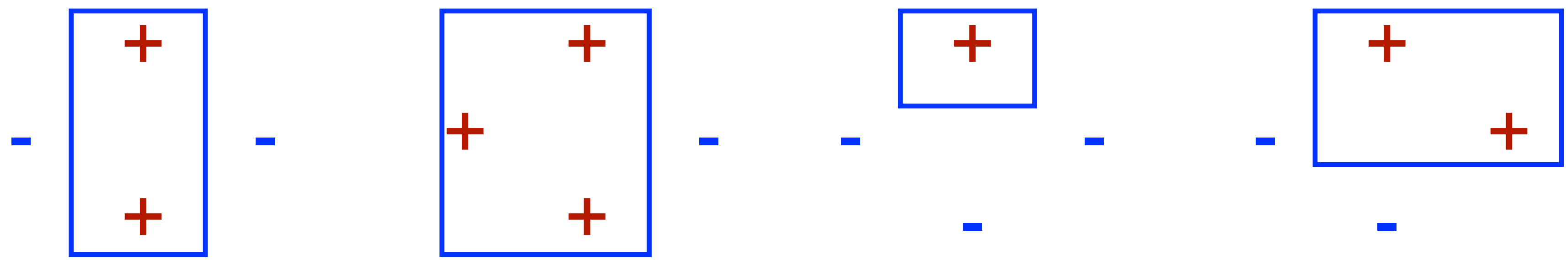


  - Unrealizable dichotomies for four points:

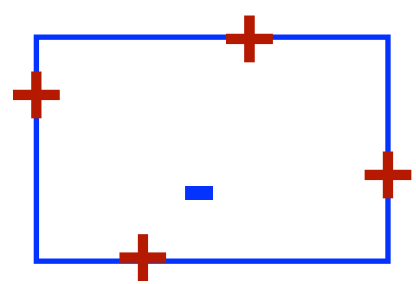
  - Thus, $\mathrm{VCdim}(\text{hyperplanes in } \mathbb{R}^d) = d+1$.

Axis-Aligned Rectangles in the Plane

- **Observations:**
  - The following four points can be shattered:



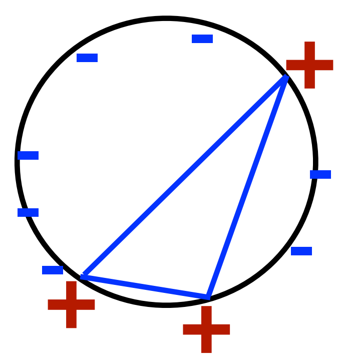
  - No set of five points can be shattered: label negatively the point that is not near the sides.



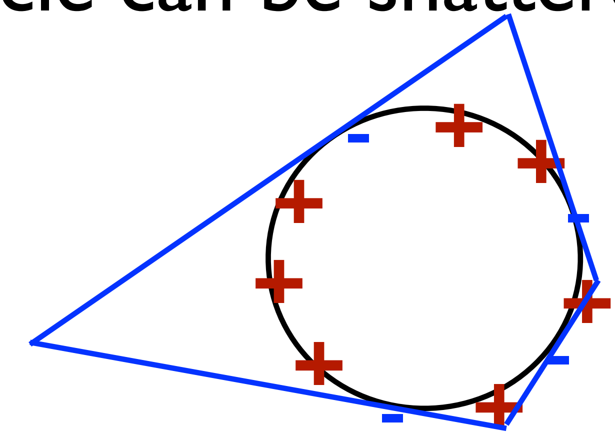
  - Thus, $\text{VCdim}(\text{axis-aligned rectangles}) = 4$.

Convex Polygons in the Plane

- Observations:
  - $2d+1$ points on a circle can be shattered by a *d*-gon:



|positive points| < |negative points|

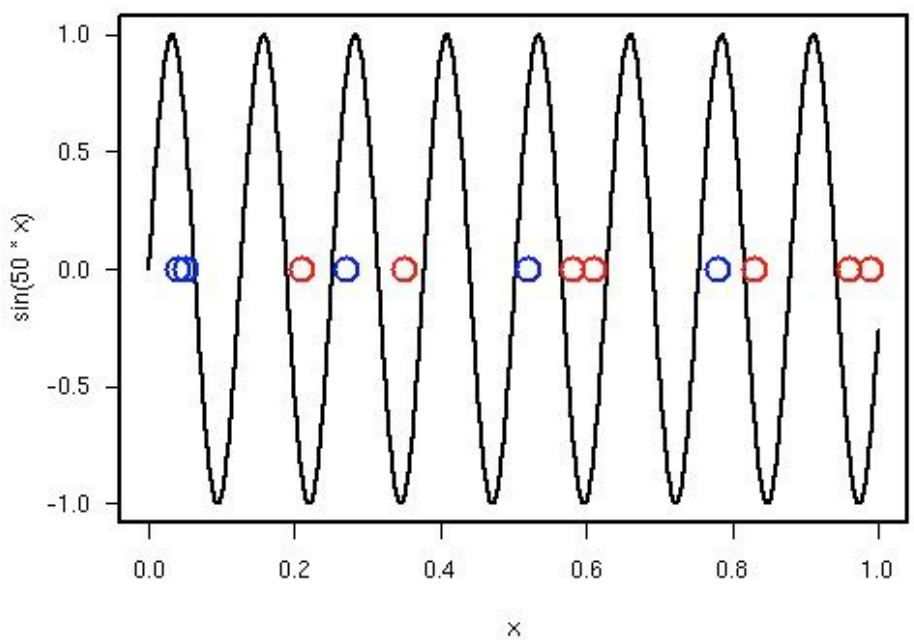


|positive points| > |negative points|

  - It can be shown that choosing the points on the circle maximizes the number of possible dichotomies. Thus, $\mathrm{VCdim}(\text{convex } d\text{-gons}) = 2d+1$. Also, $\mathrm{VCdim}(\text{convex polygons}) = +\infty$.

Sine Functions

- Observations:
  - Any finite set of points on the real line can be shattered by $\{t \mapsto \sin(\omega t) \colon \omega \in \mathbb{R}\}$.
  - Thus, $\text{VCdim}(\text{sine functions}) = +\infty$.



Sauer's Lemma

(Vapnik & Chervonenkis, 1968-1971; Sauer, 1972)

- **Theorem**: let H be a hypothesis set with $\mathrm{VCdim}(H) = d$ then, for all $m \in \mathbb{N}$,

$$\Pi_H(m) \leq \sum_{i=0}^{d} \binom{m}{i}.$$

- **Proof**: the proof is by induction on $m+d$. The statement clearly holds for $m=1$ and $d=0$ or $d=1$. Assume that it holds for $(m-1, d-1)$ and $(m-1, d)$.
  - Fix a set $S = \{x_1, \ldots, x_m\}$ with $\Pi_H(m)$ dichotomies and let $G = H_{|S}$ be the set of concepts H induces by restriction to S.

- Consider the following families over $S' = \{x_1, \ldots, x_{m-1}\}$:

$$G_1 = G_{|S'} \quad G_2 = \{g' \subseteq S' \colon (g' \in G) \wedge (g' \cup \{x_m\} \in G)\}.$$

x_1	x_2	$\cdots$	x_{m-1}	x_m
1	1	0	1	0
1	1	0	1	1
0	1	1	1	1
1	0	0	1	0
1	0	0	0	1
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$

- Observe that $|G_1| + |G_2| = |G|$.

- Since $\mathrm{VCdim}(G_1) \leq d$, by the induction hypothesis,

$$|G_1| \leq \Pi_{G_1}(m-1) \leq \sum_{i=0}^{d} \binom{m-1}{i}.$$

- By definition of G_2, if a set $Z \subseteq S'$ is shattered by G_2, then the set $Z \cup \{x_m\}$ is shattered by G. Thus,

$$\mathrm{VCdim}(G_2) \leq \mathrm{VCdim}(G) - 1 = d - 1$$

and by the induction hypothesis,

$$|G_2| \leq \Pi_{G_2}(m-1) \leq \sum_{i=0}^{d-1} \binom{m-1}{i}.$$

- Thus, $|G| \leq \sum_{i=0}^{d} \binom{m-1}{i} + \sum_{i=0}^{d-1} \binom{m-1}{i}$
$= \sum_{i=0}^{d} \binom{m-1}{i} + \binom{m-1}{i-1} = \sum_{i=0}^{d} \binom{m}{i}$.

Sauer's Lemma - Consequence

- **Corollary**: let H be a hypothesis set with $\mathrm{VCdim}(H) = d$ then, for all $m \geq d$,

$$\Pi_H(m) \leq \left(\frac{em}{d}\right)^d = O(m^d).$$

- **Proof**:

$$\begin{aligned}
\sum_{i=0}^{d} \binom{m}{i} &\leq \sum_{i=0}^{d} \binom{m}{i} \left(\frac{m}{d}\right)^{d-i} \\
&\leq \sum_{i=0}^{m} \binom{m}{i} \left(\frac{m}{d}\right)^{d-i} \\
&= \left(\frac{m}{d}\right)^{d} \sum_{i=0}^{m} \binom{m}{i} \left(\frac{d}{m}\right)^{i} \\
&= \left(\frac{m}{d}\right)^{d} \left(1 + \frac{d}{m}\right)^{m} \leq \left(\frac{m}{d}\right)^{d} e^{d}.
\end{aligned}$$

Remarks

- Remarkable property of growth function:
  - either $\mathrm{VCdim}(H) = d < +\infty$ and $\Pi_H(m) = O(m^d)$
  - or $\mathrm{VCdim}(H) = +\infty$ and $\Pi_H(m) = 2^m$.

Generalization Bound - VC Dimension

- **Corollary**: Let H be a family of functions taking values in $\{-1,+1\}$ with VC dimension d. Then, for any $\delta > 0$, with probability at least $1-\delta$, for any $h \in H$,

$$R(h) \leq \widehat{R}(h) + \sqrt{\frac{2d \log \frac{em}{d}}{m}} + \sqrt{\frac{\log \frac{1}{\delta}}{2m}}.$$

- **Proof**: Corollary combined with Sauer's lemma.
- **Note**: The general form of the result is

$$R(h) \leq \widehat{R}(h) + O\left(\sqrt{\frac{\log(m/d)}{(m/d)}}\right).$$

Comparison - Standard VC Bound

(Vapnik & Chervonenkis, 1971; Vapnik, 1982)

- **Theorem**: Let H be a family of functions taking values in $\{-1, +1\}$ with VC dimension d. Then, for any $\delta > 0$, with probability at least $1 - \delta$, for any $h \in H$,

$$R(h) \leq \widehat{R}(h) + \sqrt{\frac{8d \log \frac{2em}{d} + 8 \log \frac{4}{\delta}}{m}}.$$

- **Proof**: Derived from growth function bound

$$\Pr\left[\left|R(h) - \widehat{R}(h)\right| > \epsilon\right] \leq 4\Pi_H(2m) \exp\left(-\frac{m\epsilon^2}{8}\right).$$

This lecture

- Rademacher complexity
- Growth Function
- VC dimension
- Lower bound

VCDim Lower Bound - Realizable Case

(Ehrenfeucht et al., 1988)

- Theorem: let H be a hypothesis set with VC-dimension $d > 1$. Then, for any learning algorithm L,

$$\exists D, \exists f \in H, \ \Pr_{S \sim D^m}\left[R_D(h_S, f) > \frac{d-1}{32m}\right] \geq 1/100.$$

- Proof: choose D such that L can do no better than tossing a coin for some points.
    - Let $X = \{x_0, x_1, \ldots, x_{d-1}\}$ be a set fully shattered. For any $\epsilon > 0$, define D with support X by

$$\Pr_D[x_0] = 1 - 8\epsilon \quad \text{and} \quad \forall i \in [1, d-1], \Pr_D[x_i] = \frac{8\epsilon}{d-1}.$$

- We can assume without loss of generality that L makes no error on x_0.
- For a sample S, let $\overline{S}$ denote the set of its elements falling in $X_1 = \{x_1, \ldots, x_{d-1}\}$ and let $\mathcal{S}$ be the set of samples of size m with at most $(d-1)/2$ points in X_1.
- Fix a sample $S \in \mathcal{S}$. Using $|X - \overline{S}| \geq (d-1)/2$,

$$
\begin{aligned}
\mathop{\mathrm{E}}_{f \sim U}[R_D(h_S, f)] &= \sum_f \sum_{x \in X} 1_{h(x) \neq f(x)} \Pr[x] \Pr[f] \\
&\geq \sum_f \sum_{x \notin \overline{S}} 1_{h(x) \neq f(x)} \Pr[x] \Pr[f] \\
&= \sum_{x \notin \overline{S}} \Big(\sum_f 1_{h(x) \neq f(x)} \Pr[f] \Big) \Pr[x] \\
&= \frac{1}{2} \sum_{x \notin \overline{S}} \Pr[x] \geq \frac{1}{2} \frac{d-1}{2} \frac{8\epsilon}{d-1} = 2\epsilon.
\end{aligned}
$$

- Since the inequality holds for all $S \in \mathcal{S}$, it also holds in expectation: $\mathrm{E}_{S, f \sim U}[R_D(h_S, f)] \geq 2\epsilon$. This implies that there exists a labeling f_0 such that $\mathrm{E}_S[R_D(h_S, f_0)] \geq 2\epsilon$.
- Since $\Pr_D[X - \{x_0\}] \leq 8\epsilon$, we also have $R_D(h_S, f_0) \leq 8\epsilon$. Thus,

$$2\epsilon \leq \mathop{\mathrm{E}}_{S}[R_D(h_S, f_0)] \leq 8\epsilon \mathop{\Pr}_{S \in \mathcal{S}}[R_D(h_S, f_0) \geq \epsilon] + (1 - \mathop{\Pr}_{S \in \mathcal{S}}[R_D(h_S, f_0) \geq \epsilon])\epsilon.$$

- Collecting terms in $\mathop{\Pr}_{S \in \mathcal{S}}[R_D(h_S, f_0) \geq \epsilon]$, we obtain:

$$\mathop{\Pr}_{S \in \mathcal{S}}[R_D(h_S, f_0) \geq \epsilon] \geq \frac{1}{7\epsilon}(2\epsilon - \epsilon) = \frac{1}{7}.$$

- Thus, the probability over all samples S (not necessarily in $\mathcal{S}$) can be lower bounded as

$$\mathop{\Pr}_{S}[R_D(h_S, f_0) \geq \epsilon] \geq \mathop{\Pr}_{S \in \mathcal{S}}[R_D(h_S, f_0) \geq \epsilon] \Pr[\mathcal{S}] \geq \frac{1}{7} \Pr[\mathcal{S}].$$

- This leads us to seeking a lower bound for $\Pr[\mathcal{S}]$. The probability that more than $(d-1)/2$ points be drawn in a sample of size m verifies the Chernoff bound for any $\gamma > 0$:

$$1 - \Pr[\mathcal{S}] = \Pr[S_m \geq 8\epsilon m(1+\gamma)] \leq e^{-8\epsilon m \frac{\gamma^2}{3}}.$$

- Thus, for $\epsilon = (d-1)/(32m)$ and $\gamma = 1$,

$$\Pr[S_m \geq \tfrac{d-1}{2}] \leq e^{-(d-1)/12} \leq e^{-1/12} \leq 1 - 7\delta,$$

for $\delta \leq .01$. Thus, $\Pr[\mathcal{S}] \geq 7\delta$ and

$$\Pr_S[R_D(h_S, f_0) \geq \epsilon] \geq \delta.$$

Agnostic PAC Model

- Definition: concept class C is PAC-learnable if there exists a learning algorithm L such that:
  - for all $c \in C, \epsilon > 0, \delta > 0$, and all distributions D,

$$\Pr_{S \sim D}\left[R(h_S) - \inf_{h \in H} R(h) \leq \epsilon\right] \geq 1 - \delta,$$

  - for samples S of size $m = poly(1/\epsilon, 1/\delta)$ for a fixed polynomial.

VCDim Lower Bound - Non-Realizable Case

(Anthony and Bartlett, 1999)

- Theorem: let H be a hypothesis set with VC dimension $d > 1$. Then, for any learning algorithm L,

$\exists D$ over $X \times \{0, 1\}$,

$$\Pr_{S \sim D^m}\left[R_D(h_S) - \inf_{h \in H} R_D(h) > \sqrt{\frac{d}{320m}}\right] \geq 1/64.$$

- Equivalently, for any learning algorithm, the sample complexity verifies

$$m \geq \frac{d}{320\epsilon^2}.$$

References

- Martin Anthony, Peter L. Bartlett. *Neural network learning: theoretical foundations*. Cambridge University Press. 1999.

- Anselm Blumer, A. Ehrenfeucht, David Haussler, and Manfred K. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *Journal of the ACM (JACM)*, Volume 36, Issue 4, 1989.

- A. Ehrenfeucht, David Haussler, Michael Kearns, Leslie Valiant. A general lower bound on the number of examples needed for learning. Proceedings of *1st COLT*. pp. 139-154, 1988.

- Koltchinskii, Vladimir and Panchenko, Dmitry. Empirical margin distributions and bounding the generalization error of combined classifiers. The Annals of Statistics, 30(1), 2002.

- Pascal Massart. Some applications of concentration inequalities to statistics. Annales de la Faculte des Sciences de Toulouse, IX:245–303, 2000.

- N. Sauer. On the density of families of sets. *Journal of Combinatorial Theory* (A), 13:145-147, 1972.

References

- Vladimir N. Vapnik. *Estimation of Dependences Based on Empirical Data*. Springer, 1982.
- Vladimir N. Vapnik. *The Nature of Statistical Learning Theory.* Springer, 1995.
- Vladimir N. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, New York, 1998.
- Vladimir N. Vapnik and Alexey Chervonenkis. *Theory of Pattern Recognition*. Nauka, Moscow (in Russian). 1974.
- Vladimir N. Vapnik and Alexey Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Prob. and its Appl.*, vol. 16, no. 2, pp. 264-280, 1971.

Foundations of Machine Learning
Support Vector Machines

Mehryar Mohri
Courant Institute and Google Research
mohri@cims.nyu.edu

Binary Classification Problem

- Training data: sample drawn i.i.d. from set $X \subseteq \mathbb{R}^N$ according to some distribution D,

$$S = ((x_1, y_1), \ldots, (x_m, y_m)) \in X \times \{-1, +1\}.$$

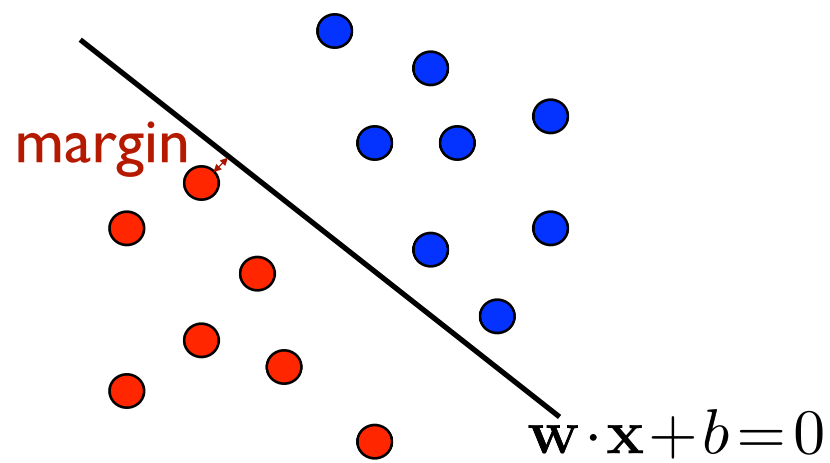
- Problem: find hypothesis $h : X \mapsto \{-1, +1\}$ in H (classifier) with small generalization error $R(h)$.
    - choice of hypothesis set H : learning guarantees of previous lecture.

      → linear classification (hyperplanes) if dimension N is not too large.

This Lecture

- Support Vector Machines - separable case
- Support Vector Machines - non-separable case
- Margin guarantees

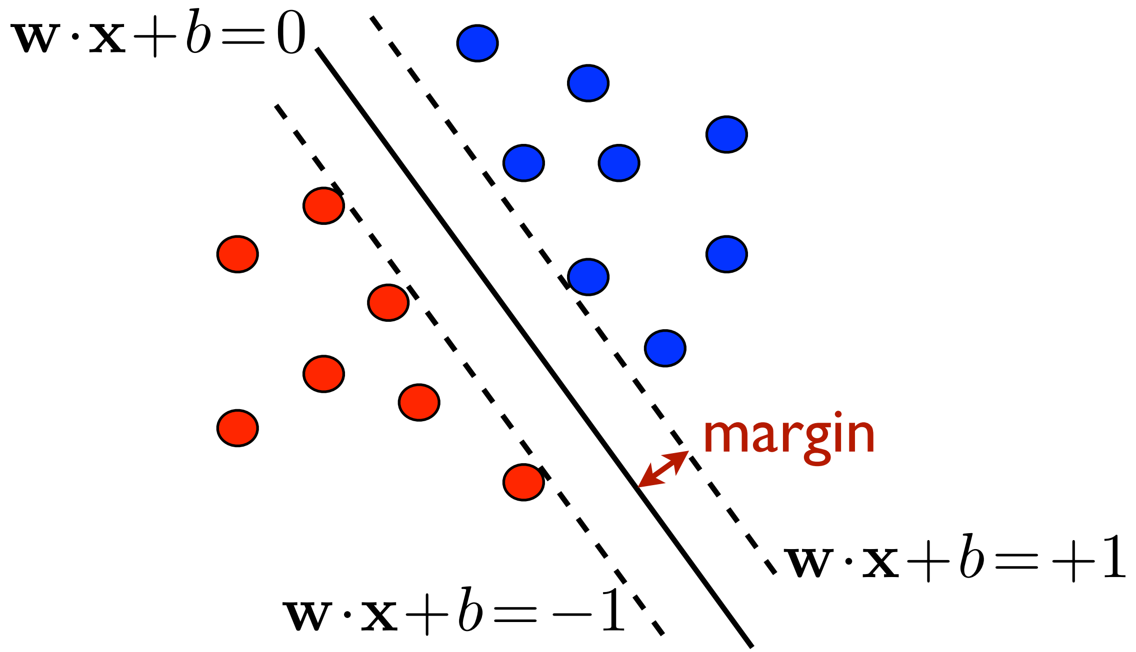
Linear Separation



- classifiers: $H=\{\mathbf{x}\mapsto \text{sgn}(\mathbf{w}\cdot\mathbf{x}+b) : \mathbf{w}\in\mathbb{R}^N, b\in\mathbb{R}\}$.
- geometric margin: $\rho = \min_{i\in[1,m]} \frac{|\mathbf{w}\cdot\mathbf{x}_i+b|}{\|\mathbf{w}\|}$.
- which separating hyperplane?

Optimal Hyperplane: Max. Margin

(Vapnik and Chervonenkis, 1965)



$$\rho = \max_{\mathbf{w},b\,:\,y_i(\mathbf{w}\cdot\mathbf{x}_i+b)\geq 0} \; \min_{i\in[1,m]} \frac{|\mathbf{w}\cdot\mathbf{x}_i + b|}{\|\mathbf{w}\|}.$$

Maximum Margin

$$
\begin{aligned}
\rho &= \max_{\mathbf{w},b\colon\, y_i(\mathbf{w}\cdot\mathbf{x}_i+b)\geq 0} \ \min_{i\in[1,m]} \frac{|\mathbf{w}\cdot\mathbf{x}_i+b|}{\|\mathbf{w}\|} \\
&= \max_{\substack{\mathbf{w},b\colon\, y_i(\mathbf{w}\cdot\mathbf{x}_i+b)\geq 0\\ \min_{i\in[1,m]} |\mathbf{w}\cdot\mathbf{x}_i+b|=1}} \ \min_{i\in[1,m]} \frac{|\mathbf{w}\cdot\mathbf{x}_i+b|}{\|\mathbf{w}\|} \qquad \text{(scale-invariance)} \\
&= \max_{\substack{\mathbf{w},b\colon\, y_i(\mathbf{w}\cdot\mathbf{x}_i+b)\geq 0\\ \min_{i\in[1,m]} |\mathbf{w}\cdot\mathbf{x}_i+b|=1}} \frac{1}{\|\mathbf{w}\|} \\
&= \max_{\mathbf{w},b\colon\, y_i(\mathbf{w}\cdot\mathbf{x}_i+b)\geq 1} \frac{1}{\|\mathbf{w}\|}. \qquad \text{(min. reached)}
\end{aligned}
$$

Optimization Problem

- Constrained optimization:

$$\min_{\mathbf{w},b} \ \frac{1}{2}\|\mathbf{w}\|^2$$
$$\text{subject to } \ y_i(\mathbf{w}\cdot\mathbf{x}_i + b) \geq 1, i \in [1, m].$$

- Properties:
  - Convex optimization.
  - Unique solution for linearly separable sample.

Optimal Hyperplane Equations

- Lagrangian: for all $\mathbf{w}, b, \alpha_i \geq 0$,

$$L(\mathbf{w}, b, \alpha) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{m} \alpha_i [y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1].$$

- KKT conditions:

$$\nabla_{\mathbf{w}} L = \mathbf{w} - \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i = 0 \iff \boxed{\mathbf{w} = \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i .}$$

$$\nabla_b L = -\sum_{i=1}^{m} \alpha_i y_i = 0 \iff \boxed{\sum_{i=1}^{m} \alpha_i y_i = 0.}$$

$$\boxed{\forall i \in [1, m],\ \alpha_i [y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1] = 0.}$$

Support Vectors

- Complementarity conditions:

$$\alpha_i[y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1] = 0 \implies \alpha_i = 0 \vee y_i(\mathbf{w} \cdot \mathbf{x}_i + b) = 1.$$

- Support vectors: vectors $\mathbf{x}_i$ such that

$$\alpha_i \neq 0 \wedge y_i(\mathbf{w} \cdot \mathbf{x}_i + b) = 1.$$

  - Note: support vectors are not unique.

Moving to The Dual

- Plugging in the expression of $\mathbf{w}$ in L gives:

$$L = \underbrace{\frac{1}{2}\left\|\sum_{i=1}^{m}\alpha_i y_i \mathbf{x}_i\right\|^2 - \sum_{i,j=1}^{m}\alpha_i\alpha_j y_i y_j(\mathbf{x}_i\cdot\mathbf{x}_j)}_{-\frac{1}{2}\sum_{i,j=1}^{m}\alpha_i\alpha_j y_i y_j(\mathbf{x}_i\cdot\mathbf{x}_j)} - \underbrace{\sum_{i=1}^{m}\alpha_i y_i b}_{0} + \sum_{i=1}^{m}\alpha_i.$$

- Thus,

$$L = \sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{m}\alpha_i\alpha_j y_i y_j(\mathbf{x}_i\cdot\mathbf{x}_j).$$

Equivalent Dual Opt. Problem

- Constrained optimization:

$$\max_{\alpha} \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j)$$

$$\text{subject to: } \alpha_i \geq 0 \wedge \sum_{i=1}^{m} \alpha_i y_i = 0, i \in [1, m].$$

- Solution:

$$h(x) = \operatorname{sgn}\Big(\sum_{i=1}^{m} \alpha_i y_i (\mathbf{x}_i \cdot \mathbf{x}) + b \Big),$$

with $b = y_i - \sum_{j=1}^{m} \alpha_j y_j (\mathbf{x}_j \cdot \mathbf{x}_i)$ for any SV $\mathbf{x}_i$.

Leave-One-Out Error

- **Definition**: let h_S be the hypothesis output by learning algorithm L after receiving sample S of size m. Then, the leave-one-out error of L over S is:

$$\widehat{R}_{\text{loo}}(L) = \frac{1}{m} \sum_{i=1}^{m} 1_{h_{S-\{x_i\}}(x_i) \neq f(x_i)}.$$

- **Property**: unbiased estimate of expected error of hypothesis trained on sample of size $m-1$,

$$\boxed{\mathop{\mathrm{E}}_{S \sim D^m}[\widehat{R}_{\text{loo}}(L)]} = \frac{1}{m} \sum_{i=1}^{m} \mathop{\mathrm{E}}_{S}[1_{h_{S-\{x_i\}}(x_i) \neq f(x_i)}] = \mathop{\mathrm{E}}_{S}[1_{h_{S-\{x\}}(x) \neq f(x)}]$$

$$= \mathop{\mathrm{E}}_{S' \sim D^{m-1}}[\mathop{\mathrm{E}}_{x \sim D}[1_{h_{S'}(x) \neq f(x)}]] \boxed{= \mathop{\mathrm{E}}_{S' \sim D^{m-1}}[R(h_{S'})]}.$$

Leave-One-Out Analysis

- **Theorem**: let h_S be the optimal hyperplane for a sample S and let $N_{\rm SV}(S)$ be the number of support vectors defining h_S. Then,

$$\mathop{\rm E}_{S \sim D^m}[R(h_S)] \leq \mathop{\rm E}_{S \sim D^{m+1}}\left[\frac{N_{\rm SV}(S)}{m+1}\right].$$

- **Proof**: Let $S \sim D^{m+1}$ be a sample linearly separable and let $x \in S$. If $h_{S-\{x\}}$ misclassifies x, then x must be a SV for h_S. Thus,

$$\widehat{R}_{\rm loo}(\text{opt.-hyp.}) \leq \frac{N_{\rm SV}(S)}{m+1}.$$

Notes

- Bound on expectation of error only, not the probability of error.
- Argument based on sparsity (number of support vectors). We will see later other arguments in support of the optimal hyperplanes based on the concept of margin.

This Lecture

- Support Vector Machines - separable case
- Support Vector Machines - non-separable case
- Margin guarantees

Support Vector Machines

(Cortes and Vapnik, 1995)

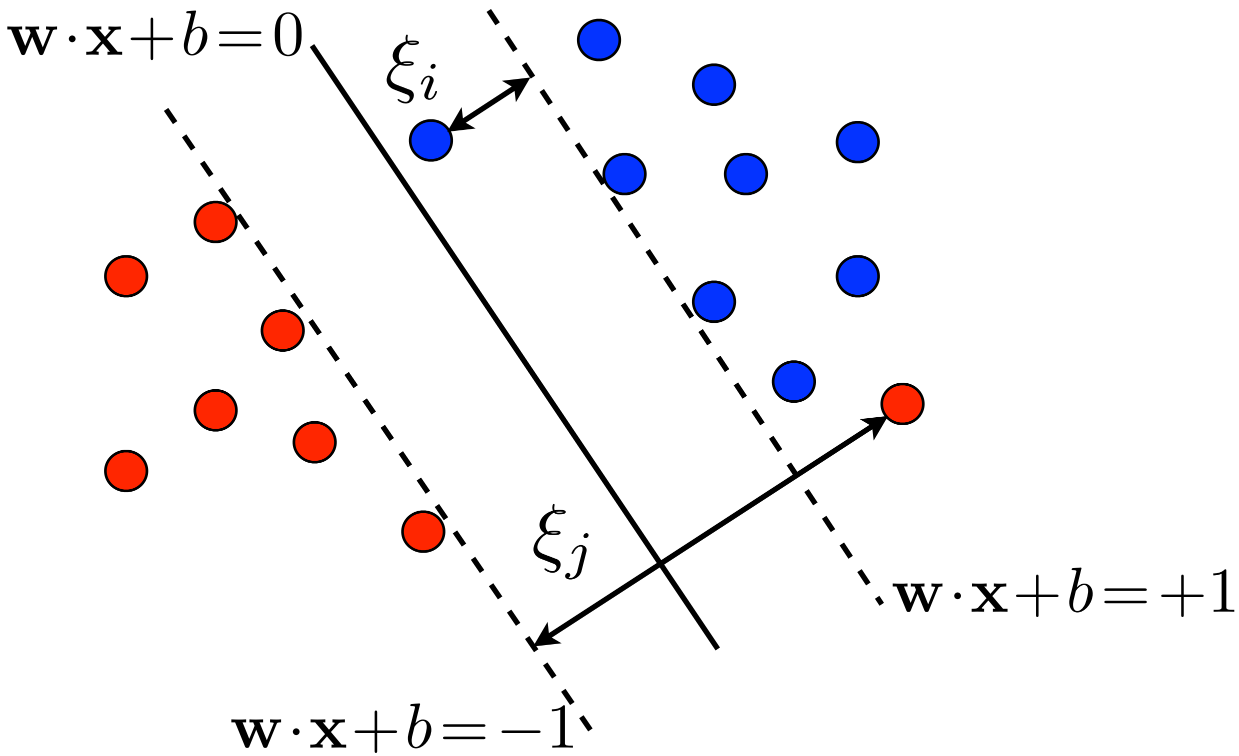
- Problem: data often not linearly separable in practice. For any hyperplane, there exists $\mathbf{x}_i$ such that

$$y_i \left[\mathbf{w} \cdot \mathbf{x}_i + b\right] \not\geq 1.$$

- Idea: relax constraints using slack variables $\xi_i \geq 0$

$$y_i \left[\mathbf{w} \cdot \mathbf{x}_i + b\right] \geq 1 - \xi_i.$$

Soft-Margin Hyperplanes



- Support vectors: points along the margin or outliers.
- Soft margin: $\rho = 1/\|\mathbf{w}\|$.

Optimization Problem

(Cortes and Vapnik, 1995)

- Constrained optimization:

$$\min_{\mathbf{w},b,\xi} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m}\xi_i$$

$$\text{subject to} \quad y_i(\mathbf{w}\cdot\mathbf{x}_i + b) \geq 1 - \xi_i \;\wedge\; \xi_i \geq 0, i \in [1, m].$$

- Properties:
  - $C \geq 0$ trade-off parameter.
  - Convex optimization.
  - Unique solution.

Notes

- Parameter C: trade-off between maximizing margin and minimizing training error. How do we determine C?
- The general problem of determining a hyperplane minimizing the error on the training set is NP-complete (as a function of the dimension).
- Other convex functions of the slack variables could be used: this choice and a similar one with squared slack variables lead to a convenient formulation and solution.

SVM - Equivalent Problem

- **Optimization:**

$$\min_{\mathbf{w},b} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m}\Big(1 - y_i(\mathbf{w}\cdot\mathbf{x}_i + b)\Big)_+.$$

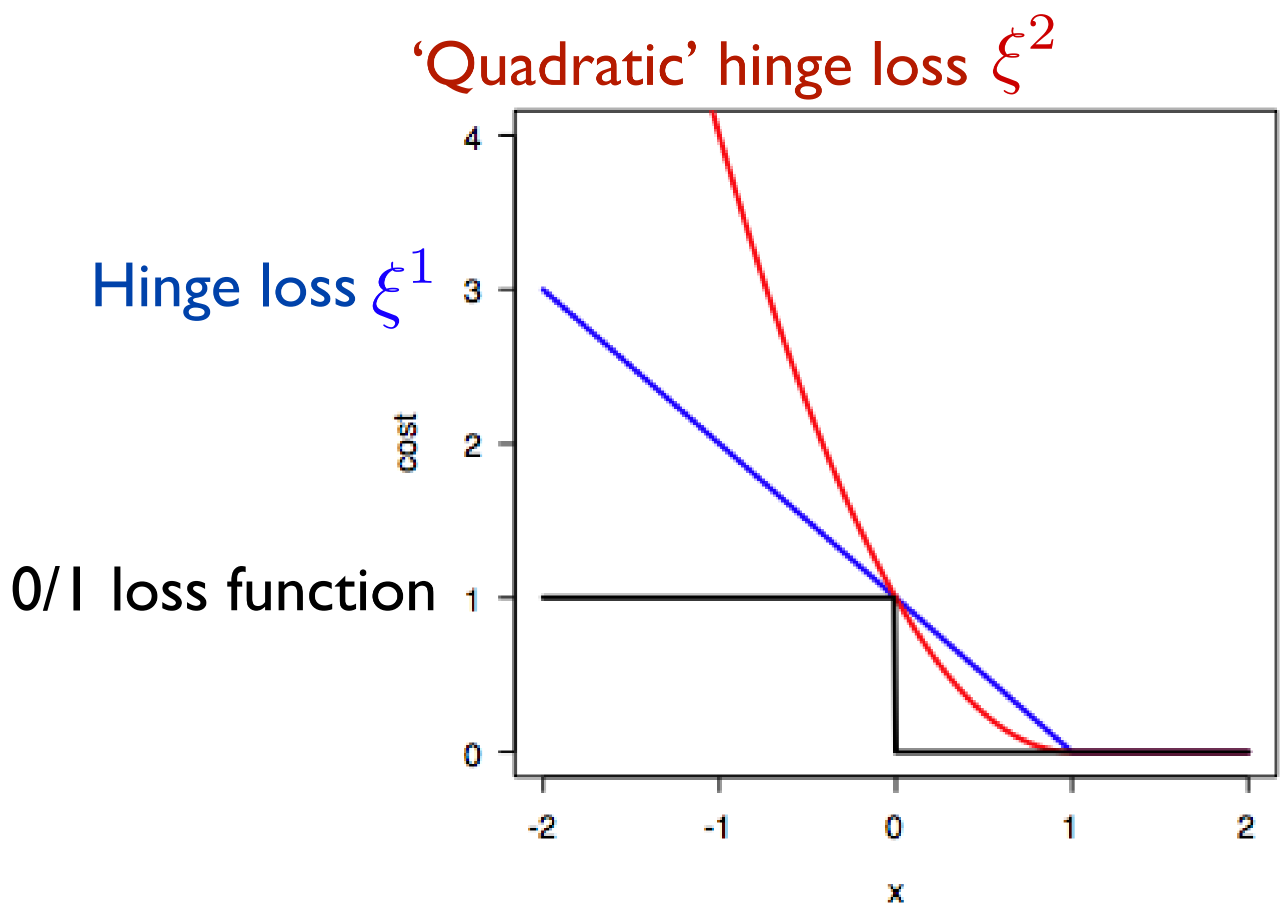
- **Loss functions:**
  - hinge loss:

$$L(h(x), y) = (1 - yh(x))_+.$$

  - quadratic hinge loss:

$$L(h(x), y) = (1 - yh(x))_+^2.$$

Hinge Loss


'Quadratic' hinge loss ξ^2
Hinge loss ξ^1
0/1 loss function
cost
x
4
3
2
1
0
-2
-1
0
1
2

SVMs Equations

- Lagrangian: for all $\mathbf{w}, b, \alpha_i \geq 0, \beta_i \geq 0,$

$$L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^m \xi_i - \sum_{i=1}^m \alpha_i[y_i(\mathbf{w}\cdot\mathbf{x}_i + b) - 1 + \xi_i] - \sum_{i=1}^m \beta_i \xi_i\,.$$

- KKT conditions:

$$\nabla_w L = \mathbf{w} - \sum_{i=1}^m \alpha_i y_i \mathbf{x}_i = 0 \iff \boxed{\mathbf{w} = \sum_{i=1}^m \alpha_i y_i \mathbf{x}_i.}$$

$$\nabla_b L = -\sum_{i=1}^m \alpha_i y_i = 0 \iff \boxed{\sum_{i=1}^m \alpha_i y_i = 0.}$$

$$\nabla_{\xi_i} L = C - \alpha_i - \beta_i = 0 \iff \boxed{\alpha_i + \beta_i = C.}$$

$$\boxed{\begin{aligned} &\forall i \in [1, m],\ \alpha_i[y_i(\mathbf{w}\cdot\mathbf{x}_i + b) - 1 + \xi_i] = 0 \\ &\qquad\qquad \beta_i \xi_i = 0. \end{aligned}}$$

Support Vectors

- Complementarity conditions:

$$\alpha_i[y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 + \xi_i] = 0 \implies \alpha_i = 0 \vee y_i(\mathbf{w} \cdot \mathbf{x}_i + b) = 1 - \xi_i.$$

- Support vectors: vectors $\mathbf{x}_i$ such that

$$\alpha_i \neq 0 \wedge y_i(\mathbf{w} \cdot \mathbf{x}_i + b) = 1 - \xi_i.$$

  - Note: support vectors are not unique.

Moving to The Dual

- Plugging in the expression of w in L gives:

$$L = \underbrace{\frac{1}{2}\Big\|\sum_{i=1}^m \alpha_i y_i \mathbf{x}_i\Big\|^2 - \sum_{i,j=1}^m \alpha_i \alpha_j y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j)}_{-\frac{1}{2}\sum_{i,j=1}^m \alpha_i \alpha_j y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j)} - \underbrace{\sum_{i=1}^m \alpha_i y_i b}_{0} + \sum_{i=1}^m \alpha_i.$$

- Thus,

$$L = \sum_{i=1}^m \alpha_i - \frac{1}{2}\sum_{i,j=1}^m \alpha_i \alpha_j y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j).$$

- The condition $\beta_i \geq 0$ is equivalent to $\alpha_i \leq C$.

Dual Optimization Problem

- Constrained optimization:

$$\max_{\alpha} \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j)$$

$$\text{subject to: } 0 \leq \alpha_i \boxed{\leq C} \wedge \sum_{i=1}^{m} \alpha_i y_i = 0, i \in [1, m].$$

- Solution:

$$h(x) = \operatorname{sgn}\Big(\sum_{i=1}^{m} \alpha_i y_i (\mathbf{x}_i \cdot \mathbf{x}) + b \Big),$$

with $b = y_i - \sum_{j=1}^{m} \alpha_j y_j (\mathbf{x}_j \cdot \mathbf{x}_i)$ for any $\mathbf{x}_i$ with $0 < \alpha_i \boxed{< C.}$

This Lecture

- Support Vector Machines - separable case
- Support Vector Machines - non-separable case
- Margin guarantees

High-Dimension

- Learning guarantees: for hyperplanes in dimension N with probability at least $1-\delta$,

$$R(h) \leq \widehat{R}(h) + \sqrt{\frac{2(N+1)\log\frac{em}{N+1}}{m}} + \sqrt{\frac{\log\frac{1}{\delta}}{2m}}.$$

  - bound is uninformative for $N \gg m$.
  - but SVMs have been remarkably successful in high-dimension.
  - can we provide a theoretical justification?
  - analysis of underlying scoring function.

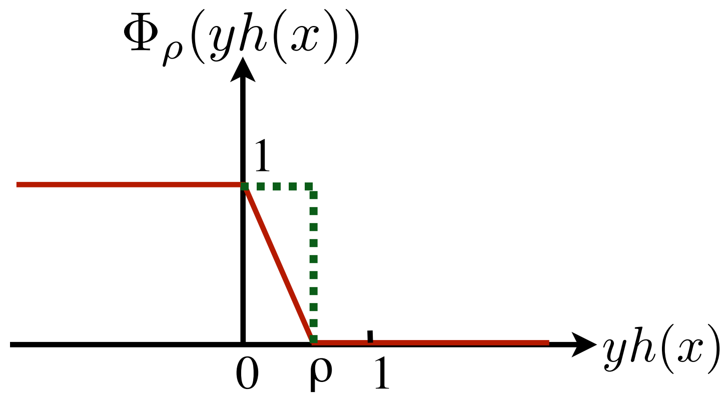
Confidence Margin

- Definition: the confidence margin of a real-valued function h at $(x, y) \in X \times Y$ is $\rho_h(x, y) = yh(x)$.
  - interpreted as the hypothesis' confidence in prediction.
  - if correctly classified coincides with $|h(x)|$.
  - relationship with geometric margin for linear functions $h \colon \mathbf{x} \mapsto \mathbf{w} \cdot \mathbf{x} + b$: for x in the sample,

$$|\rho_h(x, y)| \geq \rho_{\text{geom}} \|\mathbf{w}\|.$$

Confidence Margin Loss

- **Definition**: for any confidence margin parameter $\rho > 0$ the ρ-margin loss function Φ_ρ is defined by



- For a sample $S = (x_1, \ldots, x_m)$ and real-valued hypothesis h, the empirical margin loss is

$$\widehat{R}_\rho(h) = \frac{1}{m} \sum_{i=1}^{m} \Phi_\rho(y_i h(x_i)) \leq \boxed{\frac{1}{m} \sum_{i=1}^{m} 1_{y_i h(x_i) < \rho}}.$$

General Margin Bound

- **Theorem**: Let H be a set of real-valued functions. Fix $\rho > 0$. For any $\delta > 0$, with probability at least $1-\delta$, the following holds for all $h \in H$:

$$R(h) \leq \widehat{R}_\rho(h) + \frac{2}{\rho}\mathfrak{R}_m(H) + \sqrt{\frac{\log \frac{1}{\delta}}{2m}}$$

$$R(h) \leq \widehat{R}_\rho(h) + \frac{2}{\rho}\widehat{\mathfrak{R}}_S(H) + 3\sqrt{\frac{\log \frac{2}{\delta}}{2m}}.$$

- **Proof**: Let $\widetilde{H} = \{z = (x, y) \mapsto yh(x) : h \in H\}$. Consider the family of functions taking values in $[0, 1]$:

$$\widetilde{\mathcal{H}} = \{\Phi_\rho \circ f : f \in \widetilde{H}\}.$$

- By the theorem of Lecture 3, with probability at least $1-\delta$, for all $g \in \widetilde{\mathcal{H}}$,

$$\mathrm{E}[g(z)] \leq \frac{1}{m}\sum_{i=1}^{m} g(z_i) + 2\mathfrak{R}_m(\widetilde{\mathcal{H}}) + \sqrt{\frac{\log\frac{1}{\delta}}{2m}}.$$

- Thus,

$$\mathrm{E}[\Phi_\rho(yh(x))] \leq \widehat{R}_\rho(h) + 2\mathfrak{R}_m\big(\Phi_\rho \circ \widetilde{H}\big) + \sqrt{\frac{\log\frac{1}{\delta}}{2m}}.$$

- Since Φ_ρ is $\frac{1}{\rho}$ - Lipschitz, by Talagrand's lemma,

$$\mathfrak{R}_m\big(\Phi_\rho \circ \widetilde{H}\big) \leq \frac{1}{\rho}\mathfrak{R}_m(\widetilde{H}) = \frac{1}{\rho m}\mathop{\mathrm{E}}_{\sigma,S}\Big[\sup_{h\in H}\sum_{i=1}^{m}\sigma_i y_i h(x_i)\Big] = \frac{1}{\rho}\mathfrak{R}_m\big(H\big).$$

- Since $1_{yh(x)<0} \leq \Phi_\rho(yh(x))$, this shows the first statement, and similarly the second one.

Rademacher Complexity of Linear Hypotheses

- **Theorem**: Let $S \subseteq \{x \colon \|\mathbf{x}\| \leq R\}$ be a sample of size m and let $H = \{\mathbf{x} \mapsto \mathbf{w} \cdot \mathbf{x} \colon \|\mathbf{w}\| \leq \Lambda\}$. Then,

$$\widehat{\mathfrak{R}}_S(H) \leq \sqrt{\frac{R^2 \Lambda^2}{m}}.$$

- **Proof**:

$$\begin{aligned}
\widehat{\mathfrak{R}}_S(H) &= \frac{1}{m} \operatorname*{E}_{\sigma} \left[\sup_{\|\mathbf{w}\| \leq \Lambda} \sum_{i=1}^{m} \sigma_i \mathbf{w} \cdot \mathbf{x}_i \right] = \frac{1}{m} \operatorname*{E}_{\sigma} \left[\sup_{\|\mathbf{w}\| \leq \Lambda} \mathbf{w} \cdot \sum_{i=1}^{m} \sigma_i \mathbf{x}_i \right] \\
&\leq \frac{\Lambda}{m} \operatorname*{E}_{\sigma} \left[\left\| \sum_{i=1}^{m} \sigma_i \mathbf{x}_i \right\| \right] \leq \frac{\Lambda}{m} \left[\operatorname*{E}_{\sigma} \left[\left\| \sum_{i=1}^{m} \sigma_i \mathbf{x}_i \right\|^2 \right] \right]^{1/2} \\
&\leq \frac{\Lambda}{m} \left[\operatorname*{E}_{\sigma} \left[\sum_{i=1}^{m} \|\mathbf{x}_i\|^2 \right] \right]^{1/2} \leq \frac{\Lambda \sqrt{m R^2}}{m} = \sqrt{\frac{R^2 \Lambda^2}{m}}.
\end{aligned}$$

Margin Bound - Linear Classifiers

- **Corollary**: Let $\rho > 0$ and $H = \{\mathbf{x} \mapsto \mathbf{w} \cdot \mathbf{x} \colon \|\mathbf{w}\| \leq \Lambda\}$. Assume that $X \subseteq \{\mathbf{x} \colon \|\mathbf{x}\| \leq R\}$. Then, for any $\delta > 0$, with probability at least $1 - \delta$, for any $h \in H$,

$$R(h) \leq \widehat{R}_\rho(h) + 2\sqrt{\frac{R^2\Lambda^2/\rho^2}{m}} + 3\sqrt{\frac{\log \frac{2}{\delta}}{2m}}.$$

- **Proof**: Follows directly general margin bound and bound on $\widehat{\mathfrak{R}}_S(H)$ for linear classifiers.
  - Finer relative deviation margin bounds (Cortes, MM, Suresh; ICML 2021).

High-Dimensional Feature Space

- Observations:
  - generalization bound does not depend on the dimension but on the margin.
  - this suggests seeking a large-margin hyperplane in a higher-dimensional feature space.
- Computational problems:
  - taking dot products in a high-dimensional feature space can be very costly.
  - solution based on kernels (next lecture).

References

- Corinna Cortes and Vladimir Vapnik, Support-Vector Networks, *Machine Learning*, 20, 1995.
- Koltchinskii, Vladimir and Panchenko, Dmitry. Empirical margin distributions and bounding the generalization error of combined classifiers. The Annals of Statistics, 30(1), 2002.
- Ledoux, M. and Talagrand, M. (1991). *Probability in Banach Spaces*. Springer, New York.
- Vladimir N. Vapnik. *Estimation of Dependences Based on Empirical Data*. Springer, Basederlin, 1982.
- Vladimir N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 1995.
- Vladimir N. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, New York, 1998.

Appendix

Saddle Point

- Let $(\mathbf{w}^*, b^*, \alpha^*)$ be the saddle point of the Langrangian. Multiplying both sides of the equation giving b^* by $\alpha_i^* y_i$ and taking the sum leads to:

$$\sum_{i=1}^{m} \alpha_i^* y_i b = \sum_{i=1}^{m} \alpha_i^* y_i^2 - \sum_{i,j=1}^{m} \alpha_i^* \alpha_j^* y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j).$$

- Using $y_i^2 = 1$, $\sum_{i=1}^{m} \alpha_i^* y_i = 0$, and $\mathbf{w}^* = \sum_{i=1}^{m} \alpha_i^* y_i \mathbf{x}_i$ yields

$$0 = \sum_{i=1}^{m} \alpha_i^* - \|\mathbf{w}^*\|^2.$$

- Thus, the margin is also given by:

$$\boxed{\rho^2 = \frac{1}{\|\mathbf{w}^*\|_2^2} = \frac{1}{\|\alpha^*\|_1}.}$$

Talagrand's Contraction Lemma

(Ledoux and Talagrand, 1991; pp. 112-114)

- **Theorem**: Let $\Phi\colon \mathbb{R} \to \mathbb{R}$ be an L-Lipschitz function. Then, for any hypothesis set H of real-valued functions,

$$\widehat{\mathfrak{R}}_S(\Phi \circ H) \leq L\, \widehat{\mathfrak{R}}_S(H).$$

- **Proof**: fix sample $S = (x_1, \ldots, x_m)$. By definition,

$$\begin{aligned}
\mathfrak{R}_S(\Phi \circ H) &= \frac{1}{m} \mathop{\mathrm{E}}_{\boldsymbol{\sigma}} \left[\sup_{h \in H} \sum_{i=1}^{m} \sigma_i (\Phi \circ h)(x_i) \right] \\
&= \frac{1}{m} \mathop{\mathrm{E}}_{\sigma_1, \ldots, \sigma_{m-1}} \left[\mathop{\mathrm{E}}_{\sigma_m} \left[\sup_{h \in H} u_{m-1}(h) + \sigma_m (\Phi \circ h)(x_m) \right] \right],
\end{aligned}$$

with $u_{m-1}(h) = \sum_{i=1}^{m-1} \sigma_i (\Phi \circ h)(x_i)$.

Talagrand's Contraction Lemma

- Now, assuming that the suprema are reached, there exist $h_1, h_2 \in H$ such that

$$
\begin{aligned}
&\mathop{\mathrm{E}}_{\sigma_m}\left[\sup_{h\in H} u_{m-1}(h) + \sigma_m(\Phi \circ h)(x_m)\right]\Big] \\
&= \frac{1}{2}[u_{m-1}(h_1) + (\Phi \circ h_1)(x_m)] + \frac{1}{2}[u_{m-1}(h_2) - (\Phi \circ h_2)(x_m)] \\
&\leq \frac{1}{2}[u_{m-1}(h_1) + u_{m-1}(h_2) + sL(h_1(x_m) - h_2(x_m))] \\
&= \frac{1}{2}[u_{m-1}(h_1) + sLh_1(x_m)] + \frac{1}{2}[u_{m-1}(h_2) - sLh_2(x_m)] \\
&\leq \mathop{\mathrm{E}}_{\sigma_m}\left[\sup_{h\in H} u_{m-1}(h) + \sigma_m L h(x_m)\right],
\end{aligned}
$$

where $s = \mathrm{sgn}(h_1(x_m) - h_2(x_m))$.

Talagrand's Contraction Lemma

- When the suprema are not reached, the same can be shown modulo ϵ, followed by $\epsilon \to 0$.
- Proceeding similarly for other σ_is directly leads to the result.

VC Dimension of Canonical Hyperplanes

- **Theorem**: Let $S \subseteq \{\mathbf{x} : \|\mathbf{x}\| \leq R\}$. Then, the VC dimension d of the set of canonical hyperplanes $\{x \mapsto \text{sgn}(\mathbf{w} \cdot \mathbf{x}) \colon \min_{x \in S} |\mathbf{w} \cdot \mathbf{x}| = 1 \wedge \|\mathbf{w}\| \leq \Lambda\}$ verifies

$$d \leq R^2 \Lambda^2.$$

- **Proof**: Let $\{\mathbf{x}_1, \ldots, \mathbf{x}_d\}$ be a set fully shattered. Then, for all $\mathbf{y} \in \{-1, +1\}^d$, there exists $\mathbf{w}$ such

$$\forall i \in [1, d], 1 \leq y_i(\mathbf{w} \cdot \mathbf{x}_i).$$

- Summing up the inequalities gives

$$d \leq \mathbf{w} \cdot \sum_{i=1}^{d} y_i \mathbf{x}_i \leq \|\mathbf{w}\| \|\sum_{i=1}^{d} y_i \mathbf{x}_i\| \leq \Lambda \|\sum_{i=1}^{d} y_i \mathbf{x}_i\|.$$

- Taking the expectation over $\mathbf{y} \sim U$ (uniform) yields

$$d \leq \Lambda \operatorname*{E}_{\mathbf{y}\sim U}[\|\sum_{i=1}^{d} y_i \mathbf{x}_i\|] \leq \Lambda \Big[\operatorname*{E}_{\mathbf{y}\sim U}[\|\sum_{i=1}^{d} y_i \mathbf{x}_i\|^2] \Big]^{1/2} \text{ (Jensen's ineq.)}$$

$$= \Lambda \Big[\sum_{i,j=1}^{d} \mathrm{E}[y_i y_j](\mathbf{x}_i \cdot \mathbf{x}_j) \Big]^{1/2}$$

$$= \Lambda \Big[\sum_{i=1}^{d} (\mathbf{x}_i \cdot \mathbf{x}_i) \Big]^{1/2} \leq \Lambda \big[dR^2 \big]^{1/2} = \Lambda R \sqrt{d}.$$

- Thus, $\sqrt{d} \leq \Lambda R$.

Foundations of Machine Learning
Kernel Methods

Mehryar Mohri
Courant Institute and Google Research
mohri@cims.nyu.edu

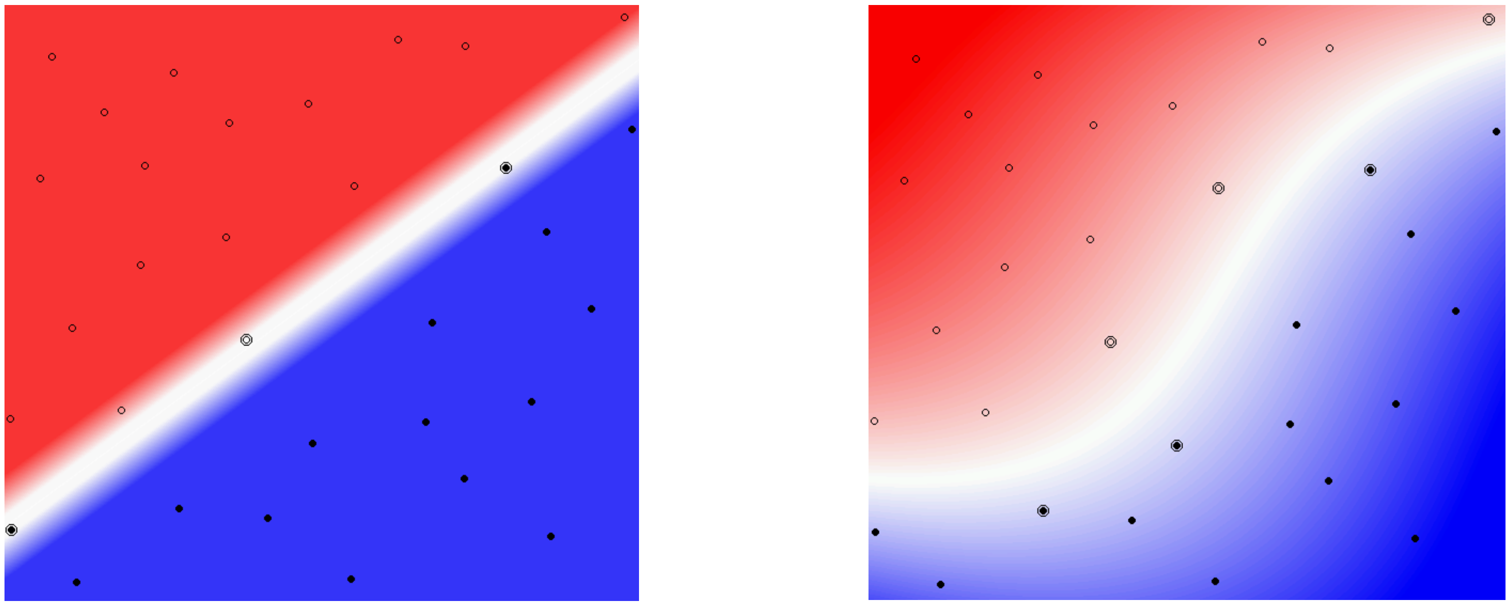
Motivation

- Efficient computation of inner products in high dimension.
- Non-linear decision boundary.
- Non-vectorial inputs.
- Flexible selection of more complex features.

This Lecture

- Kernels
- Kernel-based algorithms
- Closure properties
- Sequence Kernels
- Negative kernels

Non-Linear Separation



- Linear separation impossible in most problems.
- Non-linear mapping from input space to high-dimensional feature space: $\Phi\colon X \to F$.
- Generalization ability: independent of $\dim(F)$, depends only on margin and sample size.

Kernel Methods

- Idea:
  - Define $K: X \times X \to \mathbb{R}$, called kernel, such that:
  $$\Phi(x) \cdot \Phi(y) = K(x, y).$$
  - K often interpreted as a similarity measure.
- Benefits:
  - Efficiency: K is often more efficient to compute than Φ and the dot product.
  - Flexibility: K can be chosen arbitrarily so long as the existence of Φ is guaranteed (PDS condition or Mercer's condition).

PDS Condition

- **Definition**: a kernel $K\colon X \times X \to \mathbb{R}$ is positive definite symmetric (PDS) if for any $\{x_1, \ldots, x_m\} \subseteq X$, the matrix $\mathbf{K} = [K(x_i, x_j)]_{ij} \in \mathbb{R}^{m \times m}$ is symmetric positive semi-definite (SPSD).
- $\mathbf{K}$ SPSD if symmetric and one of the 2 equiv. cond.'s:
  - its eigenvalues are non-negative.
  - for any $\mathbf{c} \in \mathbb{R}^{m \times 1}$, $\mathbf{c}^\top \mathbf{K} \mathbf{c} = \sum_{i,j=1}^{m} c_i c_j K(x_i, x_j) \geq 0.$
- **Terminology**: PDS for kernels, SPSD for kernel matrices (see (Berg et al., 1984)).

Example - Polynomial Kernels

- **Definition:**

$$\forall x, y \in \mathbb{R}^N, \;\; K(x, y) = (x \cdot y + c)^d, \quad c > 0.$$

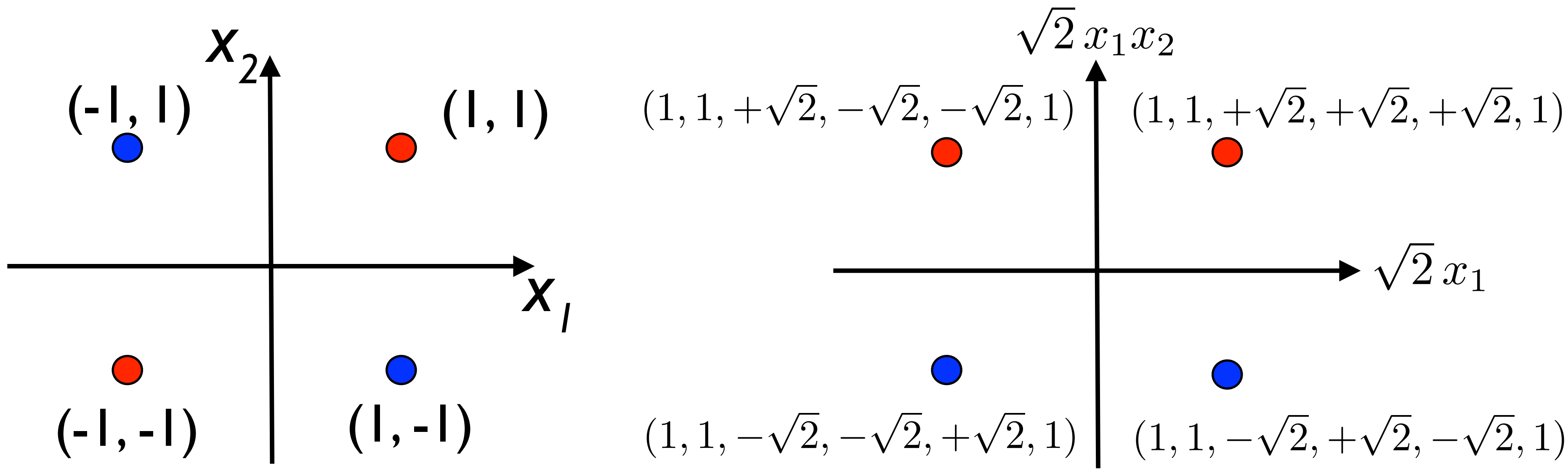
- **Example:** for $N = 2$ and $d = 2$,

$$K(x, y) = (x_1 y_1 + x_2 y_2 + c)^2$$

$$= \begin{bmatrix} x_1^2 \\ x_2^2 \\ \sqrt{2}\, x_1 x_2 \\ \sqrt{2c}\, x_1 \\ \sqrt{2c}\, x_2 \\ c \end{bmatrix} \cdot \begin{bmatrix} y_1^2 \\ y_2^2 \\ \sqrt{2}\, y_1 y_2 \\ \sqrt{2c}\, y_1 \\ \sqrt{2c}\, y_2 \\ c \end{bmatrix}.$$

XOR Problem

- Use second-degree polynomial kernel with $c = 1$:



Linearly non-separable

Linearly separable by $x_1 x_2 = 0.$

Normalized Kernels

- **Definition**: the normalized kernel K' associated to a kernel K is defined by

$$\forall x, x' \in \mathcal{X}, \; K'(x, x') = \begin{cases} 0 & \text{if } (K(x,x) = 0) \vee (K(x',x') = 0) \\ \frac{K(x,x')}{\sqrt{K(x,x)K(x',x')}} & \text{otherwise.} \end{cases}$$

- If K is PDS, then K' is PDS:

$$\sum_{i,j=1}^{m} \frac{c_i c_j K(x_i, x_j)}{\sqrt{K(x_i,x_i)K(x_j,x_j)}} = \sum_{i,j=1}^{m} \frac{c_i c_j \langle \Phi(x_i), \Phi(x_j) \rangle}{\|\Phi(x_i)\|_H \, \|\Phi(x_j)\|_{\mathbb{H}}} = \left\| \sum_{i=1}^{m} \frac{c_i \Phi(x_i)}{\|\Phi(x_i)\|_H} \right\|_{\mathbb{H}}^2 \geq 0.$$

- By definition, for all x with $K(x,x) \neq 0$,

$$K'(x, x) = 1.$$

Other Standard PDS Kernels

- Gaussian kernels:

$$K(x, y) = \exp\left(-\frac{||x-y||^2}{2\sigma^2}\right), \ \sigma \neq 0.$$

  - Normalized kernel of $(\mathbf{x}, \mathbf{x}') \mapsto \exp\left(\frac{\mathbf{x} \cdot \mathbf{x}'}{\sigma^2}\right)$.

- Sigmoid Kernels:

$$K(x, y) = \tanh(a(x \cdot y) + b), \ a, b \geq 0.$$

Reproducing Kernel Hilbert Space

(Aronszajn, 1950)

- Theorem: Let $K\colon X \times X \to \mathbb{R}$ be a PDS kernel. Then, there exists a Hilbert space H and a mapping Φ from X to H such that

$$\forall x, y \in X, \;\; K(x, y) = \Phi(x) \cdot \Phi(y).$$

- Proof: For any $x \in X$, define $\Phi(x)\colon X \to \mathbb{R}^X$ as follows:

$$\forall y \in X, \;\; \Phi(x)(y) = K(x, y).$$

  - Let $H_0 = \Big\{ \sum_{i \in I} a_i \Phi(x_i) \colon a_i \in \mathbb{R}, x_i \in X, \mathrm{card}(I) < \infty \Big\}.$
  - We are going to define an inner product $\langle \cdot, \cdot \rangle$ on H_0.

- Definition: for any $f = \sum_{i \in I} a_i \Phi(x_i)$, $g = \sum_{j \in J} b_j \Phi(y_j)$,

$$\langle f, g \rangle = \sum_{i \in I, j \in J} a_i b_j K(x_i, y_j) = \sum_{j \in J} b_j f(y_j) = \sum_{i \in I} a_i g(x_i).$$

- $\langle \cdot, \cdot \rangle$ does not depend on representations of f and g.
- $\langle \cdot, \cdot \rangle$ is bilinear and symmetric.
- $\langle \cdot, \cdot \rangle$ is positive semi-definite since K is PDS: for any f,

$$\langle f, f \rangle = \sum_{i, j \in I} a_i a_j K(x_i, x_j) \geq 0.$$

- note: for any $f_1, \ldots, f_m$ and $c_1, \ldots, c_m$,

$$\sum_{i,j=1}^{m} c_i c_j \langle f_i, f_j \rangle = \Big\langle \sum_{i=1}^{m} c_i f_i, \sum_{j=1}^{m} c_j f_j \Big\rangle \geq 0.$$

$\longrightarrow$ $\langle \cdot, \cdot \rangle$ is a PDS kernel on H_0.

- $\langle \cdot, \cdot \rangle$ is definite:
  - first, Cauchy-Schwarz inequality for PDS kernels. If K is PDS, $\mathbf{M} = \begin{pmatrix} K(x,x) & K(x,y) \\ K(y,x) & K(y,y) \end{pmatrix}$ is SPSD for all $x, y \in X$. In particular, the product of its eigenvalues, $\det(\mathbf{M})$ is non-negative:

$$\det(\mathbf{M}) = K(x,x)K(y,y) - K(x,y)^2 \geq 0.$$

  - since $\langle \cdot, \cdot \rangle$ is a PDS kernel, for any $f \in H_0$ and $x \in X$,

$$\langle f, \Phi(x) \rangle^2 \leq \langle f, f \rangle \langle \Phi(x), \Phi(x) \rangle.$$

  - observe the reproducing property of $\langle \cdot, \cdot \rangle$:

$$\forall f \in H_0, \forall x \in X, \ \ f(x) = \sum_{i \in I} a_i K(x_i, x) = \langle f, \Phi(x) \rangle.$$

  - Thus, $[f(x)]^2 \leq \langle f, f \rangle K(x,x)$ for all $x \in X$, which shows the definiteness of $\langle \cdot, \cdot \rangle$.

- Thus, $\langle \cdot, \cdot \rangle$ defines an inner product on H_0, which thereby becomes a pre-Hilbert space.
- H_0 can be completed to form a Hilbert space H in which it is dense.

Notes:

- H is called the reproducing kernel Hilbert space (RKHS) associated to K.
- A Hilbert space such that there exists $\Phi: X \to H$ with $K(x, y) = \Phi(x) \cdot \Phi(y)$ for all $x, y \in X$ is also called a feature space associated to K. Φ is called a feature mapping.
- Feature spaces associated to K are in general not unique.

This Lecture

- Kernels
- Kernel-based algorithms
- Closure properties
- Sequence Kernels
- Negative kernels

SVMs with PDS Kernels

(Boser, Guyon, and Vapnik, 1992)

- Constrained optimization:

$$\max_{\alpha} \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j K(x_i, x_j)$$

$\Phi(x_i) \cdot \Phi(x_j)$

$$\text{subject to: } 0 \leq \alpha_i \leq C \wedge \sum_{i=1}^{m} \alpha_i y_i = 0, i \in [1, m].$$

- Solution:

$$h(x) = \operatorname{sgn}\Big(\sum_{i=1}^{m} \alpha_i y_i K(x_i, x) + b \Big),$$

with $b = y_i - \sum_{j=1}^{m} \alpha_j y_j K(x_j, x_i)$ for any x_i with $0 < \alpha_i < C.$

Rad. Complexity of Kernel-Based Hypotheses

- **Theorem**: Let $K\colon X \times X \to \mathbb{R}$ be a PDS kernel and let $\Phi: X \to \mathbb{H}$ be a feature mapping associated to K. Let $S \subseteq \{x : K(x,x) \leq R^2\}$ be a sample of size m, and let $H = \{\mathbf{x} \mapsto \mathbf{w} \cdot \Phi(x) : \|\mathbf{w}\|_{\mathbb{H}} \leq \Lambda\}$. Then,

$$\widehat{\mathfrak{R}}_S(H) \leq \frac{\Lambda\sqrt{\mathrm{Tr}[\mathbf{K}]}}{m} \leq \sqrt{\frac{R^2\Lambda^2}{m}}.$$

- **Proof**: $\widehat{\mathfrak{R}}_S(H) = \frac{1}{m}\mathop{\mathrm{E}}_{\sigma}\Big[\sup_{\|\mathbf{w}\|\leq\Lambda} \mathbf{w} \cdot \sum_{i=1}^m \sigma_i \Phi(x_i)\Big] \leq \frac{\Lambda}{m}\mathop{\mathrm{E}}_{\sigma}\Big[\Big\|\sum_{i=1}^m \sigma_i \Phi(x_i)\Big\|\Big]$

$$\text{(Jensen's ineq.)} \leq \frac{\Lambda}{m}\bigg[\mathop{\mathrm{E}}_{\sigma}\Big[\Big\|\sum_{i=1}^m \sigma_i \Phi(x_i)\Big\|^2\Big]\bigg]^{1/2} \leq \frac{\Lambda}{m}\bigg[\mathop{\mathrm{E}}_{\sigma}\Big[\sum_{i=1}^m \|\Phi(x_i)\|^2\Big]\bigg]^{1/2}$$

$$= \frac{\Lambda}{m}\bigg[\mathop{\mathrm{E}}_{\sigma}\Big[\sum_{i=1}^m K(x_i, x_i)\Big]\bigg]^{1/2} = \frac{\Lambda\sqrt{\mathrm{Tr}[\mathbf{K}]}}{m} \leq \sqrt{\frac{R^2\Lambda^2}{m}}.$$

Generalization: Representer Theorem

(Kimeldorf and Wahba, 1971)

- **Theorem**: Let $K\colon X \times X \to \mathbb{R}$ be a PDS kernel with H the corresponding RKHS. Then, for any non-decreasing function $G\colon \mathbb{R} \to \mathbb{R}$ and any $L\colon \mathbb{R}^m \to \mathbb{R} \cup \{+\infty\}$ problem

$$\operatorname*{argmin}_{h \in H} F(h) = \operatorname*{argmin}_{h \in H} \; G(\|h\|_H) + L\big(h(x_1), \ldots, h(x_m)\big)$$

admits a solution of the form $h^* = \sum_{i=1}^{m} \alpha_i K(x_i, \cdot)$.

If G is further assumed to be increasing, then any solution has this form.

- Proof: let $H_1 = \text{span}(\{K(x_i, \cdot) : i \in [1, m]\})$. Any $h \in H$ admits the decomposition $h = h_1 + h^\perp$ according to $H = H_1 \oplus H_1^\perp$.
  - Since G is non-decreasing,
  $$G(\|h_1\|_H) \leq G\Big(\sqrt{\|h_1\|_H^2 + \|h^\perp\|_H^2}\Big) = G(\|h\|_H).$$
  - By the reproducing property, for all $i \in [1, m]$,
  $$h(x_i) = \langle h, K(x_i, \cdot)\rangle = \langle h_1, K(x_i, \cdot)\rangle = h_1(x_i).$$
  - Thus, $L\big(h(x_1), \ldots, h(x_m)\big) = L\big(h_1(x_1), \ldots, h_1(x_m)\big)$ and $F(h_1) \leq F(h)$.
  - If G is increasing, then $F(h_1) < F(h)$ when $h^\perp \neq 0$ and any solution of the optimization problem must be in H_1.

Kernel-Based Algorithms

- PDS kernels used to extend a variety of algorithms in classification and other areas:
  - regression.
  - ranking.
  - dimensionality reduction.
  - clustering.
- But, how do we define PDS kernels?

This Lecture

- Kernels
- Kernel-based algorithms
- Closure properties
- Sequence Kernels
- Negative kernels

Closure Properties of PDS Kernels

- Theorem: Positive definite symmetric (PDS) kernels are closed under:
  - sum,
  - product,
  - tensor product,
  - pointwise limit,
  - composition with a power series with non-negative coefficients.

Closure Properties - Proof

- **Proof**: closure under sum:

$$\mathbf{c}^\top \mathbf{K}\mathbf{c} \geq 0 \wedge \mathbf{c}^\top \mathbf{K}'\mathbf{c} \geq 0 \Rightarrow \mathbf{c}^\top (\mathbf{K} + \mathbf{K}')\mathbf{c} \geq 0.$$

  - closure under product: $\mathbf{K} = \mathbf{M}\mathbf{M}^\top$,

$$\begin{aligned}
\sum_{i,j=1}^{m} c_i c_j (\mathbf{K}_{ij}\mathbf{K}'_{ij}) &= \sum_{i,j=1}^{m} c_i c_j \left(\left[\sum_{k=1}^{m} \mathbf{M}_{ik}\mathbf{M}_{jk} \right] \mathbf{K}'_{ij} \right) \\
&= \sum_{k=1}^{m} \left[\sum_{i,j=1}^{m} c_i c_j \mathbf{M}_{ik}\mathbf{M}_{jk}\mathbf{K}'_{ij} \right] \\
&= \sum_{k=1}^{m} \begin{bmatrix} c_1\mathbf{M}_{1k} \\ \cdots \\ c_m\mathbf{M}_{mk} \end{bmatrix}^\top \mathbf{K}' \begin{bmatrix} c_1\mathbf{M}_{1k} \\ \cdots \\ c_m\mathbf{M}_{mk} \end{bmatrix} \geq 0.
\end{aligned}$$

- Closure under tensor product:
  - definition: for all $x_1, x_2, y_1, y_2 \in X$,

$$(K_1 \otimes K_2)(x_1, y_1, x_2, y_2) = K_1(x_1, x_2) K_2(y_1, y_2).$$

  - thus, PDS kernel as product of the kernels

$$(x_1, y_1, x_2, y_2) \to K_1(x_1, x_2) \quad (x_1, y_1, x_2, y_2) \to K_2(y_1, y_2).$$

- Closure under pointwise limit: if for all $x, y \in X$,

$$\lim_{n \to \infty} K_n(x, y) = K(x, y),$$

Then, $(\forall n, \mathbf{c}^\top \mathbf{K}_n \mathbf{c} \geq 0) \Rightarrow \lim_{n \to \infty} \mathbf{c}^\top \mathbf{K}_n \mathbf{c} = \mathbf{c}^\top \mathbf{K} \mathbf{c} \geq 0.$

- Closure under composition with power series:
  - assumptions: K PDS kernel with $|K(x, y)| < \rho$ for all $x, y \in X$ and $f(x) = \sum_{n=0}^{\infty} a_n x^n, a_n \geq 0$ power series with radius of convergence ρ.
  - $f \circ K$ is a PDS kernel since K^n is PDS by closure under product, $\sum_{n=0}^{N} a_n K^n$ is PDS by closure under sum, and closure under pointwise limit.
- Example: for any PDS kernel K, $\exp(K)$ is PDS.

This Lecture

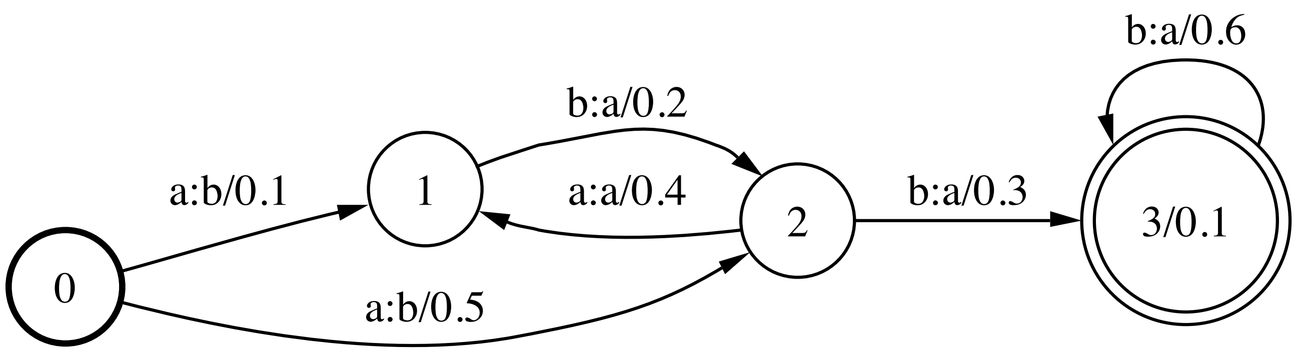
- Kernels
- Kernel-based algorithms
- Closure properties
- Sequence Kernels
- Negative kernels

Sequence Kernels

- Definition: Kernels defined over pairs of strings.
  - Motivation: computational biology, text and speech classification.
  - Idea: two sequences are related when they share some common substrings or subsequences.
  - Example: bigram kernel;

$$K(x, y) = \sum_{\text{bigram } u} \text{count}_x(u) \times \text{count}_y(u).$$

Weighted Transducers



$T(x, y) =$ Sum of the weights of all accepting paths with input x and output y.

$$T(abb, baa) = .1 \times .2 \times .3 \times .1 + .5 \times .3 \times .6 \times .1$$

Rational Kernels over Strings

(Cortes et al., 2004)

- Definition: a kernel $K:\Sigma^*\times\Sigma^*\rightarrow\mathbb{R}$ is rational if $K=T$ for some weighted transducer T.

- Definition: let $T_1:\Sigma^*\times\Delta^*\rightarrow\mathbb{R}$ and $T_2:\Delta^*\times\Omega^*\rightarrow\mathbb{R}$ be two weighted transducers. Then, the composition of T_1 and T_2 is defined for all $x\in\Sigma^*, y\in\Omega^*$ by

$$(T_1\circ T_2)(x,y)=\sum_{z\in\Delta^*} T_1(x,z)\ T_2(z,y).$$

- Definition: the inverse of a transducer $T:\Sigma^*\times\Delta^*\rightarrow\mathbb{R}$ is the transducer $T^{-1}:\Delta^*\times\Sigma^*\rightarrow\mathbb{R}$ obtained from T by swapping input and output labels.

PDS Rational Kernels
General Construction

- **Theorem**: for any weighted transducer $T\colon \Sigma^* \times \Sigma^* \to \mathbb{R}$, the function $K = T \circ T^{-1}$ is a PDS rational kernel.

- **Proof**: by definition, for all $x, y \in \Sigma^*$,

$$K(x, y) = \sum_{z \in \Delta^*} T(x, z)\, T(y, z).$$

  - K is pointwise limit of $(K_n)_{n \geq 0}$ defined by

$$\forall x, y \in \Sigma^*, \;\; K_n(x, y) = \sum_{|z| \leq n} T(x, z)\, T(y, z).$$

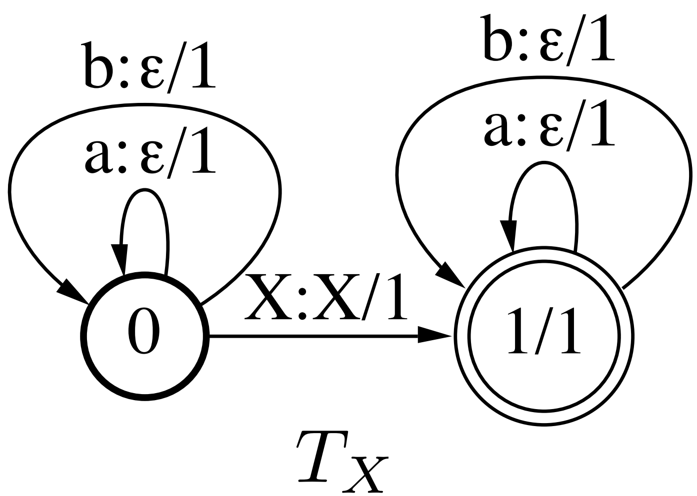
  - K_n is PDS since for any sample $(x_1, \ldots, x_m)$,

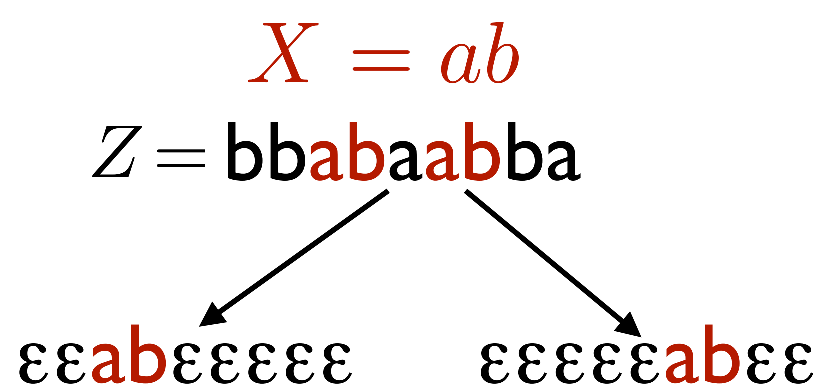
$$\mathbf{K}_n = \mathbf{A}\mathbf{A}^\top \text{ with } \mathbf{A} = (K_n(x_i, z_j))_{\substack{i \in [1, m] \\ j \in [1, N]}}.$$

PDS Sequence Kernels

- PDS sequences kernels in computational biology, text classification, other applications:
  - special instances of PDS rational kernels.
  - PDS rational kernels easy to define and modify.
  - single general algorithm for their computation: composition + shortest-distance computation.
  - no need for a specific 'dynamic-programming' algorithm and proof for each kernel instance.
  - general sub-family: based on counting transducers.

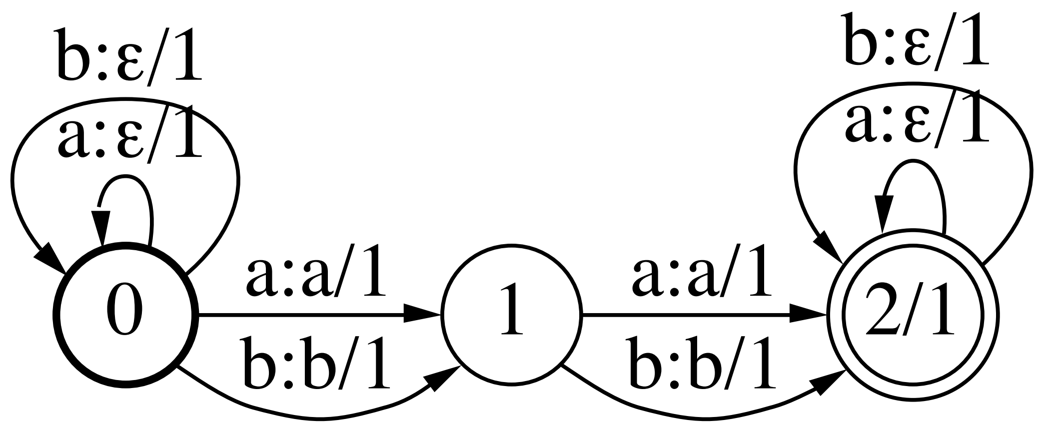
Counting Transducers





- X may be a string or an automaton representing a regular expression.
- Counts of Z in X: sum of the weights of accepting paths of $Z \circ T_X$.

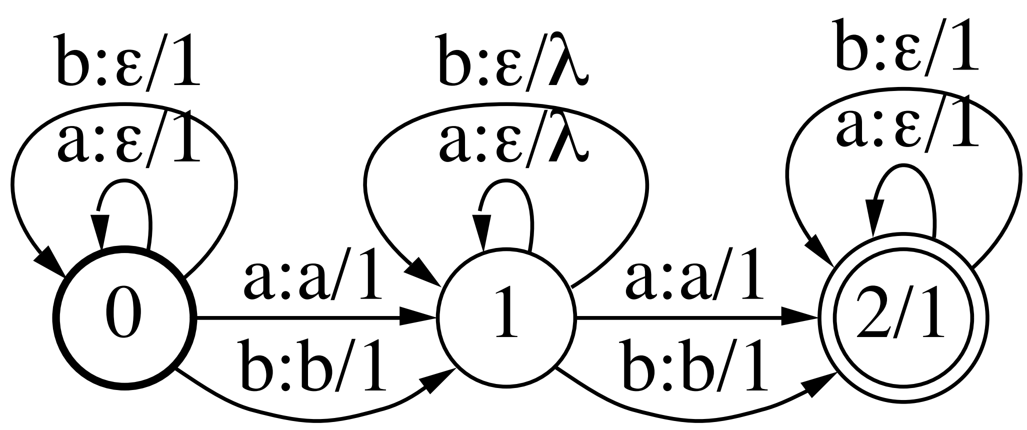
Transducer Counting Bigrams



T_{bigram}

Counts of Z given by $Z \circ T_{\text{bigram}} \circ ab$.

Transducer Counting Gappy Bigrams



$T_{\text{gappy bigram}}$

Counts of Z given by $Z \circ T_{\text{gappy bigram}} \circ ab$,
gap penalty $\lambda \in (0, 1)$.

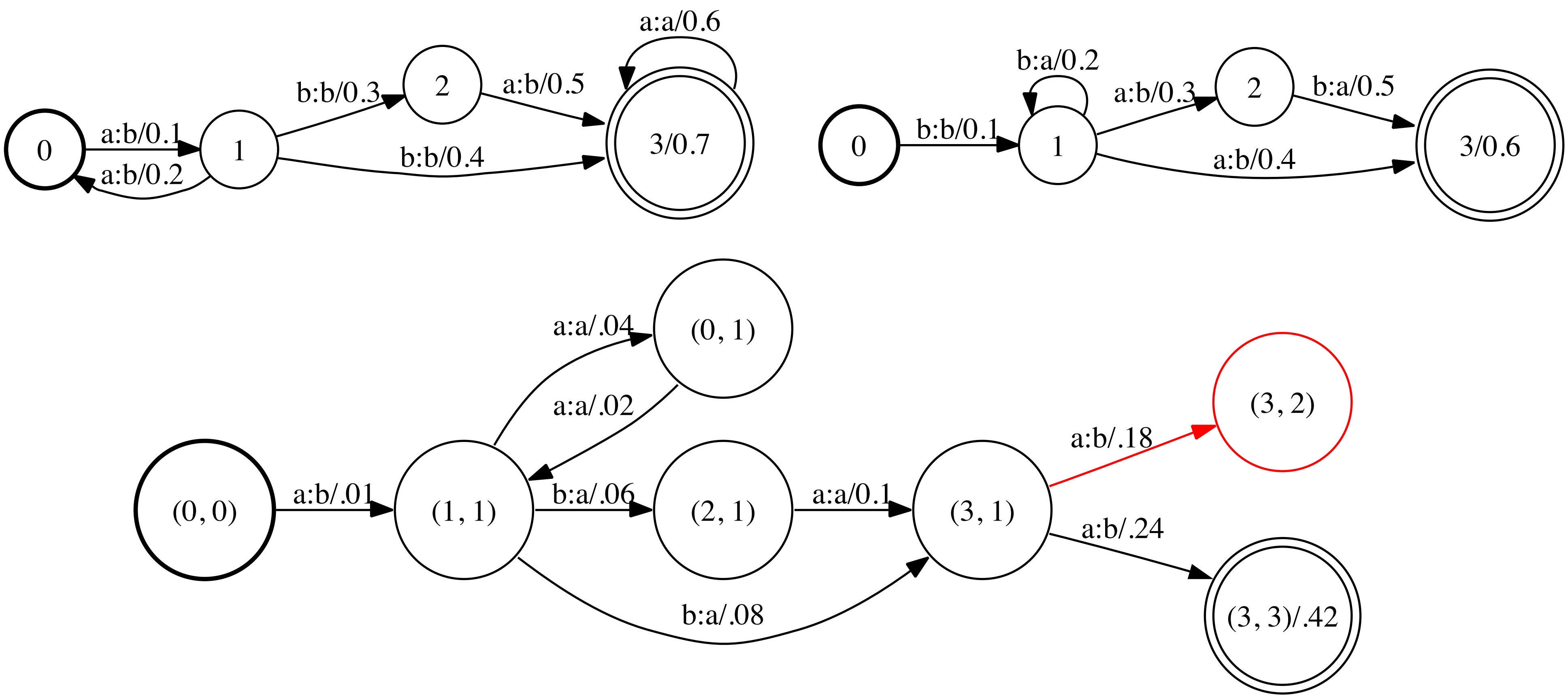
Composition

- Theorem: the composition of two weighted transducer is also a weighted transducer.
- Proof: constructive proof based on composition algorithm.
  - states identified with pairs.
  - ϵ-free case: transitions defined by

$$E = \biguplus_{\substack{(q_1, a, b, w_1, q_2) \in E_1 \\ (q_1', b, c, w_2, q_2') \in E_2}} \left\{ \Big((q_1, q_1'), a, c, w_1 \times w_2, (q_2, q_2') \Big) \right\}.$$

  - general case: use of intermediate ϵ-filter.

Composition Algorithm
ε-Free Case

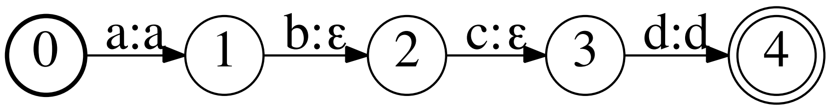


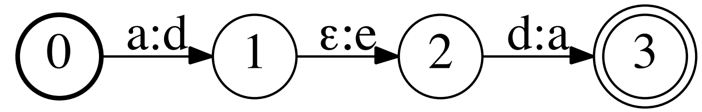
Complexity: $O(|T_1|\,|T_2|)$ in general, linear in some cases.

Redundant ε-Paths Problem

(MM, Pereira, and Riley, 1996; Pereira and Riley, 1997)

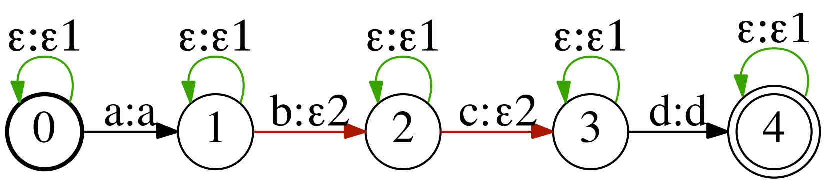
T_1

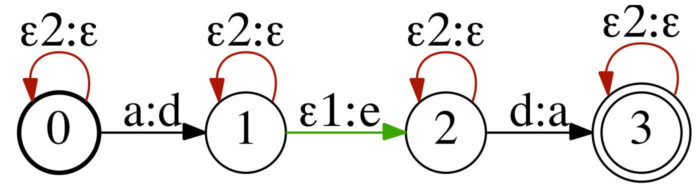




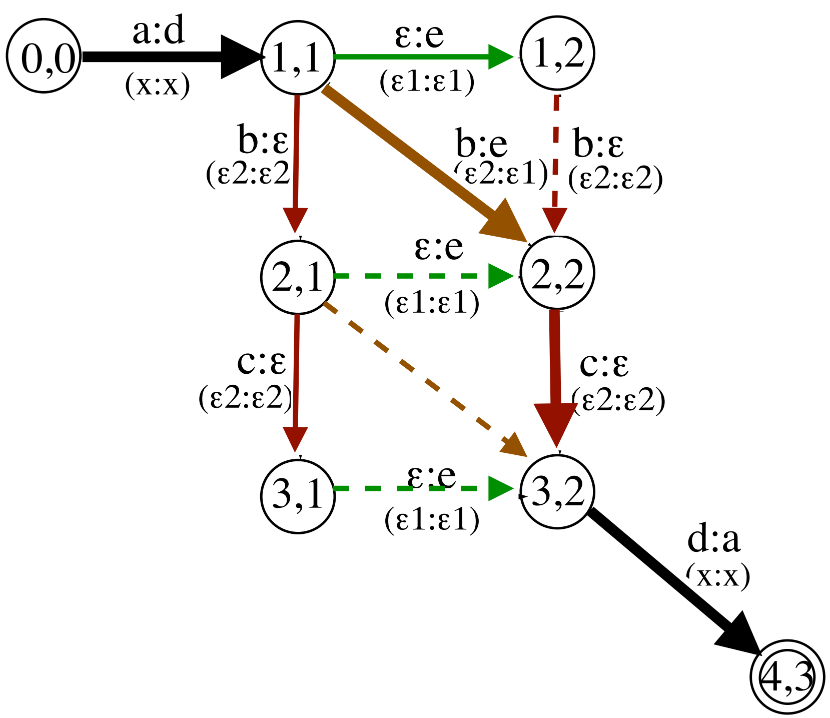
T_2

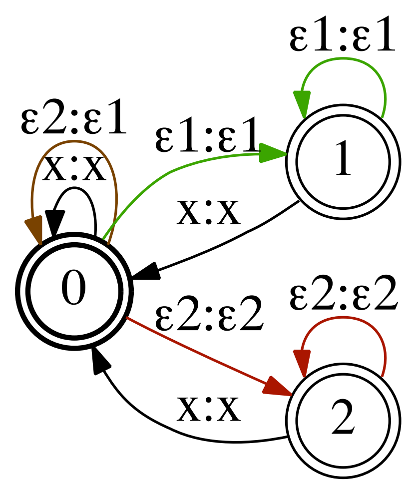
$\widetilde{T}_1$





$\widetilde{T}_2$





F

$$T = \widetilde{T}_1 \circ F \circ \widetilde{T}_2.$$

Kernels for Other Discrete Structures

- Similarly, PDS kernels can be defined on other discrete structures:
  - Images,
  - graphs,
  - parse trees,
  - automata,
  - weighted automata.

This Lecture

- Kernels
- Kernel-based algorithms
- Closure properties
- Sequence Kernels
- Negative kernels

Questions

- Gaussian kernels have the form $\exp(-d^2)$ where d is a metric.
  - for what other functions d does $\exp(-d^2)$ define a PDS kernel?
  - what other PDS kernels can we construct from a metric in a Hilbert space?

Negative Definite Kernels

(Schoenberg, 1938)

- Definition: A function $K\colon X \times X \to \mathbb{R}$ is said to be a negative definite symmetric (NDS) kernel if it is symmetric and if for all $\{x_1, \ldots, x_m\} \subseteq X$ and $\mathbf{c} \in \mathbb{R}^{m\times 1}$ with $\mathbf{1}^\top \mathbf{c} = 0$,

$$\mathbf{c}^\top \mathbf{K} \mathbf{c} \leq 0.$$

- Clearly, if K is PDS, then $-K$ is NDS, but the converse does not hold in general.

Examples

- The squared distance $||x-y||^2$ in a Hilbert space H defines an NDS kernel. If $\sum_{i=1}^m c_i = 0$,

$$\begin{aligned}
\sum_{i,j=1}^m c_i c_j ||\mathbf{x}_i - \mathbf{x}_j||^2 &= \sum_{i,j=1}^m c_i c_j (\mathbf{x}_i - \mathbf{x}_j) \cdot (\mathbf{x}_i - \mathbf{x}_j) \\
&= \sum_{i,j=1}^m c_i c_j (\|\mathbf{x}_i\|^2 + \|\mathbf{x}_j\|^2 - 2\mathbf{x}_i \cdot \mathbf{x}_j) \\
&= \sum_{i,j=1}^m c_i c_j (\|\mathbf{x}_i\|^2 + \|\mathbf{x}_j\|^2) - 2 \sum_{i=1}^m c_i \mathbf{x}_i \cdot \sum_{j=1}^m c_j \mathbf{x}_j \\
&\leq \sum_{i,j=1}^m c_i c_j (\|\mathbf{x}_i\|^2 + \|\mathbf{x}_j\|^2) \\
&= \sum_{j=1}^m c_j \Big(\sum_{i=1}^m c_i (\|\mathbf{x}_i\|^2 \Big) + \sum_{i=1}^m c_i \Big(\sum_{j=1}^m c_j \|\mathbf{x}_j\|^2 \Big) = 0.
\end{aligned}$$

NDS Kernels - Property

(Schoenberg, 1938)

- **Theorem**: Let $K\colon X \times X \to \mathbb{R}$ be an NDS kernel such that for all $x, y \in X$, $K(x, y) = 0$ iff $x = y$. Then, there exists a Hilbert space H and a mapping $\Phi\colon X \to H$ such that

$$\forall x, y \in X, \;\; K(x, y) = \|\Phi(x) - \Phi(y)\|^2.$$

Thus, under the hypothesis of the theorem, $\sqrt{K}$ defines a metric.

PDS and NDS Kernels

(Schoenberg, 1938)

- Theorem: let $K\colon X \times X \to \mathbb{R}$ be a symmetric kernel, then:
  - K is NDS iff $\exp(-tK)$ is a PDS kernel for all $t > 0$.
  - Let K' be defined for any x_0 by

$$K'(x, y) = K(x, x_0) + K(y, x_0) - K(x, y) - K(x_0, x_0)$$

for all $x, y \in X$. Then, K is NDS iff K' is PDS.

Example

- The kernel defined by $K(x, y) = \exp(-t||x - y||^2)$ is PDS for all $t > 0$ since $||x - y||^2$ is NDS.
- The kernel $\exp(-|x - y|^p)$ is not PDS for $p > 2$. Otherwise, for any $t > 0$, $\{x_1, \ldots, x_m\} \subseteq X$ and $\mathbf{c} \in \mathbb{R}^{m \times 1}$

$$\sum_{i,j=1}^{m} c_i c_j e^{-t|x_i - x_j|^p} = \sum_{i,j=1}^{m} c_i c_j e^{-|t^{1/p} x_i - t^{1/p} x_j|^p} \geq 0.$$

- This would imply that $|x - y|^p$ is NDS for $p > 2$, but that cannot be (see past homework assignments).

Conclusion

- PDS kernels:
  - rich mathematical theory and foundation.
  - general idea for extending many linear algorithms to non-linear prediction.
  - flexible method: any PDS kernel can be used.
  - widely used in modern algorithms and applications.
  - can we further learn a PDS kernel and a hypothesis based on that kernel from labeled data? (see tutorial: http://www.cs.nyu.edu/~mohri/icml2011-tutorial/).

References

- N. Aronszajn, Theory of Reproducing Kernels, *Trans. Amer. Math. Soc.*, 68, 337-404, 1950.
- Peter Bartlett and John Shawe-Taylor. Generalization performance of support vector machines and other pattern classifiers. In *Advances in kernel methods: support vector learning*, pages 43–54. MIT Press, Cambridge, MA, USA, 1999.
- Christian Berg, Jens Peter Reus Christensen, and Paul Ressel. *Harmonic Analysis on Semigroups*. Springer-Verlag: Berlin-New York, 1984.
- Bernhard Boser, Isabelle M. Guyon, and Vladimir Vapnik. *A training algorithm for optimal margin classifiers*. In proceedings of COLT 1992, pages 144-152, Pittsburgh, PA, 1992.
- Corinna Cortes, Patrick Haffner, and Mehryar Mohri. Rational Kernels: Theory and Algorithms. *Journal of Machine Learning Research (JMLR)*, 5:1035-1062, 2004.
- Corinna Cortes and Vladimir Vapnik, Support-Vector Networks, *Machine Learning*, 20, 1995.
- Kimeldorf, G. and Wahba, G. *Some results on Tchebycheffian Spline Functions*, J. Mathematical Analysis and Applications, 33, 1 (1971) 82-95.

References

- James Mercer. Functions of Positive and Negative Type, and Their Connection with the Theory of Integral Equations. In *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, Vol. 83, No. 559, pp. 69-70, 1909.

- Mehryar Mohri, Fernando C. N. Pereira, and Michael Riley. *Weighted Automata in Text and Speech Processing*, In *Proceedings of the 12th biennial European Conference on Artificial Intelligence (ECAI-96), Workshop on Extended finite state models of language*. Budapest, Hungary, 1996.

- Fernando C. N. Pereira and Michael D. Riley. Speech Recognition by Composition of Weighted Finite Automata. In Finite-State Language Processing, pages 431-453. MIT Press, 1997.

- I. J. Schoenberg, Metric Spaces and Positive Definite Functions. *Transactions of the American Mathematical Society*, Vol. 44, No. 3, pp. 522-536, 1938.

- Vladimir N. Vapnik. *Estimation of Dependences Based on Empirical Data*. Springer, Basederlin, 1982.

- Vladimir N. Vapnik. *The Nature of Statistical Learning Theory.* Springer, 1995.

- Vladimir N. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, New York, 1998.

Appendix

Mercer's Condition

(Mercer, 1909)

- **Theorem**: Let $X \times X$ be a compact subset of $\mathbb{R}^N$ and let $K : X \times X \to \mathbb{R}$ be in $L_\infty(X \times X)$ and symmetric. Then, K admits a uniformly convergent expansion

$$K(x, y) = \sum_{n=0}^{\infty} a_n \phi_n(x) \phi_n(y), \text{ with } a_n > 0,$$

iff for any function c in $L_2(X)$,

$$\int\int_{X \times X} c(x)c(y)K(x, y)dxdy \geq 0.$$

SVMs with PDS Kernels

- Constrained optimization:

Hadamard product

$$\max_{\boldsymbol{\alpha}} 2\, \mathbf{1}^\top \boldsymbol{\alpha} - (\boldsymbol{\alpha} \circ \mathbf{y})^\top \mathbf{K} (\boldsymbol{\alpha} \circ \mathbf{y})$$

$$\text{subject to: } \mathbf{0} \leq \boldsymbol{\alpha} \leq \mathbf{C} \wedge \boldsymbol{\alpha}^\top \mathbf{y} = 0.$$

- Solution:

$$h = \operatorname{sgn}\Big(\sum_{i=1}^{m} \alpha_i y_i K(x_i, \cdot) + b \Big),$$

with $b = y_i - (\boldsymbol{\alpha} \circ \mathbf{y})^\top \mathbf{K} \mathbf{e}_i$ for any x_i with $0 < \alpha_i < C.$

Foundations of Machine Learning
Boosting

Mehryar Mohri
Courant Institute and Google Research
mohri@cims.nyu.edu

Weak Learning

(Kearns and Valiant, 1994)

- Definition: concept class C is weakly PAC-learnable if there exists a (weak) learning algorithm L and $\gamma > 0$ such that:
  - for all $\delta > 0$, for all $c \in C$ and all distributions D,

$$\Pr_{S \sim D}\left[R(h_S) \leq \frac{1}{2} - \gamma\right] \geq 1 - \delta,$$

  - for samples S of size $m = poly(1/\delta)$ for a fixed polynomial.

Boosting Ideas

- Finding simple relatively accurate base classifiers often not hard $\longleftarrow$ weak learner.
- Main ideas:
  - use weak learner to create a strong learner.
  - combine base classifiers returned by weak learner (ensemble method).
- But, how should the base classifiers be combined?

AdaBoost

(Freund and Schapire, 1997)

$H \subseteq \{-1,+1\}^X$.

$\textsc{AdaBoost}(S=((x_1,y_1),\ldots,(x_m,y_m)))$

$\textbf{for } i \leftarrow 1 \textbf{ to } m \textbf{ do}$
$\quad D_1(i) \leftarrow \frac{1}{m}$
$\textbf{for } t \leftarrow 1 \textbf{ to } T \textbf{ do}$
$\quad h_t \leftarrow \text{base classifier in } H \text{ with small error } \epsilon_t = \Pr_{i \sim D_t}[h_t(x_i) \neq y_i]$
$\quad \alpha_t \leftarrow \frac{1}{2} \log \frac{1-\epsilon_t}{\epsilon_t}$
$\quad Z_t \leftarrow 2[\epsilon_t(1-\epsilon_t)]^{\frac{1}{2}} \quad \triangleright \text{ normalization factor}$
$\quad \textbf{for } i \leftarrow 1 \textbf{ to } m \textbf{ do}$
$\quad\quad D_{t+1}(i) \leftarrow \frac{D_t(i)\exp(-\alpha_t y_i h_t(x_i))}{Z_t}$
$\quad f_t \leftarrow \sum_{s=1}^{t} \alpha_s h_s$
$\textbf{return } h = \operatorname{sgn}(f_T)$

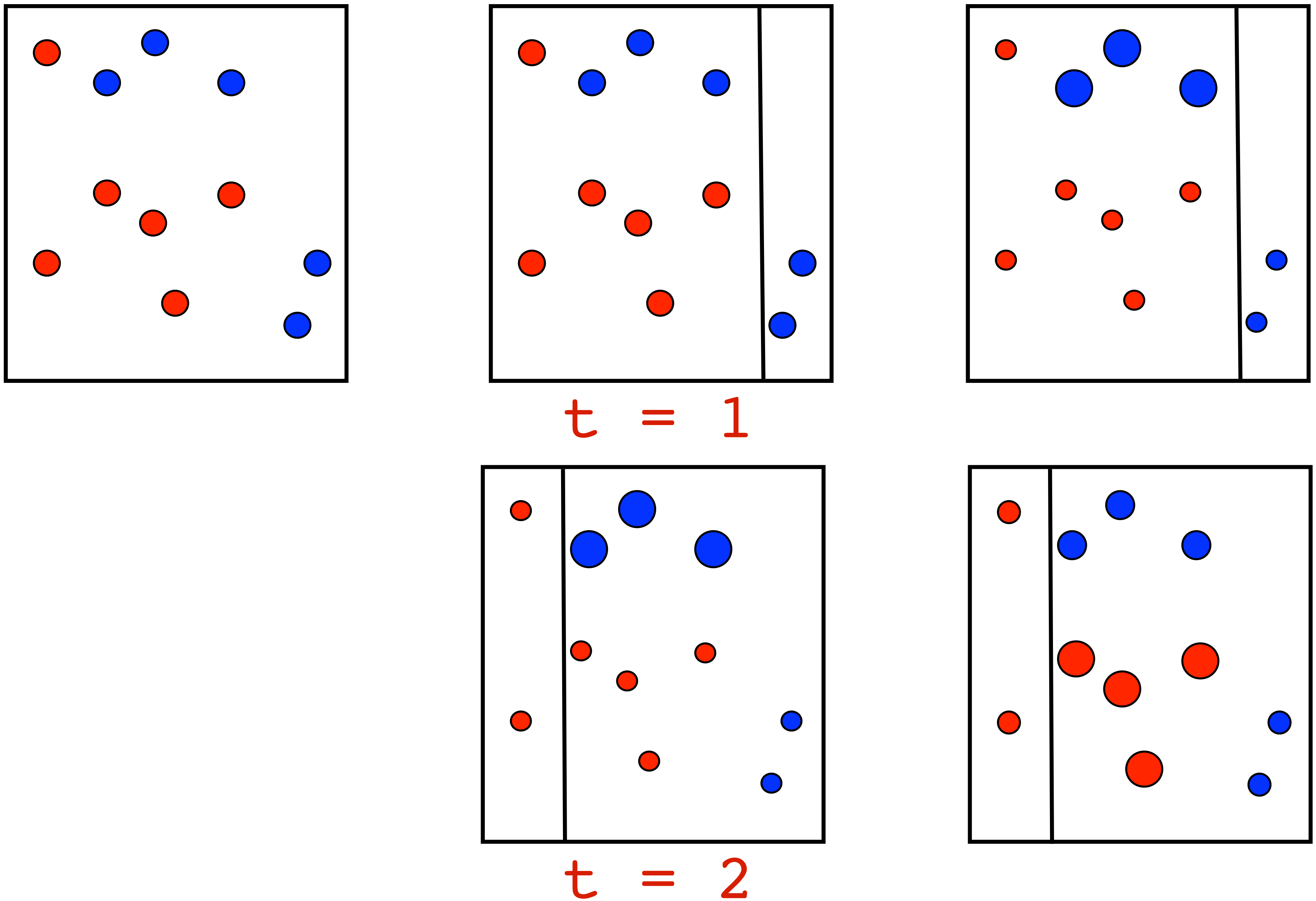
Notes

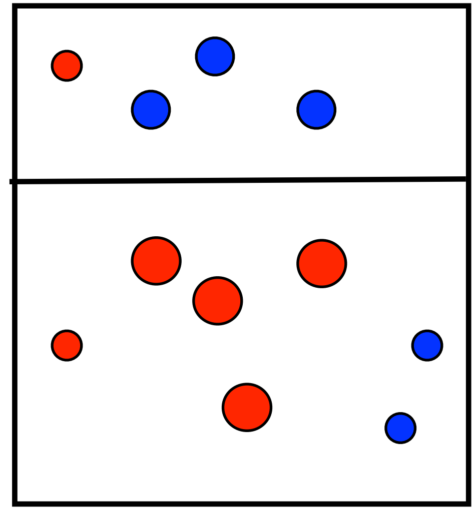
- Distributions D_t over training sample:
  - originally uniform.
  - at each round, the weight of a misclassified example is increased.
  - observation: $D_{t+1}(i) = \frac{e^{-y_i f_t(x_i)}}{m \prod_{s=1}^t Z_s}$, since

$$D_{t+1}(i) = \frac{D_t(i) e^{-\alpha_t y_i h_t(x_i)}}{Z_t} = \frac{D_{t-1}(i) e^{-\alpha_{t-1} y_i h_{t-1}(x_i)} e^{-\alpha_t y_i h_t(x_i)}}{Z_{t-1} Z_t} = \frac{1}{m} \frac{e^{-y_i \sum_{s=1}^t \alpha_s h_s(x_i)}}{\prod_{s=1}^t Z_s}.$$

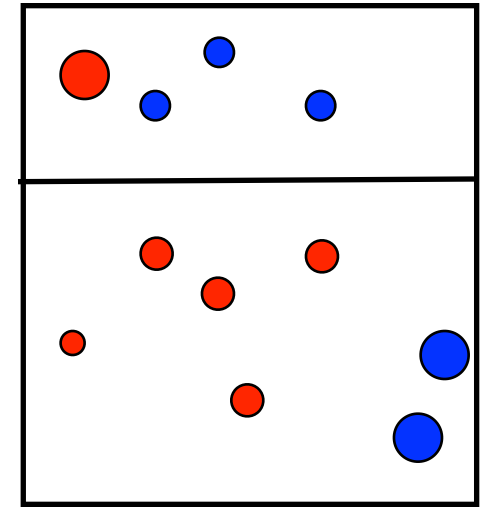
- Weight assigned to base classifier h_t: α_t directly depends on the accuracy of h_t at round t .

Illustration

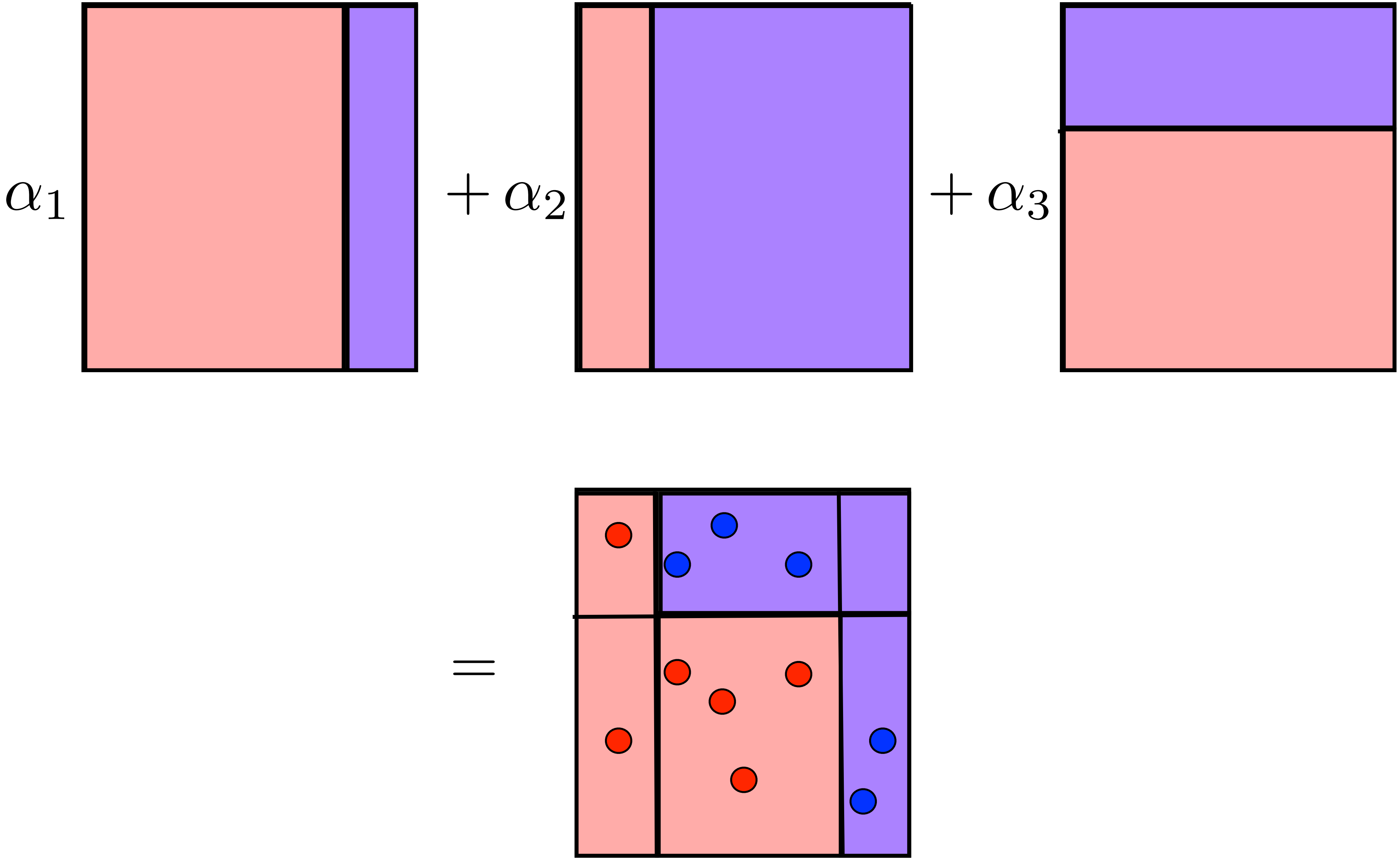




t = 3



. . .   . . .


α_1
$+\,\alpha_2$
$+\,\alpha_3$
$=$

Bound on Empirical Error

(Freund and Schapire, 1997)

- Theorem: The empirical error of the classifier output by AdaBoost verifies:

$$\widehat{R}(h) \leq \exp\left[-2\sum_{t=1}^{T}\left(\frac{1}{2}-\epsilon_t\right)^2\right].$$

  - If further for all $t \in [1, T]$, $\gamma \leq \left(\frac{1}{2} - \epsilon_t\right)$, then

$$\widehat{R}(h) \leq \exp(-2\gamma^2 T).$$

  - γ does not need to be known in advance: adaptive boosting.

- Proof: Since, as we saw, $D_{t+1}(i) = \frac{e^{-y_i f_t(x_i)}}{m \prod_{s=1}^{t} Z_s}$,

$$\widehat{R}(h) = \frac{1}{m} \sum_{i=1}^{m} 1_{y_i f(x_i) \leq 0} \leq \frac{1}{m} \sum_{i=1}^{m} \exp(-y_i f(x_i))$$

$$\leq \frac{1}{m} \sum_{i=1}^{m} \left[m \prod_{t=1}^{T} Z_t \right] D_{T+1}(i) = \prod_{t=1}^{T} Z_t.$$

- Now, since Z_t is a normalization factor,

$$\begin{aligned} Z_t &= \sum_{i=1}^{m} D_t(i) e^{-\alpha_t y_i h_t(x_i)} \\ &= \sum_{i: y_i h_t(x_i) \geq 0} D_t(i) e^{-\alpha_t} + \sum_{i: y_i h_t(x_i) < 0} D_t(i) e^{\alpha_t} \\ &= (1 - \epsilon_t) e^{-\alpha_t} + \epsilon_t e^{\alpha_t} \\ &= (1 - \epsilon_t) \sqrt{\frac{\epsilon_t}{1 - \epsilon_t}} + \epsilon_t \sqrt{\frac{1 - \epsilon_t}{\epsilon_t}} = 2\sqrt{\epsilon_t (1 - \epsilon_t)}. \end{aligned}$$

- Thus,

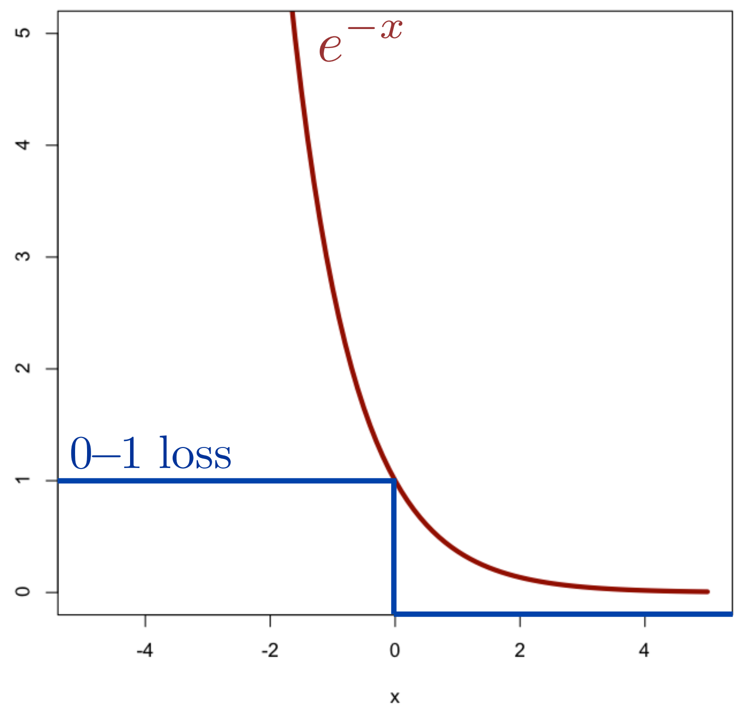
$$\prod_{t=1}^{T} Z_t = \prod_{t=1}^{T} 2\sqrt{\epsilon_t(1-\epsilon_t)} = \prod_{t=1}^{T} \sqrt{1 - 4\Big(\frac{1}{2} - \epsilon_t\Big)^2}$$
$$\leq \prod_{t=1}^{T} \exp\Big[-2\Big(\frac{1}{2} - \epsilon_t\Big)^2\Big] = \exp\Big[-2\sum_{t=1}^{T}\Big(\frac{1}{2} - \epsilon_t\Big)^2\Big].$$

- Notes:
  - α_t minimizer of $\alpha \mapsto (1-\epsilon_t)e^{-\alpha} + \epsilon_t e^{\alpha}$.
  - since $(1-\epsilon_t)e^{-\alpha_t} = \epsilon_t e^{\alpha_t}$, at each round, AdaBoost assigns the same probability mass to correctly classified and misclassified instances.
  - for base classifiers $x \mapsto [-1, +1]$, α_t can be similarly chosen to minimize Z_t.

AdaBoost = Coordinate Descent

- Objective Function: convex and differentiable.

$$F(\bar{\boldsymbol{\alpha}}) = \frac{1}{m}\sum_{i=1}^{m} e^{-y_i f(x_i)} = \frac{1}{m}\sum_{i=1}^{m} e^{-y_i \sum_{j=1}^{N} \bar{\alpha}_j h_j(x_i)}.$$



- Direction: unit vector $\mathbf{e}_k$ with best directional derivative:

$$F'(\bar{\boldsymbol{\alpha}}_{t-1}, \mathbf{e}_k) = \lim_{\eta \to 0} \frac{F(\bar{\boldsymbol{\alpha}}_{t-1} + \eta \mathbf{e}_k) - F(\bar{\boldsymbol{\alpha}}_{t-1})}{\eta}.$$

- Since $F(\bar{\boldsymbol{\alpha}}_{t-1} + \eta \mathbf{e}_k) = \frac{1}{m} \sum_{i=1}^{m} e^{-y_i \sum_{j=1}^{N} \bar{\alpha}_{t-1,j} h_j(x_i) - \eta y_i h_k(x_i)}$,

$$\begin{aligned}
F'(\bar{\boldsymbol{\alpha}}_{t-1}, \mathbf{e}_k) &= -\frac{1}{m} \sum_{i=1}^{m} y_i h_k(x_i) e^{-y_i \sum_{j=1}^{N} \bar{\alpha}_{t-1,j} h_j(x_i)} \\
&= -\frac{1}{m} \sum_{i=1}^{m} y_i h_k(x_i) \bar{D}_t(i) \bar{Z}_t \\
&= -\left[\sum_{i=1}^{m} \bar{D}_t(i) 1_{y_i h_k(x_i) = +1} - \sum_{i=1}^{m} \bar{D}_t(i) 1_{y_i h_k(x_i) = -1} \right] \frac{\bar{Z}_t}{m} \\
&= -\left[(1 - \bar{\epsilon}_{t,k}) - \bar{\epsilon}_{t,k} \right] \frac{\bar{Z}_t}{m} = \left[\boxed{2\bar{\epsilon}_{t,k} - 1} \right] \frac{\bar{Z}_t}{m}.
\end{aligned}$$

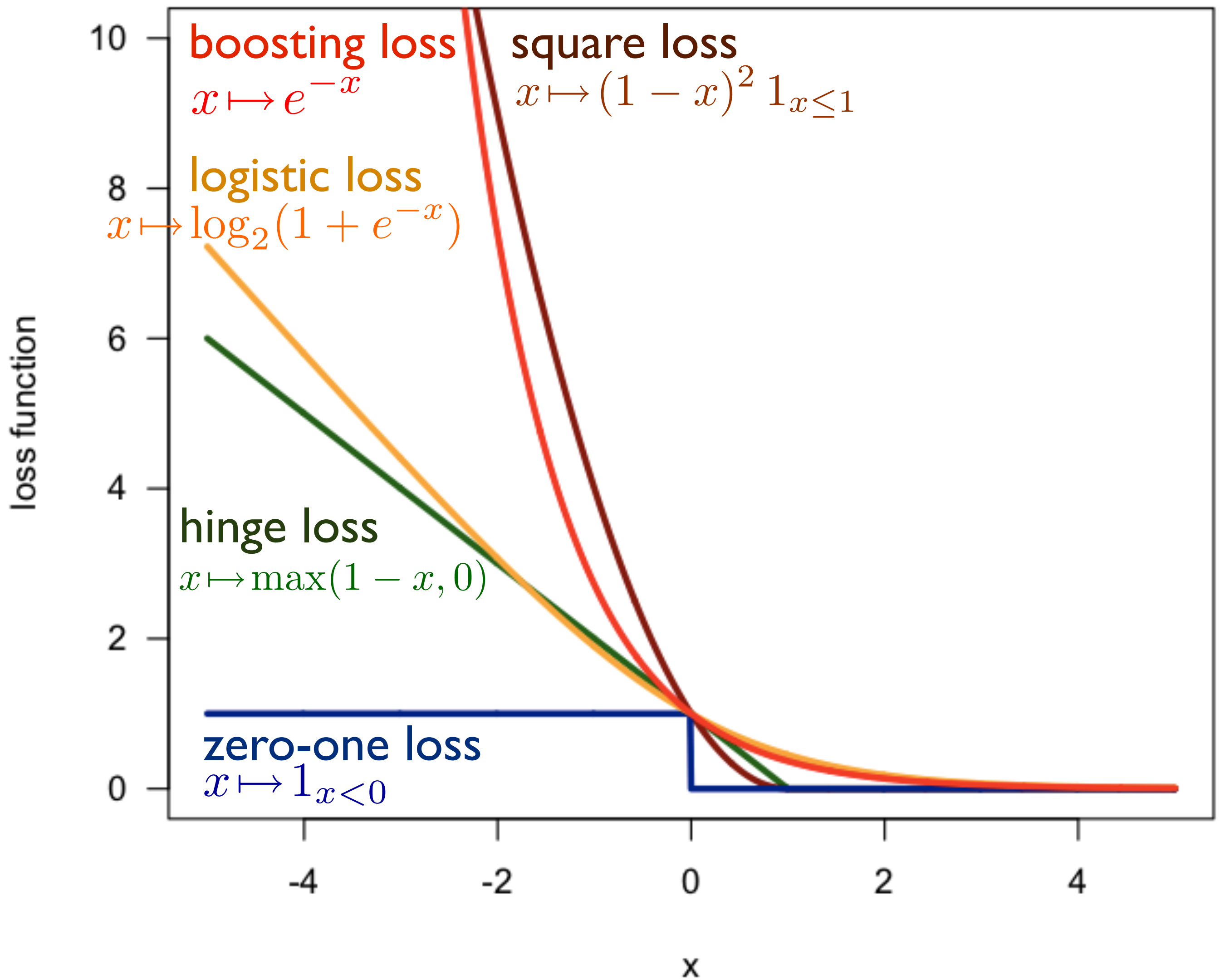
Thus, direction corresponding to base classifier with smallest error.

- **Step size**: η chosen to minimize $F(\bar{\boldsymbol{\alpha}}_{t-1} + \eta\, \mathbf{e}_k)$;

$$\begin{aligned}
\frac{dF(\bar{\boldsymbol{\alpha}}_{t-1} + \eta \mathbf{e}_k)}{d\eta} = 0 &\Leftrightarrow -\sum_{i=1}^{m} y_i h_k(x_i) e^{-y_i \sum_{j=1}^{N} \bar{\alpha}_{t-1,j} h_j(x_i)} e^{-\eta y_i h_k(x_i)} = 0 \\
&\Leftrightarrow -\sum_{i=1}^{m} y_i h_k(x_i) \bar{D}_t(i) \bar{Z}_t e^{-\eta y_i h_k(x_i)} = 0 \\
&\Leftrightarrow -\sum_{i=1}^{m} y_i h_k(x_i) \bar{D}_t(i) e^{-\eta y_i h_k(x_i)} = 0 \\
&\Leftrightarrow -\left[(1 - \bar{\epsilon}_{t,k}) e^{-\eta} - \bar{\epsilon}_{t,k} e^{\eta}\right] = 0 \\
&\Leftrightarrow \boxed{\eta = \frac{1}{2} \log \frac{1 - \bar{\epsilon}_{t,k}}{\bar{\epsilon}_{t,k}}.}
\end{aligned}$$

Thus, step size matches base classifier weight of AdaBoost.

Alternative Loss Functions



Standard Use in Practice

- Base learners: decision trees, quite often just decision stumps (trees of depth one).
- Boosting stumps:
  - data in $\mathbb{R}^N$, e.g., $N = 2$, $(\text{height}(x), \text{weight}(x))$.
  - associate a stump to each component.
  - pre-sort each component: $O(Nm \log m)$.
  - at each round, find best component and threshold.
  - total complexity: $O((m \log m)N + mNT)$.
  - stumps not weak learners: think XOR example!

Overfitting?

- Assume that $\mathrm{VCdim}(H) = d$ and for a fixed T, define

$$\mathcal{F}_T = \left\{ \mathrm{sgn}\left(\sum_{t=1}^{T} \alpha_t h_t - b \right) : \alpha_t, b \in \mathbb{R}, h_t \in H \right\}.$$

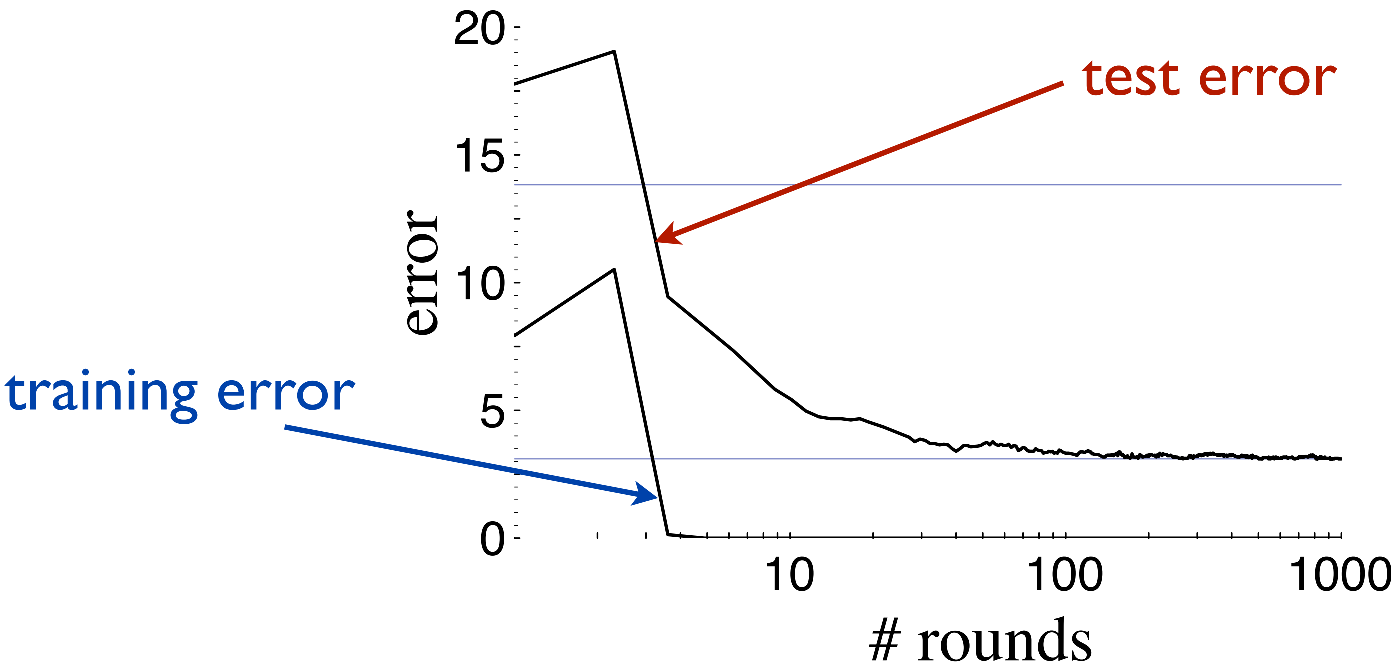
- $\mathcal{F}_T$ can form a very rich family of classifiers. It can be shown (Freund and Schapire, 1997) that:

$$\mathrm{VCdim}(\mathcal{F}_T) \leq 2(d+1)(T+1)\log_2((T+1)e).$$

- This suggests that AdaBoost could overfit for large values of T, and that is in fact observed in some cases, but in various others it is not!

Empirical Observations

- Several empirical observations (not all): AdaBoost does not seem to overfit, furthermore:



C4.5 decision trees (Schapire et al., 1998).

Rademacher Complexity of Convex Hulls

- **Theorem**: Let H be a set of functions mapping from X to $\mathbb{R}$. Let the convex hull of H be defined as

$$\text{conv}(H) = \{\sum_{k=1}^{p} \mu_k h_k : p \geq 1, \mu_k \geq 0, \sum_{k=1}^{p} \mu_k \leq 1, h_k \in H\}.$$

  Then, for any sample S, $\widehat{\mathfrak{R}}_S(\text{conv}(H)) = \widehat{\mathfrak{R}}_S(H)$.

- **Proof**: $\widehat{\mathfrak{R}}_S(\text{conv}(H)) = \dfrac{1}{m} \underset{\sigma}{\text{E}} \left[\sup_{h_k \in H, \boldsymbol{\mu} \geq 0, \|\boldsymbol{\mu}\|_1 \leq 1} \sum_{i=1}^{m} \sigma_i \sum_{k=1}^{p} \mu_k h_k(x_i) \right]$

$$= \frac{1}{m} \underset{\sigma}{\text{E}} \left[\sup_{h_k \in H} \sup_{\boldsymbol{\mu} \geq 0, \|\boldsymbol{\mu}\|_1 \leq 1} \sum_{k=1}^{p} \mu_k \Big(\sum_{i=1}^{m} \sigma_i h_k(x_i) \Big) \right]$$

$$= \frac{1}{m} \underset{\sigma}{\text{E}} \left[\sup_{h_k \in H} \max_{k \in [1,p]} \Big(\sum_{i=1}^{m} \sigma_i h_k(x_i) \Big) \right]$$

$$= \frac{1}{m} \underset{\sigma}{\text{E}} \left[\sup_{h \in H} \sum_{i=1}^{m} \sigma_i h(x_i) \right] = \widehat{\mathfrak{R}}_S(H).$$

Margin Bound - Ensemble Methods

(Koltchinskii and Panchenko, 2002)

- Corollary: Let H be a set of real-valued functions. Fix $\rho > 0$. For any $\delta > 0$, with probability at least $1-\delta$, the following holds for all $h \in \operatorname{conv}(H)$:

$$R(h) \leq \widehat{R}_\rho(h) + \frac{2}{\rho}\mathfrak{R}_m(H) + \sqrt{\frac{\log \frac{1}{\delta}}{2m}}$$

$$R(h) \leq \widehat{R}_\rho(h) + \frac{2}{\rho}\widehat{\mathfrak{R}}_S(H) + 3\sqrt{\frac{\log \frac{2}{\delta}}{2m}}.$$

- Proof: Direct consequence of margin bound of Lecture 4 and $\widehat{\mathfrak{R}}_S(\operatorname{conv}(H)) = \widehat{\mathfrak{R}}_S(H)$.

Margin Bound - Ensemble Methods

(Koltchinskii and Panchenko, 2002); see also (Schapire et al., 1998)

- **Corollary**: Let H be a family of functions taking values in $\{-1, +1\}$ with VC dimension d. Fix $\rho > 0$. For any $\delta > 0$, with probability at least $1 - \delta$, the following holds for all $h \in \mathrm{conv}(H)$:

$$R(h) \leq \widehat{R}_\rho(h) + \frac{2}{\rho}\sqrt{\frac{2d \log \frac{em}{d}}{m}} + \sqrt{\frac{\log \frac{1}{\delta}}{2m}}.$$

- **Proof**: Follows directly previous corollary and VC dimension bound on Rademacher complexity (see lecture 3).

Notes

- All of these bounds can be generalized to hold uniformly for all $\rho \in (0, 1)$, at the cost of an additional term $\sqrt{\frac{\log \log_2 \frac{2}{\rho}}{m}}$ and other minor constant factor changes (Koltchinskii and Panchenko, 2002).
- For AdaBoost, the bound applies to the functions
$$x \mapsto \frac{f(x)}{\|\boldsymbol{\alpha}\|_1} = \frac{\sum_{t=1}^T \alpha_t h_t(x)}{\|\boldsymbol{\alpha}\|_1} \in \operatorname{conv}(H).$$
- Note that T does not appear in the bound.

Margin Distribution

- **Theorem**: For any $\rho > 0$, the following holds:

$$\widehat{\Pr}\left[\frac{yf(x)}{\|\boldsymbol{\alpha}\|_1} \leq \rho\right] \leq 2^T \prod_{t=1}^{T} \sqrt{\epsilon_t^{1-\rho}(1-\epsilon_t)^{1+\rho}}.$$

- **Proof**: Using the identity $D_{t+1}(i) = \frac{e^{-y_i f(x_i)}}{m \prod_{t=1}^T Z_t}$,

$$\begin{aligned}
\frac{1}{m}\sum_{i=1}^{m} 1_{y_i f(x_i) - \|\boldsymbol{\alpha}\|_1 \rho \leq 0} &\leq \frac{1}{m}\sum_{i=1}^{m} \exp(-y_i f(x_i) + \|\boldsymbol{\alpha}\|_1 \rho) \\
&= \frac{1}{m}\sum_{i=1}^{m} e^{\|\boldsymbol{\alpha}\|_1 \rho}\left[m \prod_{t=1}^{T} Z_t\right] D_{T+1}(i) \\
&= e^{\|\boldsymbol{\alpha}\|_1 \rho} \prod_{t=1}^{T} Z_t = 2^T \prod_{t=1}^{T} \left[\sqrt{\tfrac{1-\epsilon_t}{\epsilon_t}}\right]^{\rho} \sqrt{\epsilon_t(1-\epsilon_t)}.
\end{aligned}$$

Notes

- If for all $t \in [1, T]$, $\gamma \leq (\frac{1}{2} - \epsilon_t)$, then the upper bound can be bounded by

$$\widehat{\Pr}\left[\frac{yf(x)}{\|\boldsymbol{\alpha}\|_1} \leq \rho\right] \leq \left[(1-2\gamma)^{1-\rho}(1+2\gamma)^{1+\rho}\right]^{T/2}.$$

  For $\rho < \gamma$, $(1-2\gamma)^{1-\rho}(1+2\gamma)^{1+\rho} < 1$ and the bound decreases exponentially in T.

- For the bound to be convergent: $\rho \gg O(1/\sqrt{m})$, thus $\gamma \gg O(1/\sqrt{m})$ is roughly the condition on the edge value.

L1-Geometric Margin

- Definition: the L_1-margin $\rho_f(x)$ of a linear function $f = \sum_{t=1}^{T} \alpha_t h_t$ with $\boldsymbol{\alpha} \neq 0$ at a point $x \in \mathcal{X}$ is defined by

$$\rho_f(x) = \frac{|f(x)|}{\|\boldsymbol{\alpha}\|_1} = \frac{|\sum_{t=1}^{T} \alpha_t h_t(x)|}{\|\boldsymbol{\alpha}\|_1} = \frac{\big|\boldsymbol{\alpha} \cdot \mathbf{h}(x)\big|}{\|\boldsymbol{\alpha}\|_1}.$$

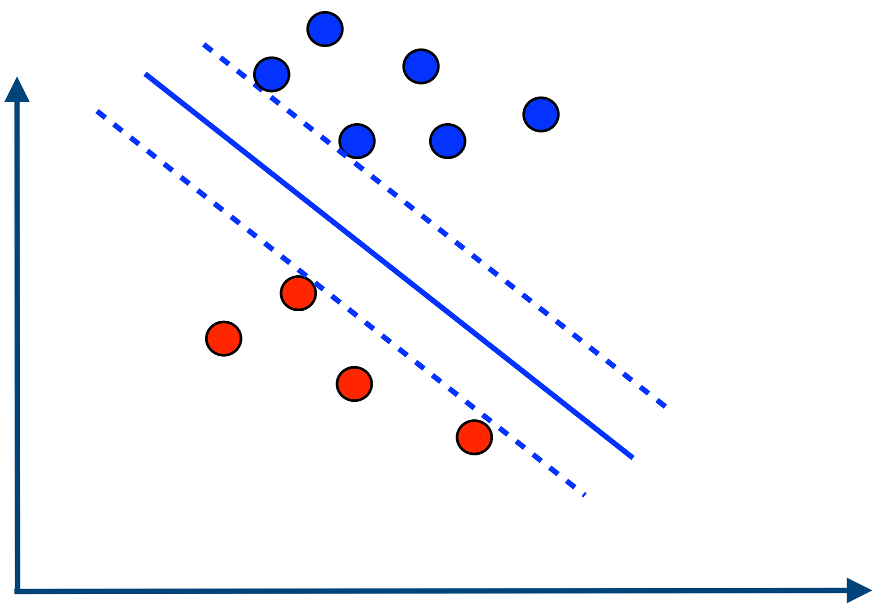
  - the L_1-margin of f over a sample $S = (x_1, \ldots, x_m)$ is its minimum margin at points in that sample:

$$\rho_f = \min_{i \in [1,m]} \rho_f(x_i) = \min_{i \in [1,m]} \frac{\big|\boldsymbol{\alpha} \cdot \mathbf{h}(x_i)\big|}{\|\boldsymbol{\alpha}\|_1}.$$

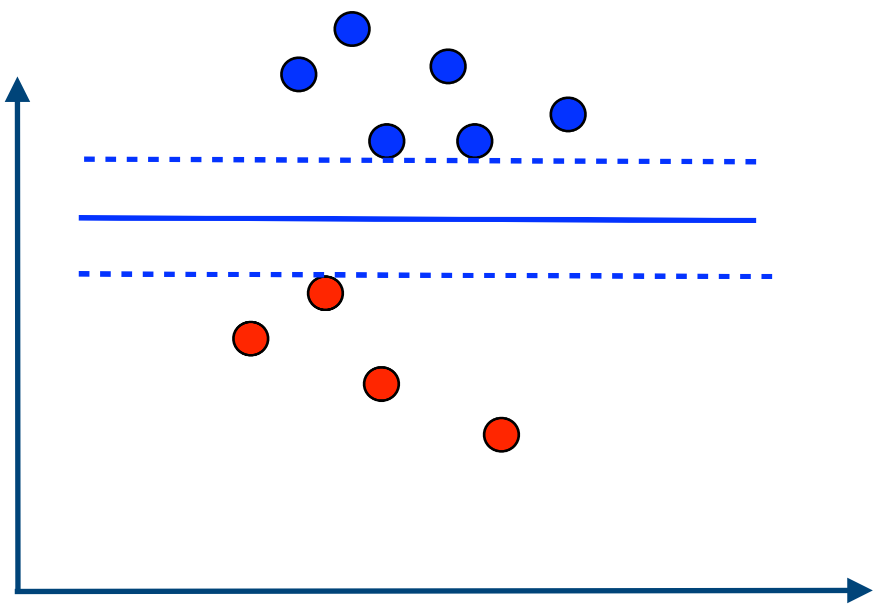
SVM vs AdaBoost

	SVM	AdaBoost
features or base hypotheses	$\mathbf{\Phi}(x) = \begin{bmatrix} \Phi_1(x) \\ \vdots \\ \Phi_N(x) \end{bmatrix}$	$\mathbf{h}(x) = \begin{bmatrix} h_1(x) \\ \vdots \\ h_N(x) \end{bmatrix}$
predictor	$x \mapsto \mathbf{w} \cdot \mathbf{\Phi}(x)$	$x \mapsto \boldsymbol{\alpha} \cdot \mathbf{h}(x)$
geom. margin	$\frac{\lvert \mathbf{w} \cdot \mathbf{\Phi}(x) \rvert}{\lVert \mathbf{w} \rVert_2} = d_2(\mathbf{\Phi}(x), \text{hyperpl.})$	$\frac{\lvert \boldsymbol{\alpha} \cdot \mathbf{h}(x) \rvert}{\lVert \boldsymbol{\alpha} \rVert_1} = d_\infty(\mathbf{h}(x), \text{hyperpl.})$
conf. margin	$y(\mathbf{w} \cdot \mathbf{\Phi}(x))$	$y(\boldsymbol{\alpha} \cdot \mathbf{h}(x))$
regularization	$\lVert \mathbf{w} \rVert_2$	$\lVert \boldsymbol{\alpha} \rVert_1$ (L1-AB)

Maximum-Margin Solutions



Norm $\|\cdot\|_2$.



Norm $\|\cdot\|_\infty$.

But, Does AdaBoost Maximize the Margin?

- No: AdaBoost may converge to a margin that is significantly below the maximum margin (Rudin et al., 2004) (e.g., 1/3 instead of 3/8)!
- Lower bound: AdaBoost can achieve asymptotically a margin that is at least $\frac{\rho_{\max}}{2}$ if the data is separable and some conditions on the base learners hold (Rätsch and Warmuth, 2002).
- Several boosting-type margin-maximization algorithms: but, performance in practice not clear or not reported.

AdaBoost's Weak Learning Condition

- Definition: the edge of a base classifier h_t for a distribution D over the training sample is

$$\gamma(t) = \frac{1}{2} - \epsilon_t = \frac{1}{2} \sum_{i=1}^{m} y_i h_t(x_i) D(i).$$

- Condition: there exists $\gamma > 0$ for any distribution D over the training sample and any base classifier

$$\gamma(t) \geq \gamma.$$

Zero-Sum Games

- Definition:
  - payoff matrix $\mathbf{M} = (\mathbf{M}_{ij}) \in \mathbb{R}^{m \times n}$.
  - m possible actions (pure strategy) for row player.
  - n possible actions for column player.
  - $\mathbf{M}_{ij}$ payoff for row player ($=$ loss for column player) when row plays i, column plays j.
- Example:

	rock	paper	scissors
rock	0	-1	1
paper	1	0	-1
scissors	-1	1	0

Mixed Strategies

(von Neumann, 1928)

- Definition: player row selects a distribution $\mathbf{p}$ over the rows, player column a distribution $\mathbf{q}$ over columns. The expected payoff for row is

$$\mathop{\mathrm{E}}_{\substack{i \sim p \\ j \sim q}}[\mathbf{M}_{ij}] = \sum_{i=1}^{m} \sum_{j=1}^{n} p_i \mathbf{M}_{ij} q_j = \mathbf{p}^\top \mathbf{M} \mathbf{q}.$$

- von Neumann's minimax theorem:

$$\max_{\mathbf{p}} \min_{\mathbf{q}} \mathbf{p}^\top \mathbf{M} \mathbf{q} = \min_{\mathbf{q}} \max_{\mathbf{p}} \mathbf{p}^\top \mathbf{M} \mathbf{q}.$$

  - equivalent form:

$$\max_{\mathbf{p}} \min_{j \in [1,n]} \mathbf{p}^\top \mathbf{M} \mathbf{e}_j = \min_{\mathbf{q}} \max_{i \in [1,m]} \mathbf{e}_i^\top \mathbf{M} \mathbf{q}.$$

John von Neumann (1903 - 1957)

AdaBoost and Game Theory

- Game:
  - Player A: selects point x_i, $i \in [1, m]$.
  - Player B: selects base hypothesis h_t, $t \in [1, T]$.
  - Payoff matrix $\mathbf{M} \in \{-1, +1\}^{m \times T}$: $\mathbf{M}_{it} = y_i h_t(x_i)$.
- von Neumann's theorem: assume finite H.

$$2\gamma^* = \min_D \max_{h \in H} \sum_{i=1}^m D(i) y_i h(x_i) = \max_{\boldsymbol{\alpha}} \min_{i \in [1,m]} y_i \sum_{t=1}^T \frac{\alpha_t h_t(x_i)}{\|\boldsymbol{\alpha}\|_1} = \rho^*.$$

Consequences

- Weak learning condition $\Longrightarrow$ non-zero margin.
  - thus, possible to search for non-zero margin.
  - AdaBoost $=$ (suboptimal) search for corresponding α; achieves at least half of the maximum margin.
- Weak learning $=$ strong condition:
  - the condition implies linear separability with margin $2\gamma^* > 0$.

Linear Programming Problem

- Maximizing the margin:

$$\rho = \max_{\boldsymbol{\alpha}} \min_{i \in [1,m]} y_i \frac{(\boldsymbol{\alpha} \cdot \mathbf{x}_i)}{||\alpha||_1}.$$

- This is equivalent to the following convex optimization LP problem:

$$\max_{\boldsymbol{\alpha}} \rho$$

$$\text{subject to : } y_i(\boldsymbol{\alpha} \cdot \mathbf{x}_i) \geq \rho$$

$$\|\boldsymbol{\alpha}\|_1 = 1.$$

- Note that:

$$\frac{|\boldsymbol{\alpha} \cdot \mathbf{x}|}{\|\boldsymbol{\alpha}\|_1} = \|\mathbf{x} - H\|_\infty, \text{ with } H = \{\mathbf{x} \colon \boldsymbol{\alpha} \cdot \mathbf{x} = 0\}.$$

Advantages of AdaBoost

- Simple: straightforward implementation.
- Efficient: complexity $O(mNT)$ for stumps:
  - when N and T are not too large, the algorithm is quite fast.
- Theoretical guarantees: but still many questions.
  - AdaBoost not designed to maximize margin.
  - regularized versions of AdaBoost.

Outliers

- AdaBoost assigns larger weights to harder examples.
- Application:
  - Detecting mislabeled examples.
  - Dealing with noisy data: regularization based on the average weight assigned to a point (soft margin idea for boosting) (Meir and Rätsch, 2003).

Weaker Aspects

- Parameters:
  - need to determine T, the number of rounds of boosting: stopping criterion.
  - need to determine base learners: risk of overfitting or low margins.
- Noise: severely damages the accuracy of Adaboost (Dietterich, 2000).

Other Boosting Algorithms

- arc-gv (Breiman, 1996): designed to maximize the margin, but outperformed by AdaBoost in experiments (Reyzin and Schapire, 2006).
- L1-regularized AdaBoost (Raetsch et al., 2001): outperfoms AdaBoost in experiments (Cortes et al., 2014).
- DeepBoost (Cortes et al., 2014): more favorable learning guarantees, outperforms both AdaBoost and L1-regularized AdaBoost in experiments.

References

- Corinna Cortes, Mehryar Mohri, and Umar Syed. Deep boosting. In *ICML*, pages 262-270, 2014.

- Leo Breiman. Bagging predictors. *Machine Learning*, 24(2): 123-140, 1996.

- Thomas G. Dietterich. An experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting, and randomization. *Machine Learning*, 40(2):139-158, 2000.

- Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119-139, 1997.

- G. Lebanon and J. Lafferty. Boosting and maximum likelihood for exponential models. In *NIPS*, pages 447–454, 2001.

- Ron Meir and Gunnar Rätsch. An introduction to boosting and leveraging. In *Advanced Lectures on Machine Learning (LNAI2600)*, 2003.

- J. von Neumann. Zur Theorie der Gesellschaftsspiele. Mathematische Annalen, 100:295-320, 1928.

References

- Cynthia Rudin, Ingrid Daubechies and Robert E. Schapire. The dynamics of AdaBoost: Cyclic behavior and convergence of margins. *Journal of Machine Learning Research*, 5: 1557-1595, 2004.
- Rätsch, G., and Warmuth, M. K. (2002) "Maximizing the Margin with Boosting", in *Proceedings of the 15th Annual Conference on Computational Learning Theory (COLT 02)*, Sidney, Australia, pp. 334–350, July 2002.
- Reyzin, Lev and Schapire, Robert E. How boosting the margin can also boost classifier complexity. In *ICML*, pages 753-760, 2006.
- Robert E. Schapire. The boosting approach to machine learning: An overview. In D. D. Denison, M. H. Hansen, C. Holmes, B. Mallick, B. Yu, editors, *Nonlinear Estimation and Classification*. Springer, 2003.
- Robert E. Schapire and Yoav Freund. *Boosting, Foundations and Algorithms*. The MIT Press, 2012.
- Robert E. Schapire, Yoav Freund, Peter Bartlett and Wee Sun Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. *The Annals of Statistics*, 26(5):1651-1686, 1998.

Foundations of Machine Learning

Maximum Entropy Models, Logistic Regression

Mehryar Mohri
Courant Institute and Google Research
mohri@cims.nyu.edu

Motivation

- Probabilistic models:
  - density estimation.
  - classification.

This Lecture

- Notions of information theory.
- Introduction to density estimation.
- Maxent models.
- Conditional Maxent models.

Entropy

(Shannon, 1948)

- Definition: the entropy of a discrete random variable X with probability mass distribution $\mathsf{p}(x) = \Pr[X = x]$ is

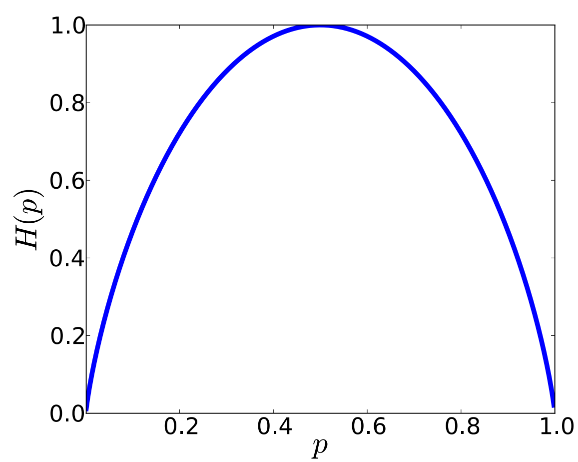
$$H(X) = -\mathrm{E}[\log \mathsf{p}(X)] = -\sum_{x \in X} \mathsf{p}(x) \log \mathsf{p}(x).$$

- Properties:
  - $H(X) \geq 0$.
  - measure of uncertainty of X.
  - maximal for uniform distribution. For a finite support, by Jensen's inequality:

$$H(X) = \mathrm{E}\left[\log \frac{1}{\mathsf{p}(X)}\right] \leq \log \mathrm{E}\left[\frac{1}{\mathsf{p}(X)}\right] = \log N.$$

Entropy

- Base of logarithm: not critical; for base 2, $-\log_2(\mathsf{p}(x))$ is the number of bits needed to represent $\mathsf{p}(x)$.
- Definition and notation: the entropy of a distribution p is defined by the same quantity and denoted by $H(\mathsf{p})$.
- Special case of Rényi entropy (Rényi, 1961).
- Binary entropy: $H(\mathsf{p}) = -\mathsf{p}\log\mathsf{p} - (1-\mathsf{p})\log(1-\mathsf{p})$.



Relative Entropy

(Shannon, 1948; Kullback and Leibler, 1951)

- Definition: the relative entropy (or Kullback-Leibler divergence) between two distributions p and q (discrete case) is

$$D(\mathsf{p} \parallel \mathsf{q}) = \mathrm{E}_{\mathsf{p}}\left[\log \frac{\mathsf{p}(X)}{\mathsf{q}(X)}\right] = \sum_{x \in \mathcal{X}} \mathsf{p}(x) \log \frac{\mathsf{p}(x)}{\mathsf{q}(x)},$$

with $0 \log \frac{0}{\mathsf{q}} = 0$ and $\mathsf{p} \log \frac{\mathsf{p}}{0} = +\infty.$

- Properties:
  - asymmetric: in general, $D(\mathsf{p} \parallel \mathsf{q}) \neq D(\mathsf{q} \parallel \mathsf{p})$ for $\mathsf{p} \neq \mathsf{q}$.
  - non-negative: $D(\mathsf{p} \parallel \mathsf{q}) \geq 0$ for all p and q.
  - definite: $(D(\mathsf{p} \parallel \mathsf{q}) = 0) \Rightarrow (\mathsf{p} = \mathsf{q})$.

Non-Negativity of Rel. Entropy

- By the concavity of log and Jensen's inequality,

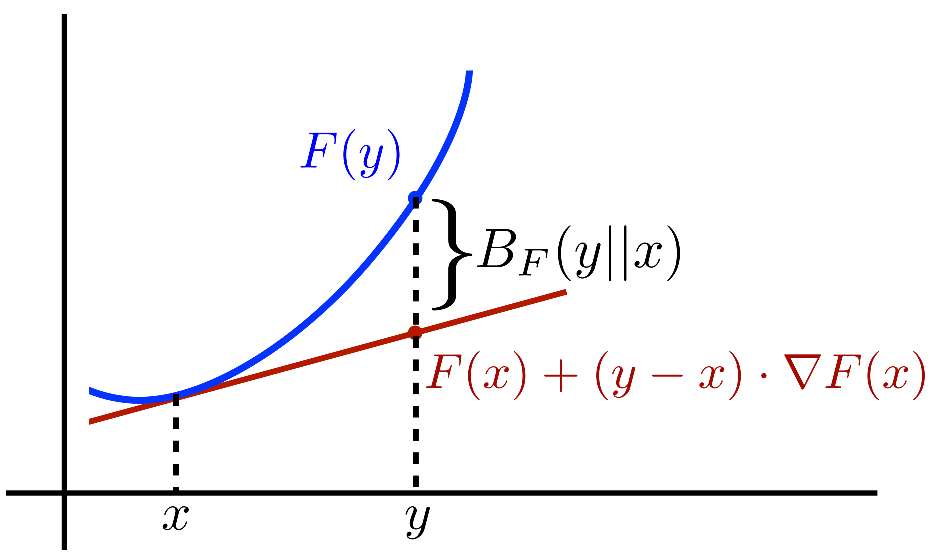
$$
\begin{aligned}
-D(\mathsf{p} \parallel \mathsf{q}) &= \sum_{x:\ \mathsf{p}(x)>0} \mathsf{p}(x) \log\left(\frac{\mathsf{q}(x)}{\mathsf{p}(x)}\right) \\
&\leq \log\left(\sum_{x:\ \mathsf{p}(x)>0} \mathsf{p}(x)\frac{\mathsf{q}(x)}{\mathsf{p}(x)}\right) \\
&= \log\left(\sum_{x:\ \mathsf{p}(x)>0} \mathsf{q}(x)\right) \leq \log(1) = 0.
\end{aligned}
$$

Bregman Divergence

(Bregman, 1967)

- Definition: let F be a convex and differentiable function defined over a convex set C in a Hilbert space $\mathbb{H}$. Then, the Bregman divergence B_F associated to F is defined by

$$B_F(x \parallel y) = F(x) - F(y) - \langle \nabla F(y), x - y \rangle .$$



Bregman Divergence

- Examples:

	$B_F(x \parallel y)$	$F(x)$
Squared L_2-distance	$\|\mathbf{x} - \mathbf{y}\|^2$	$\|\mathbf{x}\|^2$
Mahalanobis distance	$(\mathbf{x} - \mathbf{y})^\top \mathbf{K}^{-1}(\mathbf{x} - \mathbf{y})$	$\mathbf{x}^\top \mathbf{K}^{-1} \mathbf{x}$
Unnormalized relative entropy	$\widetilde{D}(\mathbf{x} \parallel \mathbf{y})$	$\sum_{i \in I} x_i \log x_i - x_i$

  - note: relative entropy not a Bregman divergence since not defined over an open set; but, on the simplex, coincides with unnormalized relative entropy

$$\widetilde{D}(\mathsf{p} \parallel \mathsf{q}) = \sum_{x \in \mathcal{X}} \mathsf{p}(x) \log \left[\frac{\mathsf{p}(x)}{\mathsf{q}(x)} \right] + \big(\mathsf{q}(x) - \mathsf{p}(x) \big).$$

Conditional Relative Entropy

- Definition: let p and q be two probability distributions over $\mathcal{X} \times \mathcal{Y}$. Then, the conditional relative entropy of p and q with respect to distribution r over $\mathcal{X}$ is defined by

$$\begin{aligned}\underset{X\sim\mathsf{r}}{\mathrm{E}}\Big[D\big(\mathsf{p}(\cdot|X)\parallel\mathsf{q}(\cdot|X)\big)\Big] &= \sum_{x\in\mathcal{X}}\mathsf{r}(x)\sum_{y\in\mathcal{Y}}\mathsf{p}(y|x)\log\frac{\mathsf{p}(y|x)}{\mathsf{q}(y|x)}\\ &= D(\widetilde{\mathsf{p}}\parallel\widetilde{\mathsf{q}}),\end{aligned}$$

  with $\widetilde{\mathsf{p}}(x,y)=\mathsf{r}(x)\mathsf{p}(y|x)$, $\widetilde{\mathsf{q}}(x,y)=\mathsf{r}(x)\mathsf{q}(y|x)$, and the conventions $0\log 0=0$, $0\log\frac{0}{0}=0$, and $\mathsf{p}\log\frac{\mathsf{p}}{0}=+\infty$.

  - note: the definition of conditional relative entropy is not intrinsic, it depends on a third distribution r.

This Lecture

- Notions of information theory.
- Introduction to density estimation.
- Maxent models.
- Conditional Maxent models.

Density Estimation Problem

- Training data: sample S of size m drawn i.i.d. from set $\mathcal{X}$ according to some distribution $\mathcal{D}$,

$$S = (x_1, \ldots, x_m).$$

- Problem: find distribution p out of hypothesis set $\mathcal{P}$ that best estimates $\mathcal{D}$.

Maximum Likelihood Solution

- Maximum Likelihood principle: select distribution $\mathsf{p} \in \mathcal{P}$ maximizing likelihood of observed sample S,

$$\begin{aligned}
\mathsf{p}_{\mathrm{ML}} &= \operatorname*{argmax}_{\mathsf{p}\in\mathcal{P}} \Pr[S|\mathsf{p}] \\
&= \operatorname*{argmax}_{\mathsf{p}\in\mathcal{P}} \prod_{i=1}^{m} \mathsf{p}(x_i) \\
&= \operatorname*{argmax}_{\mathsf{p}\in\mathcal{P}} \sum_{i=1}^{m} \log \mathsf{p}(x_i).
\end{aligned}$$

Relative Entropy Formulation

- Lemma: let $\widehat{\mathsf{p}}_S$ be the empirical distribution for sample S, then

$$\mathsf{p}_{\mathrm{ML}} = \operatorname*{argmin}_{\mathsf{p} \in \mathcal{P}} D(\widehat{\mathsf{p}}_S \parallel \mathsf{p}).$$

- Proof:

$$\begin{aligned}
D(\widehat{\mathsf{p}}_S \parallel \mathsf{p}) &= \sum_x \widehat{\mathsf{p}}_S(x) \log \widehat{\mathsf{p}}_S(x) - \sum_x \widehat{\mathsf{p}}_S(x) \log \mathsf{p}(x) \\
&= -H(\widehat{\mathsf{p}}_S) - \sum_x \frac{\sum_{i=1}^m 1_{x=x_i}}{m} \log \mathsf{p}(x) \\
&= -H(\widehat{\mathsf{p}}_S) - \sum_{i=1}^m \sum_x \frac{1_{x=x_i}}{m} \log \mathsf{p}(x) \\
&= -H(\widehat{\mathsf{p}}_S) - \sum_{i=1}^m \frac{\log \mathsf{p}(x_i)}{m}.
\end{aligned}$$

Maximum a Posteriori (MAP)

- Maximum a Posteriori principle: select distribution $\mathsf{p} \in \mathcal{P}$ that is the most likely, given the observed sample S and assuming a prior distribution $\Pr[\mathsf{p}]$ over $\mathcal{P}$,

$$\begin{aligned}
\mathsf{p}_{\mathrm{MAP}} &= \operatorname*{argmax}_{\mathsf{p} \in \mathcal{P}} \Pr[\mathsf{p}|S] \\
&= \operatorname*{argmax}_{\mathsf{p} \in \mathcal{P}} \frac{\Pr[S|\mathsf{p}] \Pr[\mathsf{p}]}{\Pr[S]} \\
&= \operatorname*{argmax}_{\mathsf{p} \in \mathcal{P}} \Pr[S|\mathsf{p}] \Pr[\mathsf{p}].
\end{aligned}$$

  - note: for a uniform prior, ML = MAP.

This Lecture

- Notions of information theory.
- Introduction to density estimation.
- Maxent models.
- Conditional Maxent models.

Density Estimation + Features

- Training data: sample S of size m drawn i.i.d. from set $\mathcal{X}$ according to some distribution $\mathcal{D}$,

$$S = (x_1, \ldots, x_m).$$

- Features: associated to elements of $\mathcal{X}$,

$$\begin{aligned} \mathbf{\Phi} \colon \mathcal{X} &\to \mathbb{R}^N \\ x &\mapsto \mathbf{\Phi}(x) = \begin{bmatrix} \Phi_1(x) \\ \vdots \\ \Phi_N(x) \end{bmatrix}. \end{aligned}$$

- Problem: find distribution p out of hypothesis set $\mathcal{P}$ that best estimates $\mathcal{D}$.
  - for simplicity, in what follows, $\mathcal{X}$ is assumed to be finite.

Features

- Feature functions Φ_j assumed to be in H and $\|\mathbf{\Phi}\|_\infty \leq \Lambda$.
- Examples of H:
  - family of threshold functions $\{\mathbf{x} \mapsto 1_{x_i \leq \theta} \colon \mathbf{x} \in \mathbb{R}^N, \theta \in \mathbb{R}\}$ defined over N variables.
  - functions defined via decision trees with larger depths.
  - k-degree monomials of the original features.
  - zero-one features (often used in NLP, e.g., presence/absence of a word or POS tag).

Maximum Entropy Principle

(E. T. Jaynes, 1957, 1983)

- Idea: empirical feature vector average close to expectation. For any $\delta > 0$, with probability at least $1 - \delta$

$$\left\| \underset{x \sim \mathcal{D}}{\mathrm{E}}[\mathbf{\Phi}(x)] - \underset{x \sim \widehat{\mathcal{D}}}{\mathrm{E}}[\mathbf{\Phi}(x)] \right\|_\infty \leq 2\mathfrak{R}_m(H) + \Lambda \sqrt{\frac{\log \frac{2}{\delta}}{2m}},$$

- Maxent principle: find distribution p that is closest to a prior distribution p_0 (typically uniform distribution) while verifying $\left\| \mathrm{E}_{x \sim \mathsf{p}}[\mathbf{\Phi}(x)] - \mathrm{E}_{x \sim \widehat{\mathcal{D}}}[\mathbf{\Phi}(x)] \right\|_\infty \leq \beta$.
- Closeness is measured using relative entropy.
  - note: no set $\mathcal{P}$ needed to be specified.

Maxent Formulation

- Optimization problem:

$$\min_{\mathsf{p} \in \Delta} D(\mathsf{p} \parallel \mathsf{p}_0)$$

$$\text{subject to: } \Big\| \mathop{\mathrm{E}}_{x \sim \mathsf{p}}[\mathbf{\Phi}(x)] - \mathop{\mathrm{E}}_{x \sim S}[\mathbf{\Phi}(x)] \Big\|_\infty \leq \beta.$$

  - convex optimization problem, unique solution.
  - $\beta = 0$: standard Maxent (or unregularized Maxent).
  - $\beta > 0$: regularized Maxent.

Relation with Entropy

- Relationship with entropy: for a uniform prior p_0,

$$
\begin{aligned}
D(\mathsf{p} \parallel \mathsf{p}_0) &= \sum_{x \in \mathcal{X}} \mathsf{p}(x) \log \frac{\mathsf{p}(x)}{\mathsf{p}_0(x)} \\
&= -\sum_{x \in \mathcal{X}} \mathsf{p}(x) \log \mathsf{p}_0(x) + \sum_{x \in \mathcal{X}} \mathsf{p}(x) \log \mathsf{p}(x) \\
&= \log |\mathcal{X}| - H(\mathsf{p}).
\end{aligned}
$$

Maxent Problem

- Optimization: convex optimization problem.

$$
\begin{aligned}
\min_{\mathsf{p}} \quad & \sum_{x \in \mathcal{X}} \mathsf{p}(x) \log \mathsf{p}(x) \\
\text{subject to:} \quad & \mathsf{p}(x) \geq 0, \forall x \in \mathcal{X} \\
& \sum_{x \in \mathcal{X}} \mathsf{p}(x) = 1 \\
& \left| \sum_{x \in \mathcal{X}} \mathsf{p}(x) \Phi_j(x) - \frac{1}{m} \sum_{i=1}^{m} \Phi_j(x_i) \right| \leq \beta, \forall j \in [1, N].
\end{aligned}
$$

Gibbs Distributions

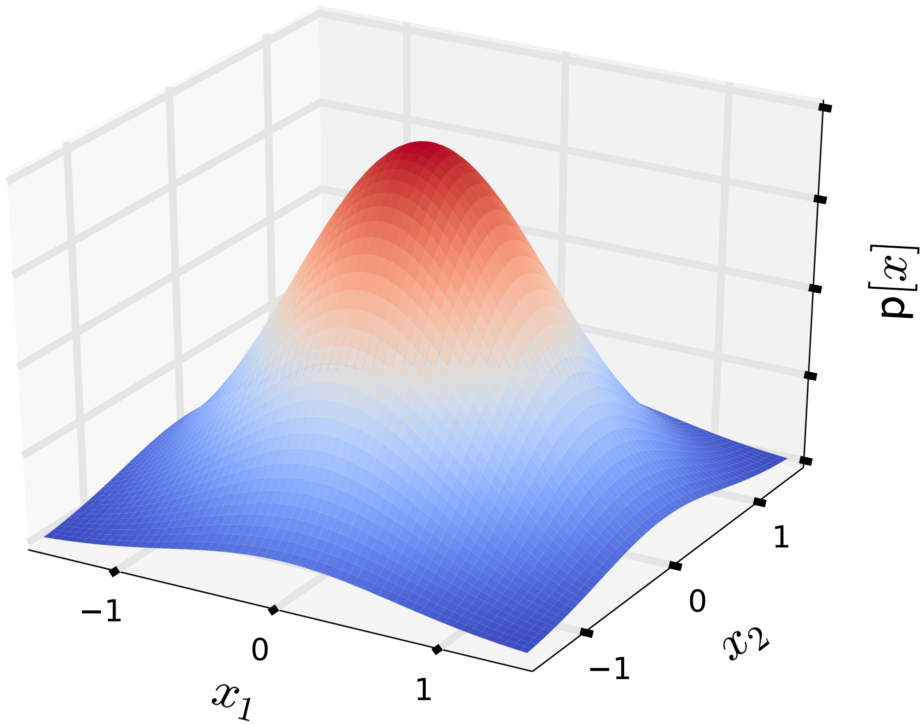
- Gibbs distributions: set $\mathcal{Q}$ of distributions $p_{\mathbf{w}}$ with $\mathbf{w} \in \mathbb{R}^N$,

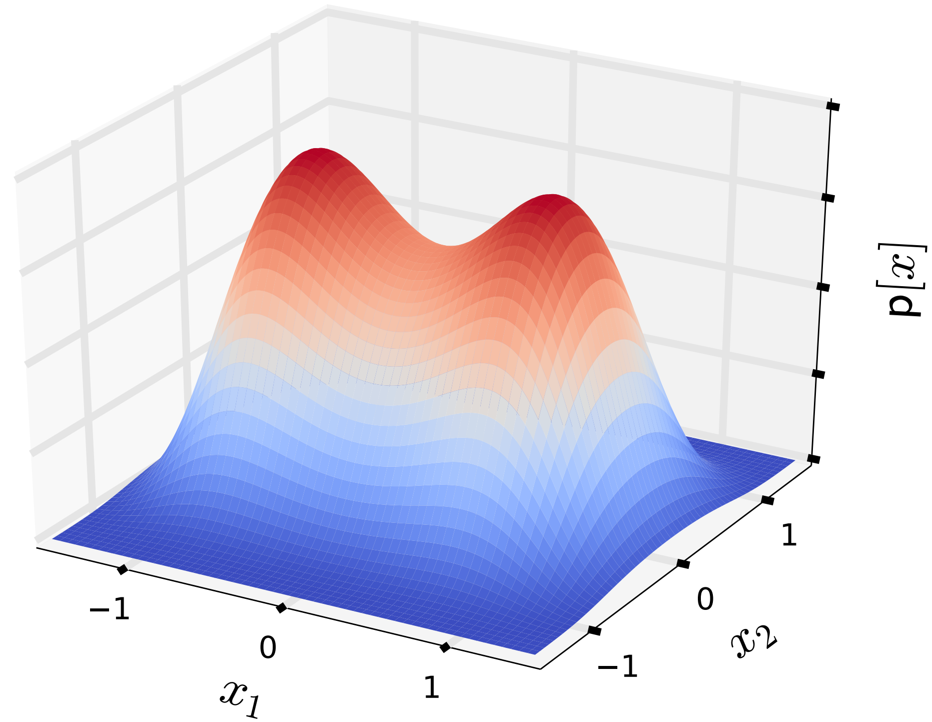
$$\mathsf{p}_{\mathbf{w}}[x] = \frac{\mathsf{p}_0[x] \exp\big(\mathbf{w} \cdot \mathbf{\Phi}(x)\big)}{Z} = \frac{\mathsf{p}_0[x] \exp\big(\sum_{j=1}^N w_j \Phi_j(x)\big)}{Z},$$

with $Z = \sum_x \mathsf{p}_0[x] \exp\big(\mathbf{w} \cdot \mathbf{\Phi}(x)\big)$.

- Rich family:
  - for linear and quadratic features: includes Gaussians and other distributions with non-PSD quadratic forms in exponents.
  - for higher-degree polynomials of raw features: more complex multi-modal distributions.

Examples





$$\mathsf{p}[(x_1, x_2)] = \frac{e^{-(x_1^2+x_2^2)}}{Z}.$$

$$\mathsf{p}[(x_1, x_2)] = \frac{e^{-(x_1^4+x_2^4)+x_1^2-x_2^2}}{Z}.$$

Dual Problems

- Regularized Maxent problem:

$$\min_{\mathsf{p}} F(\mathsf{p}) = \overline{D}(\mathsf{p} \parallel \mathsf{p}_0) + I_C(\mathop{\mathrm{E}}_{\mathsf{p}}[\mathbf{\Phi}]),$$

$$\text{with } \begin{cases} \overline{D}(\mathsf{p} \parallel \mathsf{p}_0) = D(\mathsf{p} \parallel \mathsf{p}_0) \text{ if } \mathsf{p} \in \Delta, +\infty \text{ otherwise;} \\ C = \left\{\mathbf{u} \colon \|\mathbf{u} - \mathop{\mathrm{E}}_{S}[\mathbf{\Phi}]\|_\infty \leq \beta\right\}; \\ I_C(x) = 0 \text{ if } x \in C,\ I_C(x) = +\infty \text{ otherwise.} \end{cases}$$

- Regularized Maximum Likelihood problem with Gibbs distributions:

$$\sup_{\mathbf{w}} G(\mathbf{w}) = \frac{1}{m}\sum_{i=1}^{m} \log\left[\frac{\mathsf{p}_{\mathbf{w}}[x_i]}{\mathsf{p}_0[x_i]}\right] - \beta\|\mathbf{w}\|_1.$$

Duality Theorem

(Della Pietra et al., 1997; Dudík et al., 2007; Cortes et al., 2015)

- Theorem: the regularized Maxent and ML with Gibbs distributions problems are equivalent,

$$\sup_{\mathbf{w}\in\mathbb{R}^N} G(\mathbf{w}) = \min_{\mathsf{p}} F(\mathsf{p}).$$

  - furthemore, let $\mathsf{p}^* = \operatorname*{argmin}_{\mathsf{p}} F(\mathsf{p})$, then, for any $\epsilon > 0$,

$$\Big(\big|G(\mathbf{w}) - \sup_{\mathbf{w}\in\mathbb{R}^N} G(\mathbf{w})\big| < \epsilon\Big) \Rightarrow \Big(D(\mathsf{p}^* \,\|\, \mathsf{p}_{\mathbf{w}}) \leq \epsilon\Big).$$

Notes

- Maxent formulation:
  - no explicit restriction to a family of distributions $\mathcal{P}$.
  - but solution coincides with regularized ML with a specific family $\mathcal{P}$!
  - more general Bregman divergence-based formulation.

L_1-Regularized Maxent

(Kazama and Tsujii, 2003)

- Optimization problem:

$$\inf_{\mathbf{w}\in\mathbb{R}^N} \beta\|\mathbf{w}\|_1 - \frac{1}{m}\sum_{i=1}^{m} \log \mathsf{p}_{\mathbf{w}}[x_i].$$

$$\text{where } \mathsf{p}_{\mathbf{w}}[x] = \frac{1}{Z}\exp\big(\mathbf{w}\cdot\mathbf{\Phi}(x)\big).$$

- Bayesian interpretation: equivalent to MAP with Laplacian prior $q_{\text{prior}}(\mathbf{w})$ (Williams, 1994),

$$\max_{\mathbf{w}} \log\Big(\prod_{i=1}^{m} \mathsf{p}_{\mathbf{w}}[x_i]\, \mathsf{q}_{\text{prior}}(\mathbf{w})\Big)$$

$$\text{with } q_{\text{prior}}(\mathbf{w}) = \prod_{j=1}^{N} \frac{\beta_j}{2}\exp(-\beta_j|w_j|).$$

Generalization Guarantee

(Dudík et al., 2007)

- Notation: $\mathcal{L}_{\mathcal{D}}(\mathbf{w}) = \underset{x \sim \mathcal{D}}{\mathrm{E}}[-\log \mathsf{p}_{\mathbf{w}}[x]]$, $\mathcal{L}_S(\mathbf{w}) = \underset{x \sim S}{\mathrm{E}}[-\log \mathsf{p}_{\mathbf{w}}[x]]$.
- Theorem: Fix $\delta > 0$. Let $\widehat{\mathbf{w}}$ be the solution of the L1-reg. Maxent problem for $\beta = 2\mathfrak{R}_m(H) + \Lambda\sqrt{\log(\frac{2}{\delta})/2m}$. Then, with probability at least $1 - \delta$,

$$\mathcal{L}_{\mathcal{D}}(\widehat{\mathbf{w}}) \leq \inf_{\mathbf{w}} \mathcal{L}_{\mathcal{D}}(\mathbf{w}) + 2\|\mathbf{w}\|_1 \left[2\mathfrak{R}_m(H) + \Lambda\sqrt{\frac{\log \frac{2}{\delta}}{2m}}\right].$$

Proof

- By Hölder's inequality and the concentration bound for average feature vectors,

$$\begin{aligned}
\mathcal{L}_{\mathcal{D}}(\widehat{\mathbf{w}}) - \mathcal{L}_S(\widehat{\mathbf{w}}) &= \widehat{\mathbf{w}} \cdot \big[\mathop{\mathrm{E}}_{S}[\mathbf{\Phi}] - \mathop{\mathrm{E}}_{\mathcal{D}}[\mathbf{\Phi}]\big] \\
&\leq \|\widehat{\mathbf{w}}\|_1 \, \big\| \mathop{\mathrm{E}}_{S}[\mathbf{\Phi}] - \mathop{\mathrm{E}}_{\mathcal{D}}[\mathbf{\Phi}] \big\|_\infty \leq \beta \|\widehat{\mathbf{w}}\|_1.
\end{aligned}$$

- Since $\widehat{\mathbf{w}}$ is a minimizer,

$$\begin{aligned}
\mathcal{L}_{\mathcal{D}}(\widehat{\mathbf{w}}) - \mathcal{L}_{\mathcal{D}}(\mathbf{w}) &= \mathcal{L}_{\mathcal{D}}(\widehat{\mathbf{w}}) - \mathcal{L}_S(\widehat{\mathbf{w}}) + \mathcal{L}_S(\widehat{\mathbf{w}}) - \mathcal{L}_{\mathcal{D}}(\mathbf{w}) \\
&\leq \beta \|\widehat{\mathbf{w}}\|_1 + \mathcal{L}_S(\widehat{\mathbf{w}}) - \mathcal{L}_{\mathcal{D}}(\mathbf{w}) \\
&\leq \beta \|\mathbf{w}\|_1 + \mathcal{L}_S(\mathbf{w}) - \mathcal{L}_{\mathcal{D}}(\mathbf{w}) \leq 2\beta \|\mathbf{w}\|_1. \\
&\qquad\qquad (\widehat{\mathbf{w}} \text{ minimizer of } \beta\|\mathbf{w}\|_1 + \mathcal{L}_S(\mathbf{w}))
\end{aligned}$$

L_2-Regularized Maxent

(Chen and Rosenfeld, 2000; Lebanon and Lafferty, 2001)

- Different relaxations:
  - L_1 constraints:

$$\forall j \in [1, N], \quad \left| \mathop{\mathrm{E}}_{x \sim p}[\Phi_j(x)] - \mathop{\mathrm{E}}_{x \sim \widehat{p}}[\Phi_j(x)] \right| \leq \beta_j.$$

  - L_2 constraints:

$$\left\| \mathop{\mathrm{E}}_{x \sim p}[\mathbf{\Phi}(x)] - \mathop{\mathrm{E}}_{x \sim \widehat{p}}[\mathbf{\Phi}(x)] \right\|_2 \leq B.$$

L_2-Regularized Maxent

- Optimization problem:

$$\inf_{\mathbf{w}\in\mathbb{R}^N} \beta\|\mathbf{w}\|_2^2 - \frac{1}{m}\sum_{i=1}^{m}\log \mathsf{p}_{\mathbf{w}}[x_i].$$

$$\text{where } \mathsf{p}_{\mathbf{w}}[x] = \frac{1}{Z}\exp\big(\mathbf{w}\cdot\mathbf{\Phi}(x)\big).$$

- Bayesian interpretation: equivalent to MAP with Gaussian prior $q_{\text{prior}}(\mathbf{w})$ (Goodman, 2004),

$$\max_{\mathbf{w}} \ \log\Big(\prod_{i=1}^{m}\mathsf{p}_{\mathbf{w}}[x_i]\,\mathsf{q}_{\text{prior}}(\mathbf{w})\Big)$$

$$\text{with } \mathsf{q}_{\text{prior}}(\mathbf{w}) = \prod_{j=1}^{N}\frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{w_j^2}{2\sigma^2}}.$$

This Lecture

- Notions of information theory.
- Introduction to density estimation.
- Maxent models.
- Conditional Maxent models.

Conditional Maxent Models

- Maxent models for conditional probabilities:
  - conditional probability modeling each class.
  - use in multi-class classification.
  - can use different features for each class.
  - a.k.a. multinomial logistic regression.
  - logistic regression: special case of two classes.

Problem

- Data: sample drawn i.i.d. according to some distribution D,

$$S = ((x_1, y_1), \ldots, (x_m, y_m)) \in (\mathcal{X} \times \mathcal{Y})^m.$$

  - $\mathcal{Y} = \{1, \ldots, k\}$, or $\mathcal{Y} = \{0, 1\}^k$ in multi-label case.

- Features: mapping $\mathbf{\Phi}\colon \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^N$.

- Problem: find accurate conditional probability models $\Pr[\cdot \mid x]$, $x \in \mathcal{X}$, based on $\mathbf{\Phi}$.

Conditional Maxent Principle

(Berger et al., 1996; Cortes et al., 2015)

- Idea: empirical feature vector average close to expectation. For any $\delta > 0$, with probability at least $1 - \delta$,
$$\left\| \mathop{\mathrm{E}}_{\substack{x \sim \widehat{\mathsf{p}} \\ y \sim \mathcal{D}[\cdot|x]}} [\mathbf{\Phi}(x, y)] - \mathop{\mathrm{E}}_{\substack{x \sim \widehat{\mathsf{p}} \\ y \sim \widehat{\mathsf{p}}[\cdot|x]}} [\mathbf{\Phi}(x, y)] \right\|_{\infty} \leq 2\mathfrak{R}_m(H) + \sqrt{\frac{\log \frac{2}{\delta}}{2m}}.$$
- Maxent principle: find conditional distributions $\mathsf{p}[\cdot|x]$ that are closest to priors $\mathsf{p}_0[\cdot|x]$ (typically uniform distributions) while verifying $\left\| \mathrm{E}_{\substack{x \sim \widehat{\mathsf{p}} \\ y \sim \mathsf{p}[\cdot|x]}} [\mathbf{\Phi}(x, y)] - \mathrm{E}_{\substack{x \sim \widehat{\mathsf{p}} \\ y \sim \widehat{\mathsf{p}}[\cdot|x]}} [\mathbf{\Phi}(x, y)] \right\|_{\infty} \leq \beta$.
- Closeness is measured using conditional relative entropy based on $\widehat{\mathsf{p}}$.

Cond. Maxent Formulation

(Berger et al., 1996; Cortes et al., 2015)

- Optimization problem: find distribution p solution of

$$\min_{\mathsf{p}[\cdot|x] \in \Delta} \sum_{x \in \mathcal{X}} \widehat{\mathsf{p}}[x] \, D\big(\mathsf{p}[\cdot|x] \parallel \mathsf{p}_0[\cdot|x]\big)$$

$$\text{s.t.} \left\| \mathop{\mathrm{E}}_{x \sim \widehat{\mathsf{p}}} \left[\mathop{\mathrm{E}}_{y \sim \mathsf{p}[\cdot|x]} [\mathbf{\Phi}(x, y)] \right] - \mathop{\mathrm{E}}_{(x,y) \sim S} [\mathbf{\Phi}(x, y)] \right\|_\infty \leq \beta.$$

  - convex optimization problem, unique solution.
  - $\beta = 0$: unregularized conditional Maxent.
  - $\beta > 0$: regularized conditional Maxent.

Dual Problems

- Regularized conditional Maxent problem:

$$\widetilde{F}(\mathsf{p}) = \mathop{\mathrm{E}}_{x \sim \widehat{p}} \Big[\overline{D}\big(\mathsf{p}[\cdot|x] \parallel \mathsf{p}_0[\cdot|x]\big) + I_\Delta\big(\mathsf{p}[\cdot|x]\big) \Big] + I_C\Big(\mathop{\mathrm{E}}_{\substack{x \sim \widehat{\mathsf{p}} \\ y \sim \mathsf{p}[\cdot|x]}} [\mathbf{\Phi}] \Big).$$

- Regularized Maximum Likelihood problem with conditional Gibbs distributions:

$$\widetilde{G}(\mathbf{w}) = \frac{1}{m} \sum_{i=1}^{m} \log \left[\frac{\mathsf{p}_{\mathbf{w}}[y_i|x_i]}{\mathsf{p}_0[y_i|x_i]} \right] - \beta \|\mathbf{w}\|_1 ,$$

where $\forall (x, y) \in \mathcal{X} \times \mathcal{Y}$,

$$\mathsf{p}_{\mathbf{w}}[y|x] = \frac{\mathsf{p}_0[y|x] \exp\big(\mathbf{w} \cdot \mathbf{\Phi}(x, y)\big)}{Z(x)}$$

$$Z(x) = \sum_{y \in \mathcal{Y}} \mathsf{p}_0[y|x] \exp(\mathbf{w} \cdot \mathbf{\Phi}(x, y)).$$

Duality Theorem

(Cortes et al., 2015)

- Theorem: the regularized conditional Maxent and ML with conditional Gibbs distributions problems are equivalent,

$$\sup_{\mathbf{w} \in \mathbb{R}^N} \widetilde{G}(\mathbf{w}) = \min_{\mathsf{p}} \widetilde{F}(\mathsf{p}).$$

  - furthemore, let $\mathsf{p}^* = \operatorname*{argmin}_{\mathsf{p}} \widetilde{F}(\mathsf{p})$, then, for any $\epsilon > 0$,

$$\Big(|\widetilde{G}(\mathbf{w}) - \sup_{\mathbf{w} \in \mathbb{R}^N} \widetilde{G}(\mathbf{w})| < \epsilon\Big) \Rightarrow \operatorname*{E}_{x \sim \widehat{p}}\Big[D\big(\mathsf{p}^*[\cdot|x] \parallel \mathsf{p}_{\mathbf{w}}[\cdot|x]\big)\Big] \leq \epsilon.$$

Regularized Cond. Maxent

(Berger et al., 1996; Cortes et al., 2015)

- Optimization problem: convex optimizations, regularization parameter $\lambda \geq 0$.

$$\min_{\mathbf{w} \in \mathbb{R}^N} \quad \lambda \|\mathbf{w}\|_1 - \frac{1}{m} \sum_{i=1}^{m} \log \mathsf{p}_{\mathbf{w}}[y_i | x_i]$$

$$\text{or} \quad \min_{\mathbf{w} \in \mathbb{R}^N} \quad \lambda \|\mathbf{w}\|_2^2 - \frac{1}{m} \sum_{i=1}^{m} \log \mathsf{p}_{\mathbf{w}}[y_i | x_i],$$

where $\forall (x, y) \in \mathcal{X} \times \mathcal{Y}$,

$$\mathsf{p}_{\mathbf{w}}[y|x] = \frac{\exp(\mathbf{w} \cdot \mathbf{\Phi}(x, y))}{Z(x)}$$

$$Z(x) = \sum_{y \in \mathcal{Y}} \exp(\mathbf{w} \cdot \mathbf{\Phi}(x, y)).$$

More Explicit Forms

- Optimization problem: multinomial logistic loss.

$$\min_{\mathbf{w}\in\mathbb{R}^N} \begin{cases} \lambda\|\mathbf{w}\|_1 \\ \lambda\|\mathbf{w}\|_2^2 \end{cases} + \frac{1}{m}\sum_{i=1}^{m} \log\left[\sum_{y\in\mathcal{Y}} \exp\Big(\mathbf{w}\cdot\mathbf{\Phi}(x_i, y) - \mathbf{w}\cdot\mathbf{\Phi}(x_i, y_i)\Big)\right].$$

$$\min_{\mathbf{w}\in\mathbb{R}^N} \begin{cases} \lambda\|\mathbf{w}\|_1 \\ \lambda\|\mathbf{w}\|_2^2 \end{cases} - \mathbf{w}\cdot\frac{1}{m}\sum_{i=1}^{m}\mathbf{\Phi}(x_i, y_i) + \frac{1}{m}\sum_{i=1}^{m} \log\left[\sum_{y\in\mathcal{Y}} e^{\mathbf{w}\cdot\mathbf{\Phi}(x_i, y)}\right].$$

Related Problem

- Optimization problem: log-sum-exp replaced by max.

$$\min_{\mathbf{w}\in\mathbb{R}^N} \begin{cases} \lambda\|\mathbf{w}\|_1 \\ \lambda\|\mathbf{w}\|_2^2 \end{cases} + \frac{1}{m}\sum_{i=1}^{m} \underbrace{\max_{y\in\mathcal{Y}}\Big(\mathbf{w}\cdot\mathbf{\Phi}(x_i,y) - \mathbf{w}\cdot\mathbf{\Phi}(x_i,y_i)\Big)}_{-\rho_{\mathbf{w}}(x_i,y_i)}.$$

Common Feature Choice

- Multi-class features:

$$\mathbf{\Phi}(x,y) = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \mathbf{\Gamma}(x) \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad \mathbf{w} = \begin{bmatrix} \mathbf{w}_1 \\ \vdots \\ \mathbf{w}_{y-1} \\ \mathbf{w}_y \\ \mathbf{w}_{y+1} \\ \vdots \\ \mathbf{w}_{|\mathcal{Y}|} \end{bmatrix} \longrightarrow \mathbf{w} \cdot \mathbf{\Phi}(x,y) = \mathbf{w}_y \cdot \mathbf{\Gamma}(x).$$

- L_2-regularized cond. maxent optimization:

$$\min_{\mathbf{w} \in \mathbb{R}^N} \lambda \sum_{y \in \mathcal{Y}} \|\mathbf{w}_y\|_2^2 + \frac{1}{m} \sum_{i=1}^{m} \log \left[\sum_{y \in \mathcal{Y}} \exp\Big(\mathbf{w}_y \cdot \mathbf{\Gamma}(x_i) - \mathbf{w}_{y_i} \cdot \mathbf{\Gamma}(x_i)\Big) \right].$$

Prediction

- Prediction with $\mathsf{p}_{\mathbf{w}}[y|x] = \frac{\exp(\mathbf{w}\cdot\mathbf{\Phi}(x,y))}{Z(x)}$:

$$\widehat{y}(x) = \operatorname{argmax}_{y \in \mathcal{Y}} \mathsf{p}_{\mathbf{w}}[y|x] = \operatorname{argmax}_{y \in \mathcal{Y}} \mathbf{w} \cdot \mathbf{\Phi}(x, y).$$

Binary Classification

- Simpler expression:

$$\begin{aligned}
&\sum_{y \in \mathcal{Y}} \exp\Big(\mathbf{w} \cdot \mathbf{\Phi}(x_i, y) - \mathbf{w} \cdot \mathbf{\Phi}(x_i, y_i)\Big) \\
&= e^{\mathbf{w} \cdot \mathbf{\Phi}(x_i, +1) - \mathbf{w} \cdot \mathbf{\Phi}(x_i, y_i)} + e^{\mathbf{w} \cdot \mathbf{\Phi}(x_i, -1) - \mathbf{w} \cdot \mathbf{\Phi}(x_i, y_i)} \\
&= 1 + e^{-y_i \mathbf{w} \cdot [\mathbf{\Phi}(x_i, +1) - \mathbf{\Phi}(x_i, -1)]} \\
&= 1 + e^{-y_i \mathbf{w} \cdot \mathbf{\Psi}(x_i)},
\end{aligned}$$

with $\mathbf{\Psi}(x) = \mathbf{\Phi}(x, +1) - \mathbf{\Phi}(x, -1)$.

Logistic Regression

(Berkson, 1944)

- Binary case of conditional Maxent.
- Optimization problem: regularized logistic loss.

$$\min_{\mathbf{w}\in\mathbb{R}^N} \begin{Bmatrix} \lambda\|\mathbf{w}\|_1 \\ \lambda\|\mathbf{w}\|_2^2 \end{Bmatrix} + \frac{1}{m}\sum_{i=1}^{m} \log\left[1 + e^{-y_i \mathbf{w}\cdot\boldsymbol{\Psi}(x_i)}\right].$$

  - convex optimization.
  - variety of solutions: SGD, coordinate descent, etc.
  - coordinate descent: similar to AdaBoost with logistic loss $\phi(-u) = \log_2(1 + e^{-u}) \geq 1_{u \leq 0}$ instead of exponential loss.

Generalization Bound

- Theorem: assume that $\pm\Phi_j \in H$ for all $j \in [1, N]$. Then, for any $\delta > 0$, with probability at least $1 - \delta$ over the draw of a sample S of size m, for all $f : x \mapsto \mathbf{w} \cdot \Phi(x)$,

$$R(f) \leq \frac{1}{m} \sum_{i=1}^{m} \log_{u_0}\left(1 + e^{-y_i \mathbf{w} \cdot \Phi(x_i)}\right) + 4\|\mathbf{w}\|_1 \mathfrak{R}_m(H)$$

$$+ \sqrt{\frac{\log \log_2 2\|\mathbf{w}\|_1}{m}} + \sqrt{\frac{\log \frac{2}{\delta}}{m}},$$

where $u_0 = 1 + \frac{1}{e}$.

Proof

- Proof: by the learning bound for convex ensembles holding uniformly for all ρ, with probability at least $1-\delta$, for all f and $\rho > 0$,
$$R(f) \leq \frac{1}{m}\sum_{i=1}^{m} 1_{\frac{y_i \mathbf{w}\cdot\Phi(x_i)}{\rho\|\mathbf{w}\|_1} - 1 \leq 0} + \frac{4}{\rho}\mathfrak{R}_m(H) + \sqrt{\frac{\log\log_2 \frac{2}{\rho}}{m}} + \sqrt{\frac{\log \frac{2}{\delta}}{m}}.$$

- Choosing $\rho = \frac{1}{\|\mathbf{w}\|_1}$ and using $1_{u \leq 1} \leq \log_{u_0}(1+e^{-u})$ yields immediately the learning bound of the theorem.

Logistic Regression

(Berkson, 1944)

- Logistic model:

$$\Pr[y=+1 \mid x] = \frac{e^{\mathbf{w}\cdot\mathbf{\Phi}(x,+1)}}{Z(x)},$$

where $Z(x) = e^{\mathbf{w}\cdot\mathbf{\Phi}(x,+1)} + e^{\mathbf{w}\cdot\mathbf{\Phi}(x,-1)}$

- Properties:
  - linear decision rule, sign of log-odds ratio:

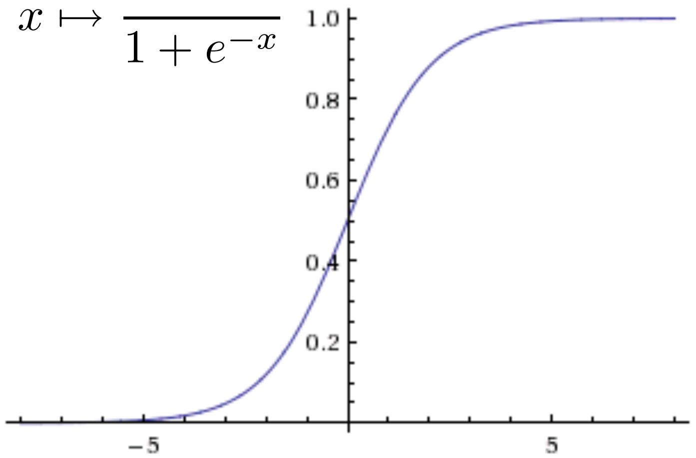
$$\log \frac{\Pr[y=+1 \mid x]}{\Pr[y=-1 \mid x]} = \mathbf{w} \cdot \big(\mathbf{\Phi}(x,+1) - \mathbf{\Phi}(x,-1)\big) = \mathbf{w} \cdot \mathbf{\Psi}(x).$$

  - logistic form:

$$\Pr[y=+1 \mid x] = \frac{1}{1 + e^{-\mathbf{w}\cdot[\mathbf{\Phi}(x,+1)-\mathbf{\Phi}(x,-1)]}} = \frac{1}{1 + e^{-\mathbf{w}\cdot\mathbf{\Psi}(x)}}.$$

Logistic/Sigmoid Function

$$f\colon x \mapsto \frac{1}{1+e^{-x}}$$



$$\Pr[y=+1 \mid x] = f\big(\mathbf{w} \cdot \mathbf{\Psi}(x)\big).$$

Applications

- Natural language processing (Berger et al., 1996; Rosenfeld, 1996; Pietra et al., 1997; Malouf, 2002; Manning and Klein, 2003; Mann et al., 2009; Ratnaparkhi, 2010).
- Species habitat modeling (Phillips et al., 2004, 2006; Dudík et al., 2007; Elith et al, 2011).
- Computer vision (Jeon and Manmatha, 2004).

Extensions

- Extensive theoretical study of alternative regularizations: (Dudík et al., 2007) (see also (Altun and Smola, 2006) though some proofs unclear).
- Maxent models with other Bregman divergences (see for example (Altun and Smola, 2006)).
- Structural Maxent models (Cortes et al., 2015):
  - extension to the case of multiple feature families.
  - empirically outperform Maxent and L1-Maxent.
  - conditional structural Maxent: coincide with deep boosting using the logistic loss.

Conclusion

- Logistic regression/maxent models:
  - theoretical foundation.
  - natural solution when probabilites are required.
  - widely used for density estimation/classification.
  - often very effective in practice.
  - distributed optimization solutions.
  - no natural non-linear L1-version (use of kernels).
  - connections with boosting.
  - connections with neural networks.

References

- Yasemin Altun, Alexander J. Smola. Unifying Divergence Minimization and Statistical Inference Via Convex Duality. COLT 2006: 139-153
- Adam Berger, Stephen Della Pietra, and Vincent Della Pietra. A maximum entropy approach to natural language processing. Computational Linguistics, (22-1), March 1996;
- Berkson, J. (1944). Application of the logistic function to bio-assay. Journal of the American Statistical Association 39, 357–365.
- Lev M. Bregman. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. USSR Computational Mathematics and Mathematical Physics, 7:200–217, 1967.
- Corinna Cortes, Vitaly Kuznetsov, Mehryar Mohri, and Umar Syed. Structural Maxent. In Proceedings of ICML, 2015.

References

- Imre Csiszar and Tusnady. Information geometry and alternating minimization procedures. Statistics and Decisions, Supplement Issue 1, 205-237, 1984.
- Imre Csiszar. A geometric interpretation of Darroch and Ratchliff's generalized iterative scaling. The Annals of Statisics, 17(3), pp. 1409-1413. 1989.
- J. Darroch and D. Ratchliff. Generalized iterative scaling for log-linear models. The Annals of Mathematical Statistics, 43(5), pp. 1470-1480, 1972.
- Stephen Della Pietra, Vincent Della Pietra, and John Lafferty. Inducing features of random fields. IEEE Transactions on Pattern Analysis and Machine Intelligence 19:4, pp.380--393, April, 1997.
- Dudík, Miroslav, Phillips, Steven J., and Schapire, Robert E. Maximum entropy density estimation with generalized regularization and an application to species distribution modeling. JMLR, 8, 2007.

References

- E. Jaynes. Information theory and statistical mechanics. Physics Reviews, 106:620–630, 1957.
- E. Jaynes. Papers on Probability, Statistics, and Statistical Physics. R. Rosenkrantz (editor), D. Reidel Publishing Company, 1983.
- Solomon Kullback and Richard A. Leibler. On information and sufficiency. Ann. Math. Statist., 22(1):79–86, 1951.
- O'Sullivan. Alternating minimzation algorithms: From Blahut-Arimoto to expectation-maximization. Codes, Curves and Signals: Common Threads in Communications, A. Vardy, (editor), Kluwer, 1998.

References

- Alfréd Rényi. On measures of entropy and information. In Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics, pages 547–561. University of California Press, 1961.
- Roni Rosenfeld. A maximum entropy approach to adaptive statistical language modelling. Computer, Speech and Language 10:187--228, 1996.
- Claude E. Shannon. A mathematical theory of communication. Bell System Technical Journal, 27:379423, 1948.

Foundations of Machine Learning
On-Line Learning

Mehryar Mohri
Courant Institute and Google Research
mohri@cims.nyu.edu

Motivation

- PAC learning:
  - distribution fixed over time (training and test).
  - IID assumption.
- On-line learning:
  - no distributional assumption.
  - worst-case analysis (adversarial).
  - mixed training and test.
  - Performance measure: mistake model, regret.

This Lecture

- Prediction with expert advice
- Linear classification

General On-Line Setting

- For $t=1$ to T do
  - receive instance $x_t \in X$.
  - predict $\widehat{y}_t \in Y$.
  - receive label $y_t \in Y$.
  - incur loss $L(\widehat{y}_t, y_t)$.
- Classification: $Y = \{0, 1\}, L(y, y') = |y' - y|$.
- Regression: $Y \subseteq \mathbb{R}, L(y, y') = (y' - y)^2$.
- Objective: minimize total loss $\sum_{t=1}^{T} L(\widehat{y}_t, y_t)$.

Prediction with Expert Advice

- For $t = 1$ to T do
  - receive instance $x_t \in X$ and advice $y_{t,i} \in Y, i \in [1, N]$.
  - predict $\widehat{y}_t \in Y$.
  - receive label $y_t \in Y$.
  - incur loss $L(\widehat{y}_t, y_t)$.
- Objective: minimize regret, i.e., difference of total loss incurred and that of best expert.

$$\text{Regret}(T) = \sum_{t=1}^{T} L(\widehat{y}_t, y_t) - \min_{i=1}^{N} \sum_{t=1}^{T} L(y_{t,i}, y_t).$$

Mistake Bound Model

- Definition: the maximum number of mistakes a learning algorithm L makes to learn c is defined by

$$M_L(c) = \max_{x_1, \ldots, x_T} |\text{mistakes}(L, c)|.$$

- Definition: for any concept class C the maximum number of mistakes a learning algorithm L makes is

$$M_L(C) = \max_{c \in C} M_L(c).$$

A mistake bound is a bound M on $M_L(C)$.

Halving Algorithm

see (Mitchell, 1997)

$\textsc{Halving}(H)$

$H_1 \leftarrow H$

for $t \leftarrow 1$ **to** T **do**

  $\textsc{Receive}(x_t)$

  $\widehat{y}_t \leftarrow \textsc{MajorityVote}(H_t, x_t)$

  $\textsc{Receive}(y_t)$

  if $\widehat{y}_t \neq y_t$ **then**

    $H_{t+1} \leftarrow \{c \in H_t \colon c(x_t) = y_t\}$

return H_{T+1}

Halving Algorithm - Bound

(Littlestone, 1988)

- Theorem: Let H be a finite hypothesis set, then

$$M_{Halving(H)} \leq \log_2 |H|.$$

- Proof: At each mistake, the hypothesis set is reduced at least by half.

VC Dimension Lower Bound

(Littlestone, 1988)

- Theorem: Let $\text{opt}(H)$ be the optimal mistake bound for H. Then,

$$\text{VCdim}(H) \leq \text{opt}(H) \leq M_{Halving(H)} \leq \log_2 |H|.$$

- Proof: for a fully shattered set, form a complete binary tree of the mistakes with height $\text{VCdim}(H)$.

Weighted Majority Algorithm

(Littlestone and Warmuth, 1988)

$\textsc{Weighted-Majority}(N \text{ experts})$ $\quad\triangleright\quad y_t, y_{t,i} \in \{0, 1\}.$

$\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\;\; \beta \in [0, 1).$

for $i \leftarrow 1$ **to** N **do**
    $w_{1,i} \leftarrow 1$
for $t \leftarrow 1$ **to** T **do**
    $\textsc{Receive}(x_t)$
    $\widehat{y}_t \leftarrow 1_{\sum_{y_{t,i}=1}^{N} w_t \geq \sum_{y_{t,i}=0}^{N} w_t}$ $\quad\triangleright$ weighted majority vot
    $\textsc{Receive}(y_t)$
    if $\widehat{y}_t \neq y_t$ **then**
        for $i \leftarrow 1$ **to** N **do**
            if $(y_{t,i} \neq y_t)$ **then**
                $w_{t+1,i} \leftarrow \beta w_{t,i}$
            else $w_{t+1,i} \leftarrow w_{t,i}$
return $\mathbf{w}_{T+1}$

Weighted Majority - Bound

- Theorem: Let m_t be the number of mistakes made by the WM algorithm till time t and m_t^* that of the best expert. Then, for all t,

$$\boxed{m_t \leq \frac{\log N + m_t^* \log \frac{1}{\beta}}{\log \frac{2}{1+\beta}}.}$$

  - Thus, $m_t \leq O(\log N) + \text{constant} \times \text{best expert}.$
  - Realizable case: $m_t \leq O(\log N).$
  - Halving algorithm: $\beta = 0$.

Weighted Majority - Proof

- Potential: $\Phi_t = \sum_{i=1}^{N} w_{t,i}$.
- Upper bound: after each error,

$$\Phi_{t+1} \leq \left[\tfrac{1}{2} + \tfrac{1}{2} \times \beta\right] \Phi_t = \left[\frac{1+\beta}{2}\right] \Phi_t.$$

Thus, $\Phi_t \leq \left[\frac{1+\beta}{2}\right]^{m_t} N.$

- Lower bound: for any expert i, $\Phi_t \geq w_{t,i} = \beta^{m_{t,i}}$.
- Comparison: $\beta^{m_t^*} \leq \left[\frac{1+\beta}{2}\right]^{m_t} N$

$$\Rightarrow m_t^* \log \beta \leq \log N + m_t \log \left[\frac{1+\beta}{2}\right]$$

$$\Rightarrow m_t \log \left[\frac{2}{1+\beta}\right] \leq \log N + m_t^* \log \frac{1}{\beta}.$$

Weighted Majority - Notes

- Advantage: remarkable bound requiring no assumption.
- Disadvantage: no deterministic algorithm can achieve a regret $R_T = o(T)$ with the binary loss.
  - better guarantee with randomized WM.
  - better guarantee for WM with convex losses.

Exponential Weighted Average

total loss incurred by expert i up to time t

- **Algorithm**:
  - weight update: $w_{t+1,i} \leftarrow w_{t,i}\, e^{-\eta L(y_{t,i}, y_t)} = e^{-\eta L_{t,i}}$.
  - prediction: $\widehat{y}_t = \frac{\sum_{i=1}^N w_{t,i} y_{t,i}}{\sum_{i=1}^N w_{t,i}}$.
- **Theorem**: assume that L is convex in its first argument and takes values in $[0,1]$. Then, for any $\eta > 0$ and any sequence $y_1, \ldots, y_T \in Y$, the regret at T satisfies

$$\text{Regret}(T) \leq \frac{\log N}{\eta} + \frac{\eta T}{8}.$$

For $\eta = \sqrt{8 \log N / T}$,

$$\boxed{\text{Regret}(T) \leq \sqrt{(T/2) \log N}}.$$

Exponential Weighted Avg - Proof

- **Potential:** $\Phi_t = \log \sum_{i=1}^N w_{t,i}.$
- **Upper bound:**

$$\begin{aligned}
\Phi_t - \Phi_{t-1} &= \log \frac{\sum_{i=1}^N w_{t-1,i}\, e^{-\eta L(y_{t,i}, y_t)}}{\sum_{i=1}^N w_{t-1,i}} \\
&= \log \Big(\operatorname*{E}_{w_{t-1}} [e^{-\eta L(y_{t,i}, y_t)}] \Big) \\
&= \log \left(\operatorname*{E}_{w_{t-1}} \left[\exp \left(-\eta \Big(L(y_{t,i}, y_t) - \operatorname*{E}_{w_{t-1}} [L(y_{t,i}, y_t)] \Big) - \eta \operatorname*{E}_{w_{t-1}} [L(y_{t,i}, y_t)] \right) \right] \right) \\
&\leq -\eta \operatorname*{E}_{w_{t-1}} [L(y_{t,i}, y_t)] + \frac{\eta^2}{8} \quad \text{(Hoeffding's ineq.)} \\
&\leq -\eta L(\operatorname*{E}_{w_{t-1}} [y_{t,i}], y_t) + \frac{\eta^2}{8} \quad \text{(convexity of first arg. of } L) \\
&= -\eta L(\widehat{y}_t, y_t) + \frac{\eta^2}{8}.
\end{aligned}$$

Exponential Weighted Avg - Proof

- Upper bound: summing up the inequalities yields

$$\Phi_T - \Phi_0 \leq -\eta \sum_{t=1}^{T} L(\widehat{y}_t, y_t) + \frac{\eta^2 T}{8}.$$

- Lower bound:

$$\begin{aligned}\Phi_T - \Phi_0 = \log \sum_{i=1}^{N} e^{-\eta L_{T,i}} - \log N &\geq \log \max_{i=1}^{N} e^{-\eta L_{T,i}} - \log N \\ &= -\eta \min_{i=1}^{N} L_{T,i} - \log N.\end{aligned}$$

- Comparison:

$$-\eta \min_{i=1}^{N} L_{T,i} - \log N \leq -\eta \sum_{t=1}^{T} L(\widehat{y}_t, y_t) + \frac{\eta^2 T}{8}$$

$$\Rightarrow \sum_{t=1}^{T} L(\widehat{y}_t, y_t) - \min_{i=1}^{N} L_{T,i} \leq \frac{\log N}{\eta} + \frac{\eta T}{8}.$$

Exponential Weighted Avg - Notes

- Advantage: bound on regret per bound is of the form $\frac{R_T}{T} = O\Big(\sqrt{\frac{\log(N)}{T}}\Big)$.
- Disadvantage: choice of η requires knowledge of horizon T.

Doubling Trick

- Idea: divide time into periods $[2^k, 2^{k+1}-1]$ of length 2^k with $k=0, \ldots, n$, $T \geq 2^n - 1$, and choose $\eta_k = \sqrt{\frac{8 \log N}{2^k}}$ in each period.

- Theorem: with the same assumptions as before, for any T, the following holds:

$$\boxed{\text{Regret}(T) \leq \frac{\sqrt{2}}{\sqrt{2}-1} \sqrt{(T/2) \log N} + \sqrt{\log N/2}.}$$

Doubling Trick - Proof

- By the previous theorem, for any $I_k = [2^k, 2^{k+1} - 1]$,

$$L_{I_k} - \min_{i=1}^{N} L_{I_k,i} \leq \sqrt{2^k/2 \, \log N}.$$

Thus, $L_T = \sum_{k=0}^{n} L_{I_k} \leq \sum_{k=0}^{n} \min_{i=1}^{N} L_{I_k,i} + \sum_{k=0}^{n} \sqrt{2^k \, (\log N)/2}$

$$\leq \min_{i=1}^{N} L_{T,i} + \sum_{k=0}^{n} 2^{\frac{k}{2}} \sqrt{(\log N)/2}.$$

with

$$\sum_{i=0}^{n} 2^{\frac{k}{2}} = \frac{\sqrt{2}^{n+1} - 1}{\sqrt{2} - 1} = \frac{2^{(n+1)/2} - 1}{\sqrt{2} - 1} \leq \frac{\sqrt{2}\sqrt{T+1} - 1}{\sqrt{2} - 1} \leq \frac{\sqrt{2}(\sqrt{T} + 1) - 1}{\sqrt{2} - 1} \leq \frac{\sqrt{2}\sqrt{T}}{\sqrt{2} - 1} + 1.$$

Notes

- Doubling trick used in a variety of other contexts and proofs.
- More general method, learning parameter function of time: $\eta_t = \sqrt{(8 \log N)/t}$. Constant factor improvement:

$$\boxed{\text{Regret}(T) \leq 2\sqrt{(T/2)\log N} + \sqrt{(1/8)\log N}.}$$

This Lecture

- Prediction with expert advice
- Linear classification

Perceptron Algorithm

(Rosenblatt, 1958)

Perceptron($\mathbf{w}_0$)

$\mathbf{w}_1 \leftarrow \mathbf{w}_0$ $\quad \triangleright$ typically $\mathbf{w}_0 = \mathbf{0}$

for $t \leftarrow 1$ **to** T **do**

  Receive($\mathbf{x}_t$)

  $\widehat{y}_t \leftarrow \operatorname{sgn}(\mathbf{w}_t \cdot \mathbf{x}_t)$

  Receive(y_t)

  if $(\widehat{y}_t \neq y_t)$ **then**

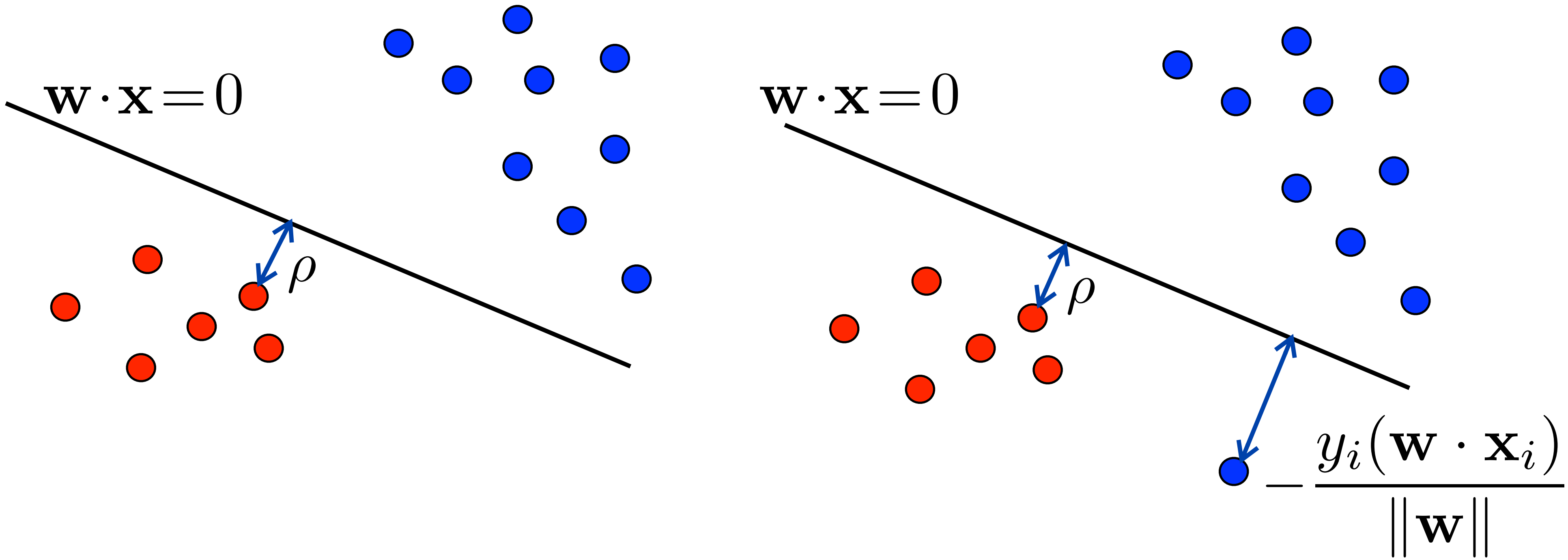
    $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t + y_t \mathbf{x}_t$ $\quad \triangleright$ more generally $\eta y_t \mathbf{x}_t, \eta > 0$

  else $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t$

return $\mathbf{w}_{T+1}$

Separating Hyperplane

- Margin and errors



Perceptron = Stochastic Gradient Descent

- **Objective function**: convex but not differentiable.

$$F(\mathbf{w}) = \frac{1}{T}\sum_{t=1}^{T} \max\Big(0, -y_t(\mathbf{w}\cdot\mathbf{x}_t)\Big) = \operatorname*{E}_{\mathbf{x}\sim\widehat{D}}[f(\mathbf{w},\mathbf{x})]$$

with $f(\mathbf{w},\mathbf{x}) = \max\big(0, -y(\mathbf{w}\cdot\mathbf{x})\big)$.

- **Stochastic gradient**: for each $\mathbf{x}_t$, the update is

$$\mathbf{w}_{t+1} \leftarrow \begin{cases} \mathbf{w}_t - \eta\nabla_{\mathbf{w}} f(\mathbf{w}_t, \mathbf{x}_t) & \text{if differentiable} \\ \mathbf{w}_t & \text{otherwise,} \end{cases}$$

where $\eta > 0$ is a learning rate parameter.

- **Here**:

$$\mathbf{w}_{t+1} \leftarrow \begin{cases} \mathbf{w}_t + \eta y_t \mathbf{x}_t & \text{if } y_t(\mathbf{w}_t\cdot\mathbf{x}_t) < 0 \\ \mathbf{w}_t & \text{otherwise.} \end{cases}$$

Perceptron Algorithm - Bound

(Novikoff, 1962)

- Theorem: Assume that $\|x_t\| \leq R$ for all $t \in [1, T]$ and that for some $\rho > 0$ and $\mathbf{v} \in \mathbb{R}^N$, for all $t \in [1, T]$,

$$\rho \leq \frac{y_t(\mathbf{v} \cdot \mathbf{x}_t)}{\|\mathbf{v}\|}.$$

  Then, the number of mistakes made by the perceptron algorithm is bounded by $\boxed{R^2/\rho^2}$.

- Proof: Let I be the set of ts at which there is an update and let M be the total number of updates.

- Summing up the assumption inequalities gives:

$$
\begin{aligned}
M\rho &\leq \frac{\mathbf{v} \cdot \sum_{t \in I} y_t \mathbf{x}_t}{\|\mathbf{v}\|} \\
&= \frac{\mathbf{v} \cdot \sum_{t \in I} (\mathbf{w}_{t+1} - \mathbf{w}_t)}{\|\mathbf{v}\|} && \text{(definition of updates)} \\
&= \frac{\mathbf{v} \cdot \mathbf{w}_{T+1}}{\|\mathbf{v}\|} \\
&\leq \|\mathbf{w}_{T+1}\| && \text{(Cauchy-Schwarz ineq.)} \\
&= \|\mathbf{w}_{t_m} + y_{t_m} \mathbf{x}_{t_m}\| && (t_m \text{ largest } t \text{ in } I) \\
&= \Big[\|\mathbf{w}_{t_m}\|^2 + \|\mathbf{x}_{t_m}\|^2 + 2 \underbrace{y_{t_m} \mathbf{w}_{t_m} \cdot \mathbf{x}_{t_m}}_{\leq 0}\Big]^{1/2} \\
&\leq \Big[\|\mathbf{w}_{t_m}\|^2 + R^2\Big]^{1/2} \\
&\leq \Big[M R^2\Big]^{1/2} = \sqrt{M} R. && \text{(applying the same to previous } t\text{s in } I)
\end{aligned}
$$

- Notes:
  - bound independent of dimension and tight.
  - convergence can be slow for small margin, it can be in $\Omega(2^N)$.
  - among the many variants: voted perceptron algorithm. Predict according to

$$\operatorname{sign}\Big(\big(\sum_{t \in I} c_t \mathbf{w}_t\big) \cdot \mathbf{x}\Big),$$

    where c_t is the number of iterations $\mathbf{w}_t$ survives.
  - $\{x_t : t \in I\}$ are the support vectors for the perceptron algorithm.
  - non-separable case: does not converge.

Perceptron - Leave-One-Out Analysis

- Theorem: Let h_S be the hypothesis returned by the perceptron algorithm for sample $S=(x_1,\ldots,x_T)\sim D$ and let $M(S)$ be the number of updates defining h_S. Then,

$$\mathop{\mathrm{E}}_{S\sim D^m}[R(h_S)] \leq \mathop{\mathrm{E}}_{S\sim D^{m+1}}\left[\frac{\min(M(S), R^2_{m+1}/\rho^2_{m+1})}{m+1}\right].$$

- Proof: Let $S\sim D^{m+1}$ be a sample linearly separable and let $\mathbf{x}\in S$. If $h_{S-\{\mathbf{x}\}}$ misclassifies $\mathbf{x}$, then $\mathbf{x}$ must be a 'support vector' for h_S (update at $\mathbf{x}$). Thus,

$$\widehat{R}_{\mathrm{loo}}(\mathrm{perceptron}) \leq \frac{M(S)}{m+1}.$$

Perceptron - Non-Separable Bound

(MM and Rostamizadeh, 2013)

- Theorem: let I denote the set of rounds at which the Perceptron algorithm makes an update when processing $\mathbf{x}_1, \ldots, \mathbf{x}_T$ and let $M_T = |I|$. Then,

$$\boxed{M_T \leq \inf_{\rho > 0, \|\mathbf{u}\|_2 \leq 1} \left[\sqrt{L_\rho(\mathbf{u})} + \frac{R}{\rho} \right]^2},$$

where $R = \max_{t \in I} \|\mathbf{x}_t\|$

$$L_\rho(\mathbf{u}) = \sum_{t \in I} \left(1 - \frac{y_t(\mathbf{u} \cdot \mathbf{x}_t)}{\rho}\right)_+.$$

- Proof: for any t, $1-\frac{y_t(\mathbf{u}\cdot\mathbf{x}_t)}{\rho} \leq \big(1-\frac{y_t(\mathbf{u}\cdot\mathbf{x}_t)}{\rho}\big)_+$, summing up these inequalities for $t \in I$ yields:

$$\begin{aligned} M_T &\leq \sum_{t\in I}\left(1-\frac{y_t(\mathbf{u}\cdot\mathbf{x}_t)}{\rho}\right)_+ + \sum_{t\in I}\frac{y_t(\mathbf{u}\cdot\mathbf{x}_t)}{\rho} \\ &\leq L_\rho(\mathbf{u}) + \frac{\sqrt{M_T}R}{\rho}, \end{aligned}$$

  by upper-bounding $\sum_{t\in I}(y_t\mathbf{u}\cdot\mathbf{x}_t)$ as in the proof for the separable case.

- solving the second-degree inequality

$$M_T \leq L_\rho(\mathbf{u}) + \frac{\sqrt{M_T}R}{\rho},$$

  gives $\sqrt{M_T} \leq \dfrac{\frac{R}{\rho}+\sqrt{\frac{R^2}{\rho^2}+4L_\rho(\mathbf{u})}}{2} \leq \dfrac{R}{\rho}+\sqrt{L_\rho(\mathbf{u})}.$

Non-Separable Case - L2 Bound

(Freund and Schapire, 1998; MM and Rostamizadeh, 2013)

- **Theorem**: let I denote the set of rounds at which the Perceptron algorithm makes an update when processing $\mathbf{x}_1, \ldots, \mathbf{x}_T$ and let $M_T = |I|$. Then,

$$M_T \leq \inf_{\rho>0, \|u\|_2 \leq 1} \left[\frac{\|\mathbf{L}_\rho(\mathbf{u})\|_2}{2} + \sqrt{\frac{\|\mathbf{L}_\rho(\mathbf{u})\|_2^2}{4} + \frac{\sqrt{\sum_{t \in I} \|\mathbf{x}_t\|^2}}{\rho}} \right]^2.$$

  - when $\|\mathbf{x}_t\| \leq R$ for all $t \in I$, this implies

$$\boxed{M_T \leq \inf_{\rho>0, \|u\|_2 \leq 1} \left(\frac{R}{\rho} + \|\mathbf{L}_\rho(\mathbf{u})\|_2 \right)^2},$$

where $\mathbf{L}_\rho(\mathbf{u}) = \left[\left(1 - \frac{y_t(\mathbf{u} \cdot \mathbf{x}_t)}{\rho}\right)_+ \right]_{t \in I}$.

- Proof: Reduce problem to separable case in higher dimension. Let $l_t = \left(1 - \frac{y_t \mathbf{u}\cdot\mathbf{x}_t}{\rho}\right)_+ 1_{t \in I}$, for $t \in [1, T]$.
  - Mapping (similar to trivial mapping):

$$\mathbf{x}_t = \begin{bmatrix} x_{t,1} \\ \vdots \\ x_{t,N} \end{bmatrix} \rightarrow \mathbf{x}'_t = \begin{array}{r} \\ \\ \\ \\ \\ \\ (N+t)\text{th component} \\ \\ \\ \\ \end{array} \begin{bmatrix} x_{t,1} \\ \vdots \\ x_{t,N} \\ 0 \\ \vdots \\ 0 \\ \Delta \\ 0 \\ \vdots \\ 0 \end{bmatrix} \qquad \mathbf{u} \rightarrow \mathbf{u}' = \begin{bmatrix} \frac{u_1}{Z} \\ \vdots \\ \frac{u_N}{Z} \\ \frac{y_1 \rho l_1}{\Delta Z} \\ \vdots \\ \frac{y_T \rho l_T}{\Delta Z} \end{bmatrix}$$

$$\|\mathbf{u}'\| = 1 \implies Z = \sqrt{1 + \frac{\rho^2 \|\mathbf{L}_\rho(\mathbf{u})\|^2}{\Delta^2}}.$$

- Observe that the Perceptron algorithm makes the same predictions and makes updates at the same rounds when processing $\mathbf{x}'_1, \ldots, \mathbf{x}'_T$.
- For any $t \in I$,

$$
\begin{aligned}
y_t(\mathbf{u}' \cdot \mathbf{x}'_t) &= y_t\Big(\frac{\mathbf{u} \cdot \mathbf{x}_t}{Z} + \Delta \frac{y_t \rho l_t}{Z\Delta}\Big) \\
&= \frac{y_t \mathbf{u} \cdot \mathbf{x}_t}{Z} + \frac{\rho l_t}{Z} \\
&= \frac{1}{Z}\big(y_t \mathbf{u} \cdot \mathbf{x}_t + [\rho - y_t(\mathbf{u} \cdot \mathbf{x}_t)]_+\big) \geq \frac{\rho}{Z}.
\end{aligned}
$$

- Summing up and using the proof in the separable case yields:

$$
M_T \frac{\rho}{Z} \leq \sum_{t \in I} y_t(\mathbf{u}' \cdot \mathbf{x}'_t) \leq \sqrt{\sum_{t \in I} \|\mathbf{x}'_t\|^2}.
$$

- The inequality can be rewritten as

$$M_T^2 \leq \Big(\frac{1}{\rho^2}+\frac{\|\mathbf{L}_\rho(\mathbf{u})\|^2}{\Delta^2}\Big)\Big(r^2+M_T\Delta^2\Big)=\frac{r^2}{\rho^2}+\frac{r^2\|\mathbf{L}_\rho(\mathbf{u})\|^2}{\Delta^2}+\frac{M_T\Delta^2}{\rho^2}+M_T\|\,\mathbf{L}_\rho(\mathbf{u})\|^2_2;$$

where $r=\sqrt{\sum_{t\in I}\|\mathbf{x}_t\|^2}$.

- Selecting Δ to minimize the bound gives $\Delta^2=\frac{\rho\|\mathbf{L}_\rho(\mathbf{u})\|_2 r}{\sqrt{M_T}}$ and leads to

$$M_T^2 \leq \frac{r^2}{\rho^2}+2\frac{\sqrt{M_T}\|\mathbf{L}_\rho(\mathbf{u})\|r}{\rho}+M_T\|\mathbf{L}_\rho(\mathbf{u})\|^2=(\tfrac{r}{\rho}+\sqrt{M_T}\|\mathbf{L}_\rho(\mathbf{u})\|_2)^2.$$

- Solving the second-degree inequality

$$M_T-\sqrt{M_T}\|\,\mathbf{L}_\rho(\mathbf{u})\|_2-\frac{r}{\rho}\leq 0$$

yields directly the first statement. The second one results from replacing r with $\sqrt{M_T}R$.

Dual Perceptron Algorithm

$\textsc{Dual-Perceptron}(\boldsymbol{\alpha}^0)$

$\boldsymbol{\alpha} \leftarrow \boldsymbol{\alpha}^0 \qquad \triangleright$ typically $\boldsymbol{\alpha}^0 = \mathbf{0}$
for $t \leftarrow 1$ **to** T **do**
  $\textsc{Receive}(\mathbf{x}_t)$
  $\widehat{y}_t \leftarrow \text{sgn}(\sum_{s=1}^{T} \alpha_s y_s (\mathbf{x}_s \cdot \mathbf{x}_t))$
  $\textsc{Receive}(y_t)$
  if $(\widehat{y}_t \neq y_t)$ **then**
    $\alpha_t \leftarrow \alpha_t + 1$
return $\boldsymbol{\alpha}$

Kernel Perceptron Algorithm

(Aizerman et al., 1964)

K PDS kernel.

$\textsc{Kernel-Perceptron}(\boldsymbol{\alpha}^0)$

$\boldsymbol{\alpha} \leftarrow \boldsymbol{\alpha}^0 \qquad \triangleright$ typically $\boldsymbol{\alpha}^0 = \mathbf{0}$

for $t \leftarrow 1$ **to** T **do**

  $\textsc{Receive}(x_t)$

  $\widehat{y}_t \leftarrow \operatorname{sgn}(\sum_{s=1}^{T} \alpha_s y_s K(x_s, x_t))$

  $\textsc{Receive}(y_t)$

  if $(\widehat{y}_t \neq y_t)$ **then**

    $\alpha_t \leftarrow \alpha_t + 1$

return $\boldsymbol{\alpha}$

Winnow Algorithm

(Littlestone, 1988)

$\textsc{Winnow}(\eta)$

$w_1 \leftarrow \mathbf{1}/N$
for $t \leftarrow 1$ **to** T **do**
  $\textsc{Receive}(\mathbf{x}_t)$
  $\widehat{y}_t \leftarrow \operatorname{sgn}(\mathbf{w}_t \cdot \mathbf{x}_t)$ $\quad\triangleright\quad y_t \in \{-1, +1\}$
  $\textsc{Receive}(y_t)$
  if $(\widehat{y}_t \neq y_t)$ **then**
    $Z_t \leftarrow \sum_{i=1}^{N} w_{t,i} \exp(\eta y_t x_{t,i})$
    for $i \leftarrow 1$ **to** N **do**
      $w_{t+1,i} \leftarrow \dfrac{w_{t,i} \exp(\eta y_t x_{t,i})}{Z_t}$
  else $\mathbf{w}_{t+1} \leftarrow \mathbf{w}_t$
return $\mathbf{w}_{T+1}$

Notes

- Winnow $=$ weighted majority:
  - for $y_{t,i} = x_{t,i} \in \{-1, +1\}$, $\operatorname{sgn}(\mathbf{w}_t \cdot \mathbf{x}_t)$ coincides with the majority vote.
  - multiplying by e^{η} or $e^{-\eta}$ the weight of correct or incorrect experts, is equivalent to multiplying by $\beta = e^{-2\eta}$ the weight of incorrect ones.
- Relationships with other algorithms: e.g., boosting and Perceptron (Winnow and Perceptron can be viewed as special instances of a general family).

Winnow Algorithm - Bound

- **Theorem**: Assume that $\|x_t\|_\infty \leq R_\infty$ for all $t \in [1, T]$ and that for some $\rho_\infty > 0$ and $\mathbf{v} \in \mathbb{R}^N$, $\mathbf{v} \geq 0$ for all $t \in [1, T]$,

$$\rho_\infty \leq \frac{y_t(\mathbf{v} \cdot \mathbf{x}_t)}{\|\mathbf{v}\|_1}.$$

  Then, the number of mistakes made by the Winnow algorithm is bounded by $\boxed{2\left(R_\infty^2/\rho_\infty^2\right)\log N}$.

- **Proof**: Let I be the set of ts at which there is an update and let M be the total number of updates.

Notes

- Comparison with perceptron bound:
  - dual norms: norms for $\mathbf{x}_t$ and $\mathbf{v}$.
  - similar bounds with different norms.
  - each advantageous in different cases:
    - Winnow bound favorable when a sparse set of experts can predict well. For example, if $\mathbf{v} = \mathbf{e}_1$ and $\mathbf{x}_t \in \{\pm 1\}^N$, $\log N$ vs N.
    - Perceptron favorable in opposite situation.

Winnow Algorithm - Bound

- **Potential:** $\Phi_t = \sum_{i=1}^{N} \frac{v_i}{\|\mathbf{v}\|} \log \frac{v_i/\|\mathbf{v}\|}{w_{t,i}}$. (relative entropy)
- **Upper bound:** for each t in I,

$$\begin{aligned}
\Phi_{t+1} - \Phi_t &= \sum_{i=1}^{N} \frac{v_i}{\|\mathbf{v}\|_1} \log \frac{w_{t,i}}{w_{t+1,i}} \\
&= \sum_{i=1}^{N} \frac{v_i}{\|\mathbf{v}\|_1} \log \frac{Z_t}{\exp(\eta y_t x_{t,i})} \\
&= \log Z_t - \eta \sum_{i=1}^{N} \frac{v_i}{\|\mathbf{v}\|_1} y_t x_{t,i} \\
&\leq \log \Big[\sum_{i=1}^{N} w_{t,i} \exp(\eta y_t x_{t,i}) \Big] - \eta \rho_\infty \\
&= \log \mathop{\mathrm{E}}_{\mathbf{w}_t} \Big[\exp(\eta y_t x_t) \Big] - \eta \rho_\infty \\
\text{(Hoeffding)} &\leq \log \Big[\exp(\eta^2 (2R_\infty)^2/8) \Big] + \underbrace{\eta y_t \mathbf{w}_t \cdot \mathbf{x}_t}_{\leq 0} - \eta \rho_\infty \\
&\leq \eta^2 R_\infty^2/2 - \eta \rho_\infty .
\end{aligned}$$

Winnow Algorithm - Bound

- **Upper bound:** summing up the inequalities yields

$$\Phi_{T+1} - \Phi_1 \leq M(\eta^2 R_\infty^2/2 - \eta\rho_\infty).$$

- **Lower bound:** note that

$$\Phi_1 = \sum_{i=1}^N \frac{v_i}{\|\mathbf{v}\|_1} \log \frac{v_i/\|\mathbf{v}\|_1}{1/N} = \log N + \sum_{i=1}^N \frac{v_i}{\|\mathbf{v}\|_1} \log \frac{v_i}{\|\mathbf{v}\|_1} \leq \log N$$

and for all $t, \Phi_t \geq 0$ (property of relative entropy).

Thus, $\Phi_{T+1} - \Phi_1 \geq 0 - \log N = -\log N.$

- **Comparison:** $-\log N \leq M(\eta^2 R_\infty^2/2 - \eta\rho_\infty)$. For $\eta = \frac{\rho_\infty}{R_\infty^2}$ we obtain

$$M \leq 2 \log N \frac{R_\infty^2}{\rho_\infty^2}.$$

Conclusion

- On-line learning:
  - wide and fast-growing literature.
  - many related topics, e.g., game theory, text compression, convex optimization.
  - online to batch bounds and techniques.
  - online version of batch algorithms, e.g., regression algorithms (see regression lecture).

References

- Aizerman, M. A., Braverman, E. M., & Rozonoer, L. I. (1964). Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control, 25*, 821-837.

- Nicolò Cesa-Bianchi, Alex Conconi, Claudio Gentile: On the Generalization Ability of On-Line Learning Algorithms. *IEEE Transactions on Information Theory* 50(9): 2050-2057. 2004.

- Nicolò Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge University Press, 2006.

- Yoav Freund and Robert Schapire. Large margin classification using the perceptron algorithm. In *Proceedings of COLT 1998*. ACM Press, 1998.

- Nick Littlestone. From On-Line to Batch Learning. COLT 1989: 269-284.

- Nick Littlestone. "Learning Quickly When Irrelevant Attributes Abound: A New Linear-threshold Algorithm" *Machine Learning* 285-318(2). 1988.

References

- Nick Littlestone, Manfred K. Warmuth: The Weighted Majority Algorithm. *FOCS 1989*: 256-261.
- Tom Mitchell. *Machine Learning*, McGraw Hill, 1997.
- Mehryar Mohri and Afshin Rostamizadeh. Perceptron Mistake Bounds. arXiv:1305.0208, 2013.
- Novikoff, A. B. (1962). On convergence proofs on perceptrons. *Symposium on the Mathematical Theory of Automata, 12*, 615-622. Polytechnic Institute of Brooklyn.
- Rosenblatt, Frank, *The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain*, Cornell Aeronautical Laboratory, Psychological Review, v65, No. 6, pp. 386-408, 1958.
- Vladimir N. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, New York, 1998.

Appendix

SVMs - Leave-One-Out Analysis

(Vapnik, 1995)

- **Theorem**: let h_S be the optimal hyperplane for a sample S and let $N_{\mathrm{SV}}(S)$ be the number of support vectors defining h_S. Then,

$$\mathop{\mathrm{E}}_{S \sim D^m}[R(h_S)] \leq \mathop{\mathrm{E}}_{S \sim D^{m+1}}\left[\frac{\min(N_{\mathrm{SV}}(S), R^2_{m+1}/\rho^2_{m+1})}{m+1}\right].$$

- **Proof**: one part proven in lecture 4. The other part due to $\alpha_i \geq 1/R^2_{m+1}$ for $\mathbf{x}_i$ misclassified by SVMs.

Comparison

- Bounds on expected error, not high probability statements.
- Leave-one-out bounds not sufficient to distinguish SVMs and perceptron algorithm. Note however:
  - same maximum margin ρ_{m+1} can be used in both.
  - but different radius R_{m+1} of support vectors.
- Difference: margin distribution.

Foundations of Machine Learning
Ranking

Mehryar Mohri
Courant Institute and Google Research
mohri@cims.nyu.edu

Motivation

- Very large data sets:
  - too large to display or process.
  - limited resources, need priorities.
  - $\longrightarrow$ ranking more desirable than classification.
- Applications:
  - search engines, information extraction.
  - decision making, auctions, fraud detection.
- Can we learn to predict ranking accurately?

Related Problem

- Rank aggregation: given n candidates and k voters each giving a ranking of the candidates, find ordering as close as possible to these.
  - closeness measured in number of pairwise misrankings.
  - problem NP-hard even for $k=4$ (Dwork et al., 2001).

This Talk

- Score-based ranking
- Preference-based ranking

Score-Based Setting

- Single stage: learning algorithm
  - receives labeled sample of pairwise preferences;
  - returns scoring function $h\colon U \rightarrow \mathbb{R}$.
- Drawbacks:
  - h induces a linear ordering for full set U.
  - does not match a query-based scenario.
- Advantages:
  - efficient algorithms.
  - good theory: VC bounds, margin bounds, stability bounds (FISS 03, RCMS 05, AN 05, AGHHR 05, CMR 07).

Score-Based Ranking

- **Training data**: sample of i.i.d. labeled pairs drawn from $U \times U$ according to some distribution D,

$$S = \Big((x_1, x'_1, y_1), \ldots, (x_m, x'_m, y_m)\Big) \in U \times U \times \{-1, 0, +1\},$$

with $y_i = \begin{cases} +1 & \text{if } x'_i >_{\text{pref}} x_i \\ 0 & \text{if } x_i =_{\text{pref}} x'_i \text{ or no information} \\ -1 & \text{if } x'_i <_{\text{pref}} x_i. \end{cases}$

- **Problem**: find hypothesis $h : U \to \mathbb{R}$ in H with small generalization error

$$R(h) = \Pr_{(x,x') \sim D} \Big[(f(x, x') \neq 0) \wedge (f(x, x')\big(h(x') - h(x)\big) \leq 0) \Big].$$

Notes

- Empirical error:

$$\widehat{R}(h) = \frac{1}{m} \sum_{i=1}^{m} 1_{(y_i \neq 0) \wedge (y_i(h(x'_i) - h(x_i)) \leq 0)} \,.$$

- The relation $x \,\mathcal{R}\, x' \Leftrightarrow f(x, x') = 1$ may be non-transitive (needs not even be anti-symmetric).
- Problem different from classification.

Distributional Assumptions

- Distribution over points: m points (literature).
  - labels for pairs.
  - $\longrightarrow$ squared number of examples $O(m^2)$.
  - dependency issue.
- Distribution over pairs: m pairs.
  - label for each pair received.
  - independence assumption.
  - same (linear) number of examples.

Confidence Margin in Ranking

- Labels assumed to be in $\{+1, -1\}$.
- Empirical margin loss for ranking: for $\rho > 0$,

$$\widehat{R}_\rho(h) = \frac{1}{m} \sum_{i=1}^{m} \Phi_\rho\Big(y_i\big(h(x'_i) - h(x_i)\big)\Big).$$

$$\widehat{R}_\rho(h) \leq \frac{1}{m} \sum_{i=1}^{m} 1_{y_i[h(x'_i) - h(x_i)] \leq \rho}.$$

Marginal Rademacher Complexities

- Distributions:
  - D_1 marginal distribution with respect to the first element of the pairs;
  - D_2 marginal distribution with respect to second element of the pairs.
- Samples: $S_1 = \big((x_1, y_1), \ldots, (x_m, y_m)\big)$
  $S_2 = \big((x'_1, y_1), \ldots, (x'_m, y_m)\big).$
- Marginal Rademacher complexities:

$$\mathfrak{R}_m^{D_1}(H) = \mathrm{E}[\widehat{\mathfrak{R}}_{S_1}(H)] \qquad \mathfrak{R}_m^{D_2}(H) = \mathrm{E}[\widehat{\mathfrak{R}}_{S_2}(H)].$$

Ranking Margin Bound

(Boyd, Cortes, MM, and Radovanovich 2012; MM, Rostamizadeh, and Talwalkar, 2012)

- Theorem: let H be a family of real-valued functions. Fix $\rho > 0$, then, for any $\delta > 0$, with probability at least $1-\delta$ over the choice of a sample of size m, the following holds for all $h \in H$:

$$R(h) \leq \widehat{R}_\rho(h) + \frac{2}{\rho}\Big(\mathfrak{R}_m^{D_1}(H) + \mathfrak{R}_m^{D_2}(H)\Big) + \sqrt{\frac{\log \frac{1}{\delta}}{2m}}.$$

Proof

- **Define:** $\widetilde{\mathcal{H}} = \{z = ((x, x'), y) \mapsto y[h(x') - h(x)] \colon h \in \mathcal{H}\}$. Then, by the general margin bound, with probability at least $1-\delta$,

$$\mathbb{E}\left[\Phi_\rho(y[h(x') - h(x)])\right] \leq \widehat{R}_{S,\rho}(h) + 2\mathfrak{R}_m\big(\Phi_\rho \circ \widetilde{\mathcal{H}}\big) + \sqrt{\frac{\log \frac{1}{\delta}}{2m}}.$$

- We have $\mathfrak{R}_m(\Phi_\rho \circ \widehat{\mathcal{H}}) \leq \frac{1}{\rho}\mathfrak{R}_m(\widehat{\mathcal{H}})$ and

$$\begin{aligned}
\mathfrak{R}_m(\widetilde{\mathcal{H}}) &= \frac{1}{m} \mathop{\mathbb{E}}_{S,\sigma}\left[\sup_{h \in \mathcal{H}} \sum_{i=1}^m \sigma_i y_i (h(x'_i) - h(x_i))\right] \\
&= \frac{1}{m} \mathop{\mathbb{E}}_{S,\sigma}\left[\sup_{h \in \mathcal{H}} \sum_{i=1}^m \sigma_i (h(x'_i) - h(x_i))\right] && (y_i\sigma_i \text{ and } \sigma_i\text{: same distrib.}) \\
&\leq \frac{1}{m} \mathop{\mathbb{E}}_{S,\sigma}\left[\sup_{h \in \mathcal{H}} \sum_{i=1}^m \sigma_i h(x'_i) + \sup_{h \in \mathcal{H}} \sum_{i=1}^m \sigma_i h(x_i)\right] && (\text{by sub-additivity of sup}) \\
&= \mathop{\mathbb{E}}_{S}\left[\mathfrak{R}_{S_2}(\mathcal{H}) + \mathfrak{R}_{S_1}(\mathcal{H})\right] && (\text{definition of } S_1 \text{ and } S_2).
\end{aligned}$$

Ranking with SVMs

see for example (Joachims, 2002)

- Optimization problem: application of SVMs.

$$\min_{\mathbf{w},\boldsymbol{\xi}} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m}\xi_i$$

$$\text{subject to: } y_i\Big[\mathbf{w}\cdot\big(\boldsymbol{\Phi}(x_i')-\boldsymbol{\Phi}(x_i)\big)\Big] \geq 1-\xi_i$$

$$\xi_i \geq 0, \quad \forall i \in [1,m].$$

- Decision function:

$$h\colon x \mapsto \mathbf{w}\cdot\boldsymbol{\Phi}(x)+b.$$

Notes

- The algorithm coincides with SVMs using feature mapping

$$(x, x') \mapsto \mathbf{\Psi}(x, x') = \mathbf{\Phi}(x') - \mathbf{\Phi}(x).$$

- Can be used with kernels:

$$\begin{aligned} K'((x_i, x'_i), (x_j, x'_j)) &= \mathbf{\Psi}(x_i, x'_i) \cdot \mathbf{\Psi}(x_j, x'_j) \\ &= K(x_i, x_j) + K(x'_i, x'_j) - K(x'_i, x_j) - K(x_i, x'_j). \end{aligned}$$

- Algorithm directly based on margin bound.

Boosting for Ranking

- Use weak ranking algorithm and create stronger ranking algorithm.
- Ensemble method: combine base rankers returned by weak ranking algorithm.
- Finding simple relatively accurate base rankers often not hard.
- How should base rankers be combined?

CD RankBoost

(Freund et al., 2003; Rudin et al., 2005)

$$H \subseteq \{0,1\}^X. \epsilon_t^0 + \epsilon_t^+ + \epsilon_t^- = 1, \epsilon_t^s(h) = \Pr_{(x,x')\sim D_t}\Big[\operatorname{sgn}\big(f(x,x')(h(x')-h(x))\big) = s\Big].$$

$\textsc{RankBoost}(S = ((x_1, x_1', y_1) \ldots, (x_m, x_m', y_m)))$

$$
\begin{array}{l}
\textbf{for } i \leftarrow 1 \textbf{ to } m \textbf{ do}\\
\qquad D_1(x_i, x_i') \leftarrow \frac{1}{m}\\
\textbf{for } t \leftarrow 1 \textbf{ to } T \textbf{ do}\\
\qquad h_t \leftarrow \text{base ranker in } H \text{ with smallest } \epsilon_t^- - \epsilon_t^+ = -\operatorname{E}_{i\sim D_t}\Big[y_i\big(h_t(x_i') - h_t(x_i)\big)\Big]\\
\qquad \alpha_t \leftarrow \frac{1}{2}\log\frac{\epsilon_t^+}{\epsilon_t^-}\\
\qquad Z_t \leftarrow \epsilon_t^0 + 2[\epsilon_t^+ \epsilon_t^-]^{\frac{1}{2}} \qquad \triangleright \text{normalization factor}\\
\qquad \textbf{for } i \leftarrow 1 \textbf{ to } m \textbf{ do}\\
\qquad\qquad D_{t+1}(x_i, x_i') \leftarrow \frac{D_t(x_i, x_i')\exp\Big[-\alpha_t y_i\Big(h_t(x_i') - h_t(x_i)\Big)\Big]}{Z_t}\\
\varphi_T \leftarrow \sum_{t=1}^T \alpha_t h_t\\
\textbf{return } \varphi_T
\end{array}
$$

Notes

- Distributions D_t over pairs of sample points:
  - originally uniform.
  - at each round, the weight of a misclassified example is increased.
  - observation: $D_{t+1}(x, x') = \frac{e^{-y[\varphi_t(x') - \varphi_t(x)]}}{|S| \prod_{s=1}^t Z_s}$, since

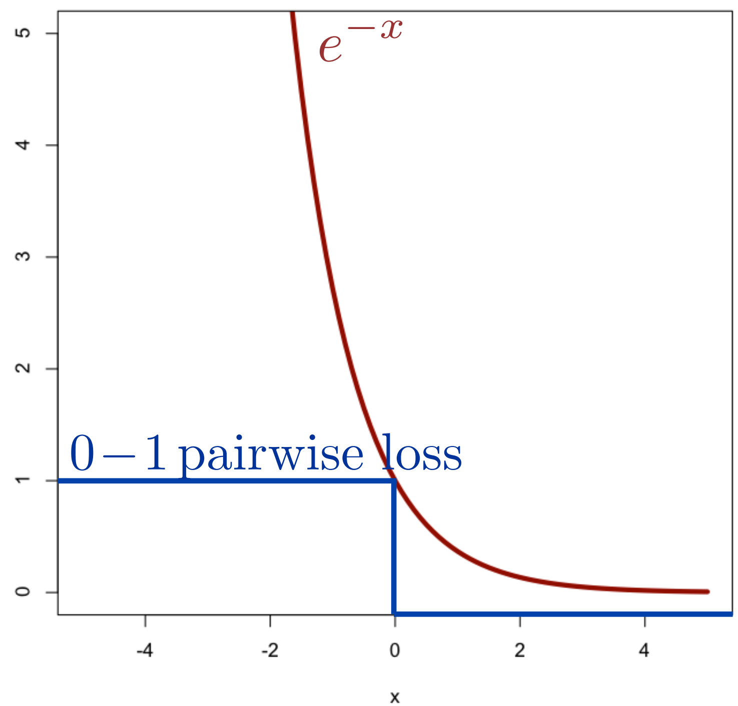
$$D_{t+1}(x, x') = \frac{D_t(x, x') e^{-y\alpha_t[h_t(x') - h_t(x)]}}{Z_t} = \frac{1}{|S|} \frac{e^{-y \sum_{s=1}^t \alpha_s [h_s(x') - h_s(x)]}}{\prod_{s=1}^t Z_s}.$$

- weight assigned to base classifier h_t: α_t directly depends on the accuracy of h_t at round t .

Coordinate Descent RankBoost

- **Objective Function**: convex and differentiable.

$$F(\boldsymbol{\alpha}) = \sum_{(x,x',y)\in S} e^{-y[\varphi_T(x')-\varphi_T(x)]} = \sum_{(x,x',y)\in S} \exp\Big(-y\sum_{t=1}^{T}\alpha_t[h_t(x')-h_t(x)]\Big).$$



- Direction: unit vector $\mathbf{e}_t$ with

$$\mathbf{e}_t = \operatorname*{argmin}_t \left.\frac{dF(\boldsymbol{\alpha} + \eta\mathbf{e}_t)}{d\eta}\right|_{\eta=0}.$$

- Since $F(\boldsymbol{\alpha} + \eta\mathbf{e}_t) = \sum_{(x,x',y)\in S} e^{-y\sum_{s=1}^T \alpha_s[h_s(x')-h_s(x)]} e^{-y\eta[h_t(x')-h_t(x)]}$,

$$\begin{aligned}
\left.\frac{dF(\alpha + \eta e_t)}{d\eta}\right|_{\eta=0} &= -\sum_{(x,x',y)\in S} y[h_t(x') - h_t(x)] \exp\Big[-y\sum_{s=1}^T \alpha_s[h_s(x') - h_s(x)]\Big] \\
&= -\sum_{(x,x',y)\in S} y[h_t(x') - h_t(x)] D_{T+1}(x,x')\Big[m\prod_{s=1}^T Z_s\Big] \\
&= \boxed{-[\epsilon_t^+ - \epsilon_t^-]}\Big[m\prod_{s=1}^T Z_s\Big].
\end{aligned}$$

Thus, direction corresponding to base classifier selected by the algorithm.

- **Step size**: obtained via

$$\frac{dF(\boldsymbol{\alpha} + \eta \mathbf{e}_t)}{d\eta} = 0$$

$$\Leftrightarrow -\sum_{(x,x',y)\in S} y[h_t(x') - h_t(x)] \exp\Big[-y\sum_{s=1}^{T} \alpha_s [h_s(x') - h_s(x)]\Big] e^{-y[h_t(x')-h_t(x)]\eta} = 0$$

$$\Leftrightarrow -\sum_{(x,x',y)\in S} y[h_t(x') - h_t(x)] D_{T+1}(x,x') \Big[m \prod_{s=1}^{T} Z_s \Big] e^{-y[h_t(x')-h_t(x)]\eta} = 0$$

$$\Leftrightarrow -\sum_{(x,x',y)\in S} y[h_t(x') - h_t(x)] D_{T+1}(x,x') e^{-y[h_t(x')-h_t(x)]\eta} = 0$$

$$\Leftrightarrow -[\epsilon_t^+ e^{-\eta} - \epsilon_t^- e^{\eta}] = 0$$

$$\Leftrightarrow \boxed{\eta = \frac{1}{2}\log\frac{\epsilon_t^+}{\epsilon_t^-}}.$$

Thus, step size matches base classifier weight used in algorithm.

Bipartite Ranking

- Training data:
  - sample of negative points drawn according to D_-
$$S_- = (x_1, \ldots, x_m) \in U.$$
  - sample of positive points drawn according to D_+
$$S_+ = (x'_1, \ldots, x'_{m'}) \in U.$$

- Problem: find hypothesis $h \colon U \to \mathbb{R}$ in H with small generalization error

$$R_D(h) = \Pr_{x \sim D_-, x' \sim D_+} \Big[h(x') < h(x) \Big].$$

Properties

- Connection between AdaBoost and RankBoost (Cortes & MM, 04; Rudin et al., 05).
  - if constant base ranker used.
  - relationship between objective functions.
- More efficient algorithm in this special case (Freund et al., 2003).
- Bipartite ranking results typically reported in terms of AUC.

AdaBoost and CD RankBoost

- Objective functions: comparison.

$$
\begin{aligned}
F_{\text{Ada}}(\boldsymbol{\alpha}) &= \sum_{x_i \in S_- \cup S_+} \exp\left(-y_i f(x_i)\right) \\
&= \sum_{x_i \in S_-} \exp\left(+f(x_i)\right) + \sum_{x_i \in S_+} \exp\left(-f(x_i)\right) \\
&= F_-(\alpha) + F_+(\alpha).
\end{aligned}
$$

$$
\begin{aligned}
F_{\text{Rank}}(\boldsymbol{\alpha}) &= \sum_{(i,j) \in S_- \times S_+} \exp\Big(-\left[f(x_j) - f(x_i)\right]\Big) \\
&= \sum_{(i,j) \in S_- \times S_+} \exp\left(+f(x_i)\right) \exp\left(-f(x_j)\right) \\
&= F_-(\alpha) F_+(\alpha).
\end{aligned}
$$

AdaBoost and CD RankBoost

(Rudin et al., 2005)

- **Property:** AdaBoost (non-separable case).
  - constant base learner $h=1$ $\longrightarrow$ equal contribution of positive and negative points (in the limit).
  - consequence: AdaBoost asymptotically achieves optimum of CD RankBoost objective.
- **Observations:** if $F_+(\boldsymbol{\alpha}) = F_-(\boldsymbol{\alpha})$,

$$\begin{aligned} d(F_{\text{Rank}}) &= F_+ d(F_-) + F_- d(F_+) \\ &= F_+ \big(d(F_-) + d(F_+) \big) \\ &= F_+ d(F_{\text{Ada}}). \end{aligned}$$

Bipartite RankBoost - Efficiency

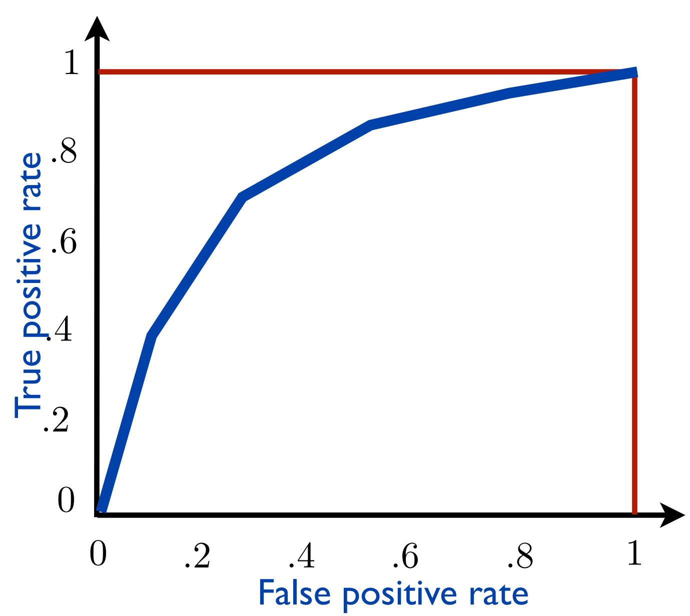
- Decomposition of distribution: for $(x, x') \in (S_-, S_+)$,
$$D(x, x') = D_-(x) D_+(x').$$
- Thus,
$$\begin{aligned} D_{t+1}(x, x') &= \frac{D_t(x, x') e^{-\alpha_t [h_t(x') - h_t(x)]}}{Z_t} \\ &= \frac{D_{t,-}(x) e^{\alpha_t h_t(x)}}{Z_{t,-}} \frac{D_{t,+}(x') e^{-\alpha_t h_t(x')}}{Z_{t,+}}, \end{aligned}$$

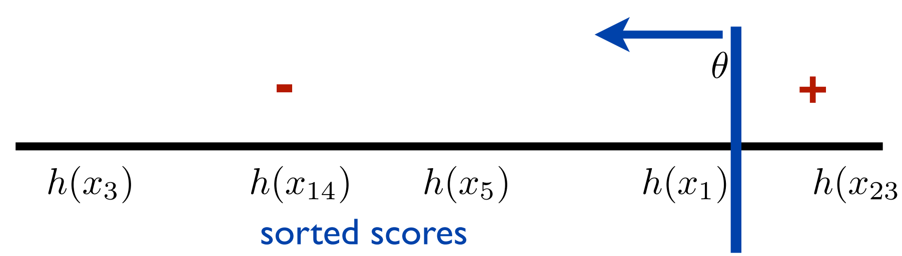
with $Z_{t,-} = \sum_{x \in S_-} D_{t,-}(x) e^{\alpha_t h_t(x)} \quad Z_{t,+} = \sum_{x' \in S_+} D_{t,+}(x') e^{-\alpha_t h_t(x')}.$

ROC Curve

(Egan, 1975)

- Definition: the receiver operating characteristic (ROC) curve is a plot of the true positive rate (TP) vs. false positive rate (FP).
  - TP: % positive points correctly labeled positive.
  - FP: % negative points incorrectly labeled positive.

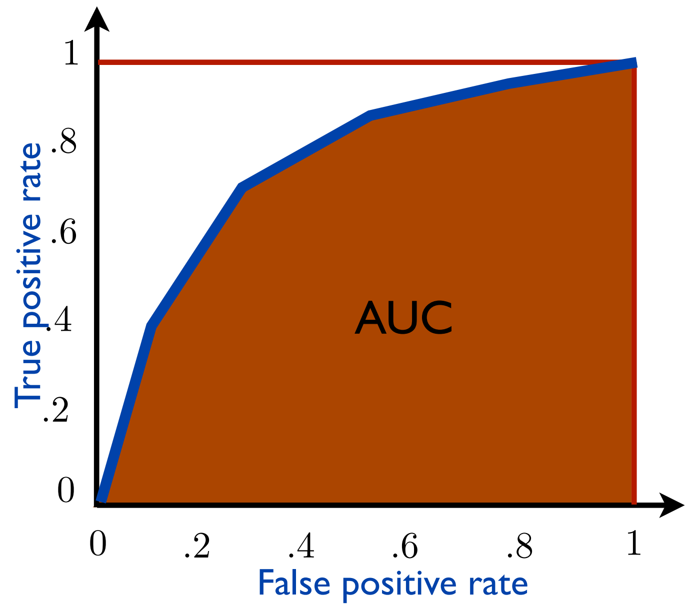




Area under the ROC Curve (AUC)

(Hanley and McNeil, 1982)

- Definition: the AUC is the area under the ROC curve. Measure of ranking quality.



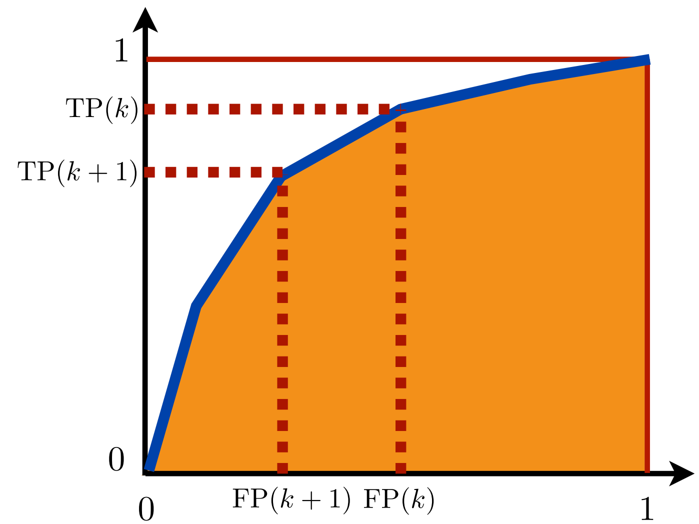
- Equivalently,

$$\mathrm{AUC}(h) = \frac{1}{m_- m_+} \sum_{i=1}^{m_-} \sum_{j=1}^{m_+} 1_{h(x_i) < h(x'_j)} = \Pr_{\substack{x \sim \widehat{D}_- \\ x' \sim \widehat{D}_+}} [h(x') > h(x)]$$

$$= 1 - \widehat{R}(h).$$

Proof

$$\begin{aligned}
\mathrm{AUC} &= \sum_{k=1}^{m-1} \frac{[\mathrm{TP}(k)+\mathrm{TP}(k+1)][\mathrm{FP}(k)-\mathrm{FP}(k+1)]}{2} && \text{(trapezoid area)}\\
&= \sum_{k=1}^{m-1} \frac{\sum_{l=k+1}^{m} 1_{y_l=+1} + \frac{1}{2} 1_{y_k=+1}}{m_+} \frac{1_{y_k=-1}}{m_-}\\
&= \frac{1}{m_+ m_-} \sum_{k=1}^{m-1} \sum_{l=k+1}^{m} 1_{y_l=+1} 1_{y_k=-1} && (1_{y_k=+1} 1_{y_k=-1} = 0)\\
&= \frac{1}{m_+ m_-} \sum_{k=1}^{m} \sum_{l=1}^{m} 1_{y_k=-1} 1_{y_l=+1} 1_{k<l}\\
&= \frac{1}{m_- m_+} \sum_{i=1}^{m_-} \sum_{j=1}^{m_+} 1_{h(x_i)<h(x'_j)}.
\end{aligned}$$



$$\mathrm{TP}(k) = \frac{\sum_{i=k}^{m} 1_{y_i=+1}}{m_+}$$

$$\mathrm{FP}(k) = \frac{\sum_{i=k}^{m} 1_{y_i=-1}}{m_-}$$

This Talk

- Score-based ranking
- Preference-based ranking

Preference-Based Setting

- Definitions:
  - U: universe, full set of objects.
  - V: finite query subset to rank, $V \subseteq U$.
  - τ^*: target ranking for V (random variable).
- Two stages: can be viewed as a reduction.
  - learn preference function $h \colon U \times U \to [0, 1]$.
  - given V, use h to determine ranking σ of V.
- Running-time: measured in terms of |calls to h|.

Preference-Based Ranking Problem

- Training data: pairs (V, τ^*) sampled i.i.d. according to D:

$$(V_1, \tau_1^*), (V_2, \tau_2^*), \ldots, (V_m, \tau_m^*) \qquad V_i \subseteq U.$$

subsets ranked by different labelers.

learn classifier

preference function $h \colon U \times U \to [0, 1]$.

- Problem: for any query set $V \subseteq U$, use h to return ranking σ_h close to target τ^* with small average error

$$R(h, \sigma) = \mathop{\mathrm{E}}_{(V,\tau^*)\sim D}[L(\sigma_{h,V}, \tau^*)].$$

Preference Function

- $h(u, v)$ close to 1 when u preferred to v, close to 0 otherwise. For the analysis, $h(u, v) \in \{0, 1\}$.
- Assumed pairwise consistent:
$$h(u, v) + h(v, u) = 1.$$
- May be non-transitive, e.g., we may have
$$h(u, v) = h(v, w) = h(w, u) = 1.$$
- Output of classifier or 'black-box'.

Loss Functions

(for fixed (V, τ^*))

- Preference loss:

$$L(h, \tau^*) = \frac{2}{n(n-1)} \sum_{u \neq v} h(u, v)\tau^*(v, u).$$

- Ranking loss:

$$L(\sigma, \tau^*) = \frac{2}{n(n-1)} \sum_{u \neq v} \sigma(u, v)\tau^*(v, u).$$

(Weak) Regret

- Preference regret:

$$\mathcal{R}'_{class}(h) = \mathop{\mathbb{E}}_{V,\tau^*}\left[L(h_{|V}, \tau^*)\right] - \mathop{\mathbb{E}}_{V}\left[\min_{\tilde{h}} \mathop{\mathbb{E}}_{\tau^*|V}\left[L(\tilde{h}, \tau^*)\right]\right].$$

- Ranking regret:

$$\mathcal{R}'_{rank}(A) = \mathop{\mathbb{E}}_{V,\tau^*,s}\left[L(A_s(V), \tau^*)\right] - \mathop{\mathbb{E}}_{V}\left[\min_{\tilde{\sigma}\in S(V)} \mathop{\mathbb{E}}_{\tau^*|V}\left[L(\tilde{\sigma}, \tau^*)\right]\right].$$

Deterministic Algorithm

(Balcan et al., 07)

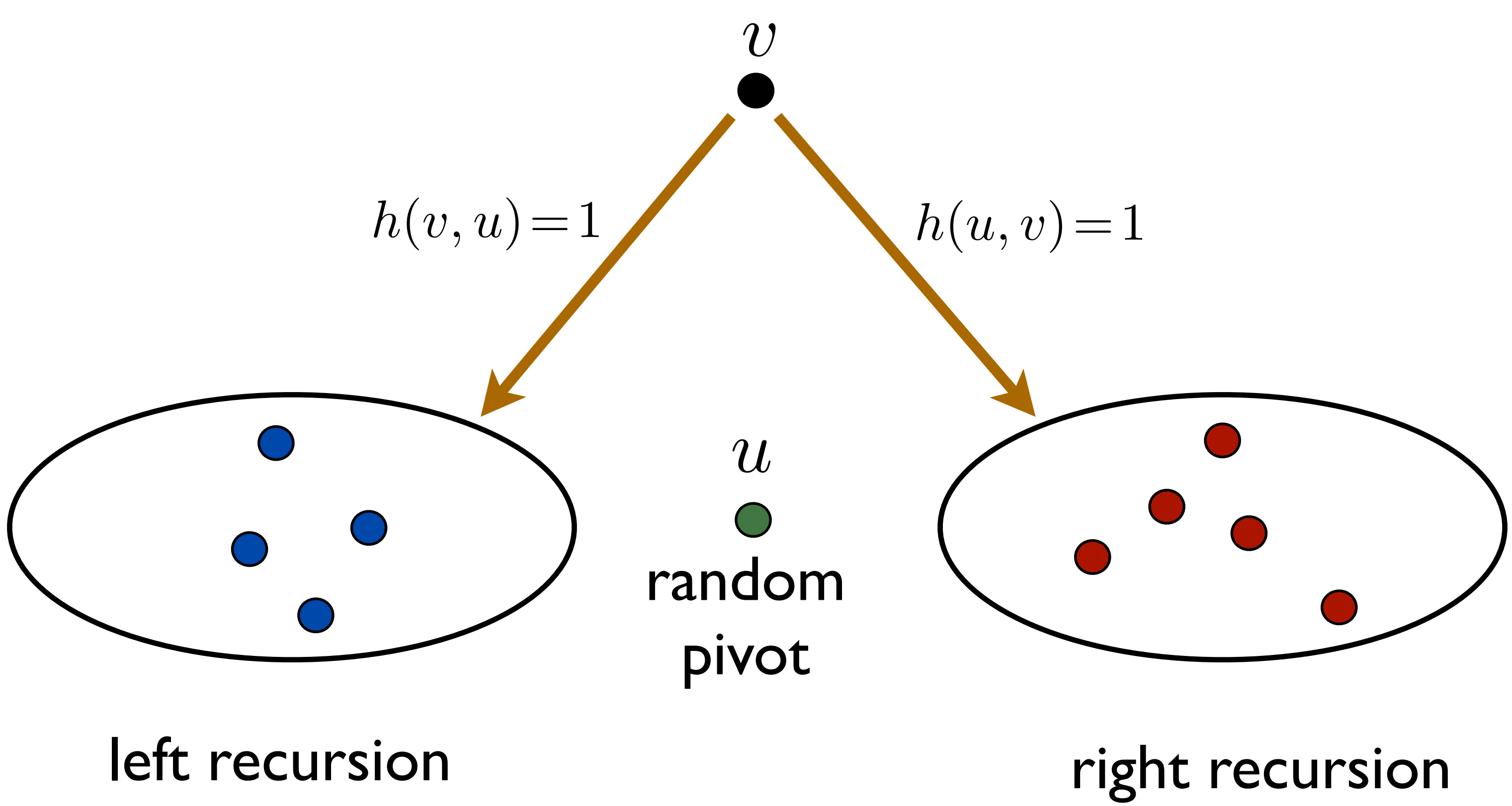
- Stage one: standard classification. Learn preference function $h : U \times U \to [0, 1]$.
- Stage two: sort-by-degree using comparison function h.
  - sort by number of points ranked below.
  - quadratic time complexity $O(n^2)$.

Randomized Algorithm

(Ailon & MM, 08)

- Stage one: standard classification. Learn preference function $h\colon U\times U\to[0,1]$.
- Stage two: randomized QuickSort (Hoare, 61) using h as comparison function.
  - comparison function non-transitive unlike textbook description.
  - but, time complexity shown to be $O(n\log n)$ in general.

Randomized QS



Deterministic Algo. - Bipartite Case

$(V = V_+ \cup V_-)$ (Balcan et al., 07)

- Bounds: for deterministic sort-by-degree algorithm
  - expected loss:

$$\mathop{\mathrm{E}}_{V,\tau^*}[L(A(V), \tau^*)] \leq 2 \mathop{\mathrm{E}}_{V,\tau^*}[L(h, \tau^*)].$$

  - regret:

$$\mathcal{R}'_{rank}(A(V)) \leq 2\, \mathcal{R}'_{class}(h).$$

- Time complexity: $\Omega(|V|^2)$.

Randomized Algo. - Bipartite Case

$(V = V_+ \cup V_-)$ (Ailon & MM, 08)

- Bounds: for randomized QuickSort.
  - expected loss (equality):

$$\mathop{\mathrm{E}}_{V,\tau^*,s}[L(Q_s^h(V), \tau^*)] = \mathop{\mathrm{E}}_{V,\tau^*}[L(h, \tau^*)].$$

  - regret:

$$\mathcal{R}'_{rank}(Q_s^h(\cdot)) \leq \mathcal{R}'_{class}(h) \quad .$$

- Time complexity:
  - full set: $O(n \log n)$.
  - top *k*: $O(n + k \log k)$.

Proof Ideas

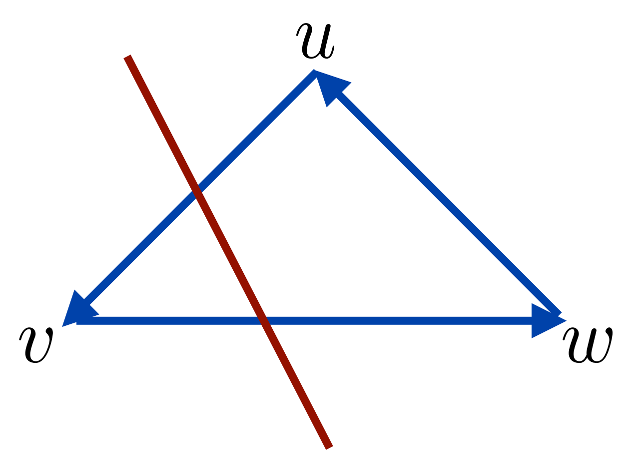
- QuickSort decomposition:

$$p_{uv} + \frac{1}{3} \sum_{w \notin \{u,v\}} p_{uvw} \Big(h(u,w)h(w,v) + h(v,w)h(w,u) \Big) = 1.$$

- Bipartite property:

$$\tau^*(u,v) + \tau^*(v,w) + \tau^*(w,u) =$$

$$\tau^*(v,u) + \tau^*(w,v) + \tau^*(u,w).$$



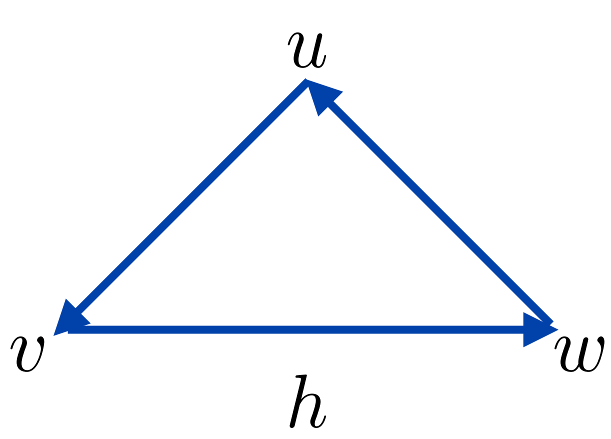
Lower Bound

- **Theorem**: for any deterministic algorithm A, there is a bipartite distribution for which

$$\mathcal{R}_{rank}(A) \geq 2\,\mathcal{R}_{class}(h).$$

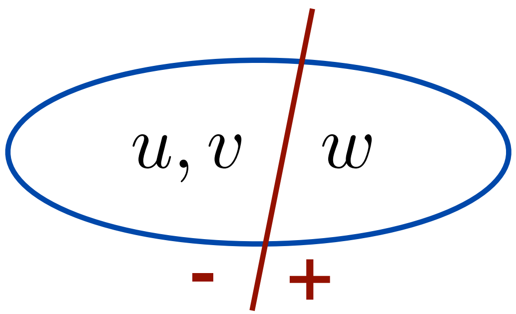
  - thus, factor of 2 $=$ best in deterministic case.
  - randomization necessary for better bound.
- **Proof**: take simple case $U = V = \{u, v, w\}$ and assume that h induces a cycle.
  - up to symmetry, A returns

    u, v, w or w, v, u.

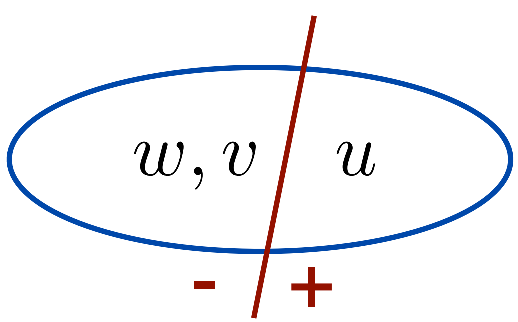


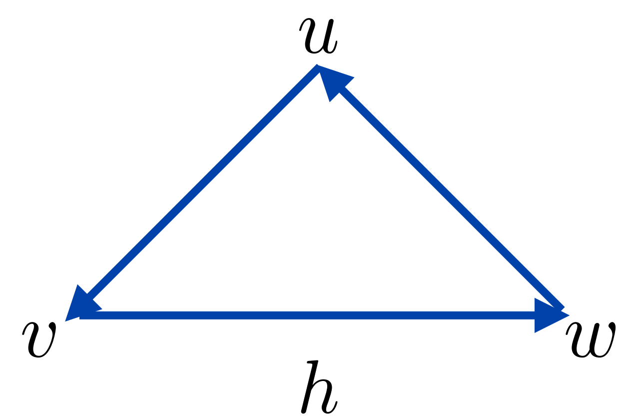
Lower Bound

- If A returns u, v, w, then choose τ^* as:



- If A returns w, v, u, then choose τ^* as:





$$L[h, \tau^*] = \frac{1}{3};$$

$$L[A, \tau^*] = \frac{2}{3}.$$

Guarantees - General Case

- Loss bound for QuickSort:

$$\underset{V,\tau^*,s}{\mathrm{E}}[L(Q_s^h(V),\tau^*)] \leq 2 \underset{V,\tau^*}{\mathrm{E}}[L(h,\tau^*)].$$

- Comparison with optimal ranking (see (CSS 99)):

$$\mathrm{E}_s[L(Q_s^h(V),\sigma_{optimal})] \leq 2\,L(h,\sigma_{optimal})$$

$$\mathrm{E}_s[L(h,Q_s^h(V))] \leq 3\,L(h,\sigma_{optimal}),$$

where $\sigma_{optimal} = \underset{\sigma}{\operatorname{argmin}}\, L(h,\sigma).$

Weight Function

- Generalization:

$$\tau^*(u,v) = \sigma^*(u,v)\,\omega(\sigma^*(u),\sigma^*(v)).$$

- Properties: needed for all previous results to hold,
  - symmetry: $\omega(i,j) = \omega(j,i)$ for all i,j.
  - monotonicity: $\omega(i,j), \omega(j,k) \leq \omega(i,k)$ for $i < j < k$.
  - triangle inequality: $\omega(i,j) \leq \omega(i,k) + \omega(k,j)$ for all triplets i,j,k.

Weight Function - Examples

- Kemeny: $w(i,j) = 1, \ \forall i, j.$
- Top-*k*: $w(i,j) = \begin{cases} 1 & \text{if } i \leq k \text{ or } j \leq k; \\ 0 & \text{otherwise.} \end{cases}$
- Bipartite: $w(i,j) = \begin{cases} 1 & \text{if } i \leq k \text{ and } j > k; \\ 0 & \text{otherwise.} \end{cases}$
- *k*-partite: can be defined similarly.

(Strong) Regret Definitions

- Ranking regret:

$$\mathcal{R}_{rank}(A) = \mathop{\mathrm{E}}_{V,\tau^*,s}[L(A_s(V), \tau^*)] - \min_{\tilde{\sigma}} \mathop{\mathrm{E}}_{V,\tau^*}[L(\tilde{\sigma}_{|V}, \tau^*)].$$

- Preference regret:

$$\mathcal{R}_{class}(h) = \mathop{\mathrm{E}}_{V,\tau^*}[L(h_{|V}, \tau^*)] - \min_{\tilde{h}} \mathop{\mathrm{E}}_{V,\tau^*}[L(\tilde{h}_{|V}, \tau^*)].$$

- All previous regret results hold if for $V_1, V_2 \supseteq \{u, v\}$,

$$\mathop{\mathrm{E}}_{\tau^*|V_1}[\tau^*(u, v)] = \mathop{\mathrm{E}}_{\tau^*|V_2}[\tau^*(u, v)]$$

for all u, v (pairwise independence on irrelevant alternatives).

References

- Agarwal, S., Graepel, T., Herbrich, R., Har-Peled, S., & Roth, D. (2005). Generalization bounds for the area under the roc curve. *JMLR* 6, 393–425.
- Agarwal, S., and Niyogi, P. (2005). Stability and generalization of bipartite ranking algorithms. COLT (pp. 32–47).
- Nir Ailon and Mehryar Mohri. An efficient reduction of ranking to classification. In Proceedings of *COLT* 2008. Helsinki, Finland, July 2008. Omnipress.
- Balcan, M.-F., Bansal, N., Beygelzimer, A., Coppersmith, D., Langford, J., and Sorkin, G. B. (2007). Robust reductions from ranking to classification. In Proceedings of *COLT* (pp. 604–619). Springer.
- Stephen Boyd, Corinna Cortes, Mehryar Mohri, and Ana Radovanovic (2012). Accuracy at the top. In *NIPS 2012*.
- Cohen, W. W., Schapire, R. E., and Singer, Y. (1999). Learning to order things. J. Artif. Intell. Res. (JAIR), 10, 243–270.
- Cossock, D., and Zhang, T. (2006). Subset ranking using regression. COLT (pp. 605–619).

References

- Corinna Cortes and Mehryar Mohri. AUC Optimization vs. Error Rate Minimization. In *Advances in Neural Information Processing Systems (NIPS 2003)*, 2004. MIT Press.

- Cortes, C., Mohri, M., and Rastogi, A. (2007a). An Alternative Ranking Problem for Search Engines. Proceedings of *WEA* 2007 (pp. 1–21). Rome, Italy: Springer.

- Corinna Cortes and Mehryar Mohri. Confidence Intervals for the Area under the ROC Curve. In *Advances in Neural Information Processing Systems (NIPS 2004)*, 2005. MIT Press.

- Crammer, K., and Singer, Y. (2001). Pranking with ranking. Proceedings of NIPS 2001, December 3-8, 2001, Vancouver, British Columbia, Canada] (pp. 641–647). MIT Press.

- Cynthia Dwork, Ravi Kumar, Moni Naor, and D. Sivakumar. Rank Aggregation Methods for the Web. *WWW 10*, 2001. ACM Press.

- J. P. Egan. *Signal Detection Theory and ROC Analysis*. Academic Press, 1975.

- Yoav Freund, Raj Iyer, Robert E. Schapire and Yoram Singer. An efficient boosting algorithm for combining preferences. *JMLR* 4:933-969, 2003.

References

- J. A. Hanley and B. J. McNeil. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, 1982.

- Ralf Herbrich, Thore Graepel, and Klaus Obermayer. Large margin rank boundaries for ordinal regression. Advances in Large Margin Classifiers, pages 115–132, 2000.

- Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. The MIT Press. 2012.

- Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The PageRank citation ranking: Bringing order to the Web, *Stanford Digital Library Technologies Project*, 1998.

- Lehmann, E. L. (1975). Nonparametrics: Statistical methods based on ranks. San Francisco, California: Holden-Day.

- Cynthia Rudin, Corinna Cortes, Mehryar Mohri, and Robert E. Schapire. Margin-Based Ranking Meets Boosting in the Middle. In *Proceedings of The 18th Annual Conference on Computational Learning Theory (COLT 2005)*, pages 63-78, 2005.

- Thorsten Joachims. Optimizing search engines using clickthrough data. *Proceedings of the 8th ACM SIGKDD*, pages 133-142, 2002.

Foundations of Machine Learning
Multi-Class Classification

Mehryar Mohri
Courant Institute and Google Research
mohri@cims.nyu.edu

Motivation

- Real-world problems often have multiple classes: text, speech, image, biological sequences.
- Algorithms studied so far: designed for binary classification problems.
- How do we design multi-class classification algorithms?
  - can the algorithms used for binary classification be generalized to multi-class classification?
  - can we reduce multi-class classification to binary classification?

Multi-Class Classification Problem

- Training data: sample drawn i.i.d. from set X according to some distribution D,

$$S=((x_1,y_1),\ldots,(x_m,y_m))\in X\times Y,$$

  - mono-label case: $\mathrm{Card}(Y)=k$.
  - multi-label case: $Y=\{-1,+1\}^k$.

- Problem: find classifier $h\colon X\to Y$ in H with small generalization error,
  - mono-label case: $R(h)=\mathrm{E}_{x\sim D}[1_{h(x)\neq f(x)}]$.
  - multi-label case: $R(h)=\mathrm{E}_{x\sim D}\left[\frac{1}{k}\sum_{l=1}^{k}1_{[h(x)]_l\neq[f(x)]_l}\right]$.

Notes

- In most tasks considered, number of classes $k \leq 100$.
- For k large, problem often not treated as a multi-class classification problem (ranking or density estimation, e.g., automatic speech recognition).
- Computational efficiency issues arise for larger ks.
- In general, classes not balanced.

Multi-Class Classification - Margin

- Hypothesis set H:
  - functions $h\colon X \times Y \to \mathbb{R}$.
  - label returned: $x \mapsto \operatorname*{argmax}_{y \in Y} h(x, y)$.
- Margin:
  - $\rho_h(x, y) = h(x, y) - \max_{y' \neq y} h(x, y')$.
  - error: $1_{\rho_h(x,y) \leq 0} \leq \Phi_\rho(\rho_h(x, y))$.
  - empirical margin loss:

$$\widehat{R}_\rho(h) = \frac{1}{m} \sum_{i=1}^{m} \Phi_\rho(\rho_h(x_i, y_i)).$$

Multi-Class Margin Bound

(MM et al. 2012; Kuznetsov, MM, and Syed, 2014)

- Theorem: let $H \subseteq \mathbb{R}^{X \times Y}$ with $Y = \{1, \ldots, k\}$. Fix $\rho > 0$. Then, for any $\delta > 0$, with probability at least $1 - \delta$, the following multi-class classification bound holds for all $h \in H$:

$$R(h) \leq \widehat{R}_\rho(h) + \frac{4k}{\rho} \mathfrak{R}_m(\Pi_1(H)) + \sqrt{\frac{\log \frac{1}{\delta}}{2m}},$$

with $\Pi_1(H) = \{x \mapsto h(x, y) \colon y \in Y, h \in H\}$.

Kernel-Based Hypotheses

- Hypothesis set $H_{K,p}$:
  - $\mathbf{\Phi}$ feature mapping associated to PDS kernel K.
  - functions $(x, y) \mapsto \mathbf{w}_y \cdot \mathbf{\Phi}(x)$, $y \in \{1, \ldots, k\}$.
  - label returned: $x \mapsto \underset{y \in \{1,\ldots,k\}}{\operatorname{argmax}} \; \mathbf{w}_y \cdot \mathbf{\Phi}(x)$.
  - for any $p \geq 1$,

$$H_{K,p} = \{(x, y) \in X \times [1, k] \mapsto \mathbf{w}_y \cdot \mathbf{\Phi}(x) \colon \mathbf{W} = (\mathbf{w}_1, \ldots, \mathbf{w}_k)^\top, \|\mathbf{W}\|_{\mathbb{H},p} \leq \Lambda\}.$$

$$\mathfrak{R}_m(\Pi_1(\mathcal{H}_{K,p})) \leq \sqrt{\frac{r^2 \Lambda^2}{m}}.$$

Multi-Class Margin Bound - Kernels

(MM et al. 2012)

- Theorem: let $K\colon X \times X \to \mathbb{R}$ be a PDS kernel and let $\Phi\colon X \to \mathbb{H}$ be a feature mapping associated to K. Fix $\rho > 0$. Then, for any $\delta > 0$, with probability at least $1 - \delta$, the following multiclass bound holds for all $h \in H_{K,p}$:

$$R(h) \leq \widehat{R}_\rho(h) + 4k\sqrt{\frac{r^2\Lambda^2}{\rho^2 m}} + \sqrt{\frac{\log \frac{1}{\delta}}{2m}},$$

where $r^2 = \sup_{x \in X} K(x, x)$.

Approaches

- Single classifier:
  - Multi-class SVMs.
  - AdaBoost.MH.
  - Conditional Maxent.
  - Decision trees.
- Combination of binary classifiers:
  - One-vs-all.
  - One-vs-one.
  - Error-correcting codes.

Multi-Class SVMs

(Weston and Watkins, 1999; Crammer and Singer, 2001)

- Optimization problem:

$$\min_{\mathbf{w},\boldsymbol{\xi}} \frac{1}{2}\sum_{l=1}^{k} \|\mathbf{w}_l\|^2 + C\sum_{i=1}^{m} \xi_i$$

$$\text{subject to: } \mathbf{w}_{y_i}\cdot\mathbf{x}_i + \delta_{y_i,l} \geq \mathbf{w}_l\cdot\mathbf{x}_i + 1 - \xi_i$$

$$\xi_i \geq 0, (i,l)\in[1,m]\times Y.$$

- Decision function:

$$h\colon x \mapsto \operatorname*{argmax}_{l\in Y}\,(\mathbf{w}_l\cdot\mathbf{x}).$$

Notes

- Directly based on generalization bounds.
- Comparison with (Weston and Watkins, 1999): single slack variable per point, maximum of slack variables (penalty for worst class):
$$\sum_{l=1}^{k} \xi_{il} \rightarrow \max_{l=1}^{k} \xi_{il}.$$
- PDS kernel instead of inner product
- Optimization: complex constraints, mk-size problem.
  - specific solution based on decomposition into m disjoint sets of constraints (Crammer and Singer, 2001).

Dual Formulation

- Optimization problem: $\boldsymbol{\alpha}_i$ ith row of matrix $\boldsymbol{\alpha} \in \mathbb{R}^{m \times k}$.

$$\max_{\boldsymbol{\alpha}=[\alpha_{ij}]} \sum_{i=1}^{m} \boldsymbol{\alpha}_i \cdot \mathbf{e}_{y_i} - \frac{1}{2} \sum_{i=1}^{m} (\boldsymbol{\alpha}_i \cdot \boldsymbol{\alpha}_j)(\mathbf{x}_i \cdot \mathbf{x}_j)$$

$$\text{subject to: } \forall i \in [1, m], (0 \leq \alpha_{iy_i} \leq C) \wedge (\forall j \neq y_i, \alpha_{ij} \leq 0) \wedge (\boldsymbol{\alpha}_i \cdot \mathbf{1} = 0).$$

- Decision function:

$$h(x) = \operatorname*{argmax}_{l \in [1, k]} \Big(\sum_{i=1}^{m} \alpha_{il} (\mathbf{x}_i \cdot \mathbf{x}) \Big).$$

AdaBoost

(Schapire and Singer, 2000)

- Training data (multi-label case):

$$(x_1, y_1), \ldots, (x_m, y_m) \in X \times \{-1, 1\}^k.$$

- Reduction to binary classification:
  - each example leads to k binary examples:

$$(x_i, y_i) \rightarrow ((x_i, 1), y_i[1]), \ldots, ((x_i, k), y_i[k]), i \in [1, m].$$

  - apply AdaBoost to the resulting problem.
  - choice of α_t.

- Computational cost: mk distribution updates at each round.

AdaBoost.MH

$H \subseteq (\{-1,+1\}^k)^{(X \times Y)}$.

$\textsc{AdaBoost.MH}(S=((x_1,y_1),\ldots,(x_m,y_m)))$

for $i \leftarrow 1$ **to** m **do**
    for $l \leftarrow 1$ **to** k **do**
        $D_1(i,l) \leftarrow \frac{1}{mk}$
for $t \leftarrow 1$ **to** T **do**
    $h_t \leftarrow$ base classifier in H with small error $\epsilon_t = \Pr_{D_t}[h_t(x_i,l) \neq y_i[l]]$
    $\alpha_t \leftarrow$ choose $\quad \triangleright$ to minimize Z_t
    $Z_t \leftarrow \sum_{i,l} D_t(i,l) \exp(-\alpha_t y_i[l] h_t(x_i,l))$
    for $i \leftarrow 1$ **to** m **do**
        for $l \leftarrow 1$ **to** k **do**
            $D_{t+1}(i,l) \leftarrow \frac{D_t(i,l)\exp(-\alpha_t y_i[l] h_t(x_i,l))}{Z_t}$
$f_T \leftarrow \sum_{t=1}^{T} \alpha_t h_t$
return $h_T = \mathrm{sgn}(f_T)$

Bound on Empirical Error

- **Theorem**: The empirical error of the classifier output by AdaBoost.MH verifies:

$$\widehat{R}(h) \leq \prod_{t=1}^{T} Z_t.$$

- **Proof**: similar to the proof for AdaBoost.
- **Choice of α_t**:
  - for $H \subseteq (\{-1, +1\}^k)^{X \times Y}$, as for AdaBoost, $\alpha_t = \frac{1}{2} \log \frac{1-\epsilon_t}{\epsilon_t}$.
  - for $H \subseteq ([-1, 1]^k)^{X \times Y}$, same choice: minimize upper bound.
  - other cases: numerical/approximation method.

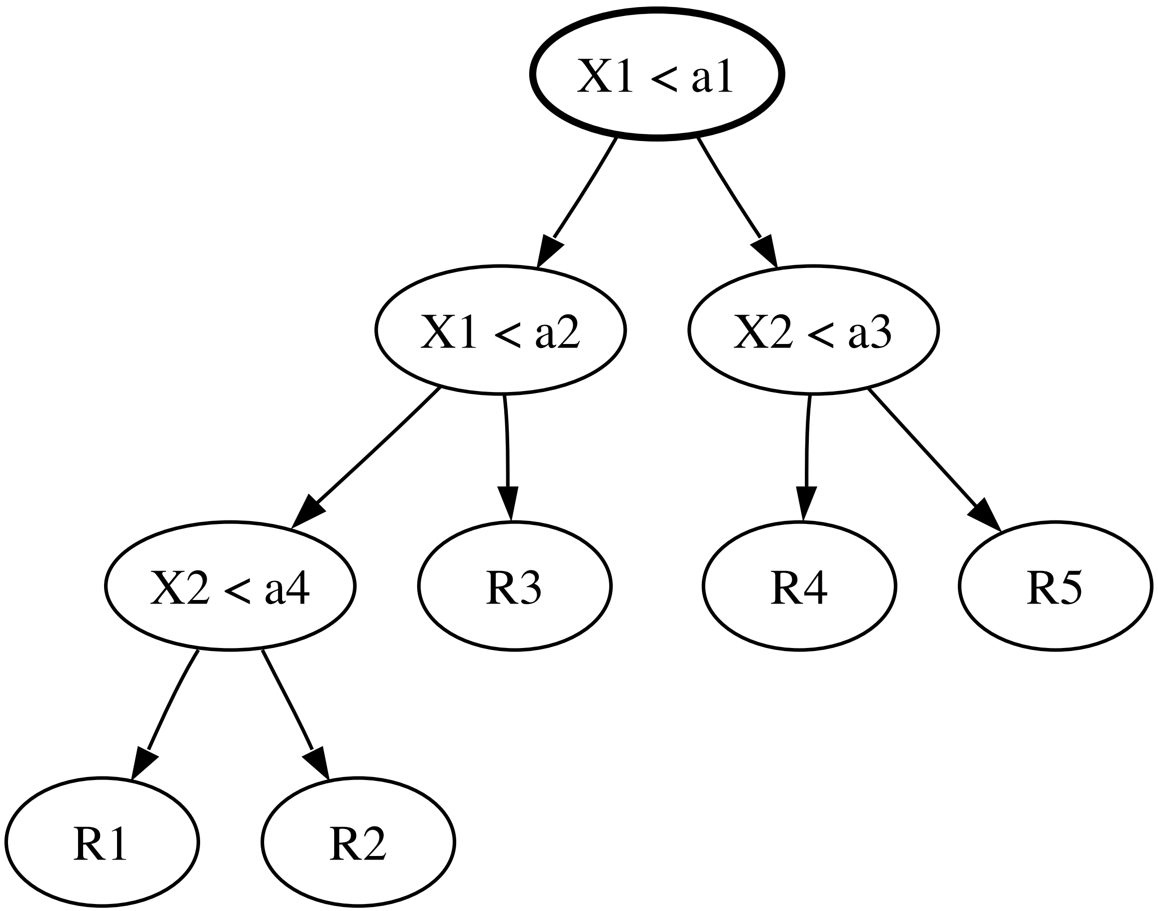
Notes

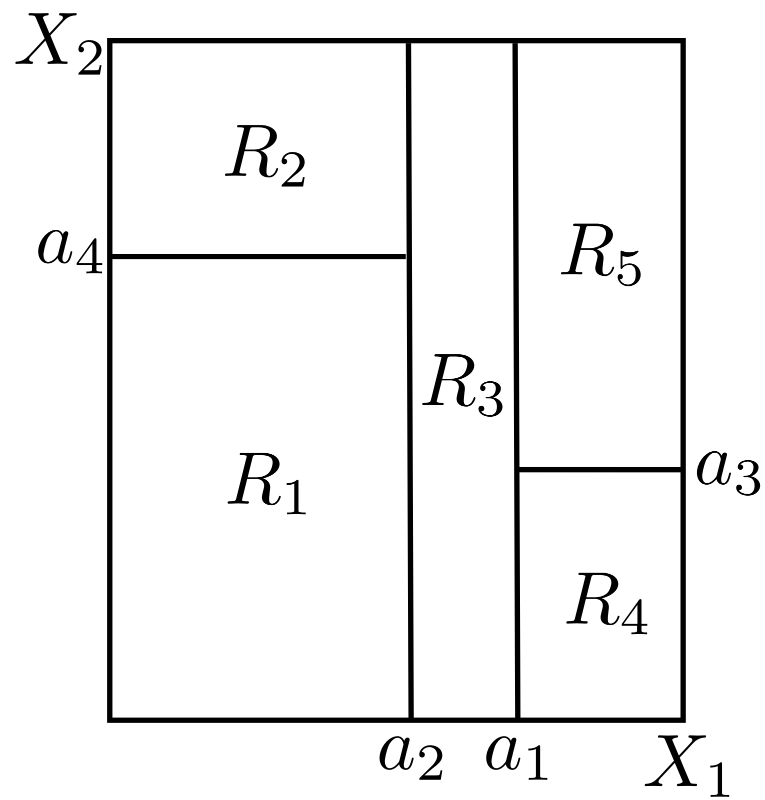
- Objective function:

$$F(\boldsymbol{\alpha}) = \sum_{i=1}^{m} \sum_{l=1}^{k} e^{-y_i[l] f_n(x_i, l)} = \sum_{i=1}^{m} \sum_{l=1}^{k} e^{-y_i[l] \sum_{t=1}^{n} \alpha_t h_t(x_i, l)}.$$

- All comments and analysis given for AdaBoost apply here.
- Alternative: Adaboost.MR, which coincides with a special case of RankBoost (ranking lecture).

Decision Trees





Different Types of Questions

- Decision trees
  - $X \in \{\text{blue}, \text{white}, \text{red}\}$: categorical questions.
  - $X \leq a$: continuous variables.
- Binary space partition (BSP) trees:
  - $\sum_{i=1}^{n} \alpha_i X_i \leq a$: partitioning with convex polyhedral regions.
- Sphere trees:
  - $||X - a_0|| \leq a$: partitioning with pieces of spheres.

Hypotheses

- In each region R_t,
  - classification: majority vote - ties broken arbitrarily,

$$\widehat{y}_t = \operatorname*{argmax}_{y \in Y} \left|\{x_i \in R_t \colon i \in [1, m], y_i = y\}\right|.$$

  - regression: average value,

$$\widehat{y}_t = \frac{1}{|S \cap R_t|} \sum_{\substack{x_i \in R_t \\ i \in [1,m]}} y_i \,.$$

- Form of hypotheses:

$$h \colon x \mapsto \sum_t \widehat{y}_t 1_{x \in R_t} \,.$$

Training

- **Problem**: general problem of determining partition with minimum empirical error is NP-hard.
- **Heuristics**: greedy algorithm.
    - for all $j \in [1, N]$, $\theta \in \mathbb{R}$, $R^+(j,\theta) = \{x_i \in R \colon x_i[j] \geq \theta, i \in [1, m]\}$
      $R^-(j,\theta) = \{x_i \in R \colon x_i[j] < \theta, i \in [1, m]\}$.

$\textsc{Decision-Trees}(S = ((x_1, y_1), \ldots, (x_m, y_m)))$

$P \leftarrow \{S\}$ $\triangleright$ initial partition
for each region $R \in P$ such that $\mathrm{Pred}(R)$ **do**
  $(j, \theta) \leftarrow \mathrm{argmin}_{(j,\theta)}\ \mathrm{error}(R^-(j,\theta)) + \mathrm{error}(R^+(j,\theta))$
  $P \leftarrow P - R \cup \{R^-(j,\theta), R^+(j,\theta)\}$
return P

Splitting/Stopping Criteria

- Problem: larger trees overfit training sample.
- Conservative splitting:
  - split node only if loss reduced by some fixed value $\eta > 0$.
  - issue: seemingly bad split dominating useful splits.
- Grow-then-prune technique (CART):
  - grow very large tree, $\mathrm{Pred}(R)$: $|R| > |n_0|$.
  - prune tree based on: $F(T) = \widehat{\mathrm{Loss}}(T) + \alpha |T|$, $\alpha \geq 0$ parameter determined by cross-validation.

Decision Tree Tools

- Most commonly used tools for learning decision trees:
  - CART (classification and regression tree) (Breiman et al., 1984).
  - C4.5 (Quinlan, 1986, 1993) and C5.0 (RuleQuest Research) a commercial system.
- Differences: minor between latest versions.

Approaches

- Single classifier:
  - SVM-type algorithm.
  - AdaBoost-type algorithm.
  - Conditional Maxent.
  - Decision trees.
- Combination of binary classifiers:
  - One-vs-all.
  - One-vs-one.
  - Error-correcting codes.

One-vs-All

- Technique:
  - for each class $l \in Y$ learn binary classifier $h_l = \operatorname{sgn}(f_l)$.
  - combine binary classifiers via voting mechanism, typically majority vote: $h \colon x \mapsto \operatorname*{argmax}_{l \in Y} f_l(x)$.
- Problem: poor justification (in general).
  - calibration: classifier scores not comparable.
  - nevertheless: simple and frequently used in practice, computational advantages in some cases.

One-vs-One

- Technique:
  - for each pair $(l, l') \in Y, l \neq l'$ learn binary classifier $h_{ll'} : X \to \{0, 1\}$.
  - combine binary classifiers via majority vote:
$$h(x) = \underset{l' \in Y}{\operatorname{argmax}} \left|\{l : h_{ll'}(x) = 1\}\right|.$$
- Problem:
  - computational: train $k(k-1)/2$ binary classifiers.
  - overfitting: size of training sample could become small for a given pair.

Computational Comparison

	Training	Testing
One-vs-all	$O(kB_{\text{train}}(m))$ $O(km^\alpha)$	$O(kB_{\text{test}})$
One-vs-one	$O(k^2 B_{\text{train}}(m/k))$ (on average) $O(k^{2-\alpha} m^\alpha)$	$O(k^2 B_{\text{test}})$ *smaller* N_{SV} *per* B

Time complexity for SVMs, α less than 3.

Error-Correcting Code Approach

(Dietterich and Bakiri, 1995)

- Idea:
  - assign F-long binary code word to each class:

    $$\longrightarrow \mathbf{M} = [\mathbf{M}_{lj}] \in \{0, 1\}^{[1,k]\times[1,F]}.$$

  - learn binary classifier $f_j \colon X \to \{0, 1\}$ for each column. Example x in class l labeled with M_{lj}.
  - classifier output: $\Big(\mathbf{f}(x) = \big(f_1(x), \ldots, f_F(x)\big)\Big)$,

$$h \colon x \mapsto \operatorname*{argmin}_{l \in Y} d_{\text{Hamming}}\Big(\mathbf{M}_l, \mathbf{f}(x)\Big).$$

Illustration

- 8 classes, code-length: 6.

codes

classes

	1	2	3	4	5	6
1	0	0	0	1	0	0
2	1	0	0	0	0	0
3	0	1	1	0	1	0
4	1	1	0	0	0	0
5	1	1	0	0	1	0
6	0	0	1	1	0	1
7	0	0	1	0	0	0
8	0	1	0	1	0	0

$f_1(x)$	$f_2(x)$	$f_3(x)$	$f_4(x)$	$f_5(x)$	$f_6(x)$
0	1	1	0	1	1

new example x

Error-Correcting Codes - Design

- Main ideas:
  - independent columns: otherwise no effective discrimination.
  - distance between rows: if the minimal Hamming distance between rows is d, then the multi-class can correct $\left\lfloor \frac{d-1}{2} \right\rfloor$ (classification) errors.
  - columns may correspond to features selected for the task.
  - one-vs-all and one-vs-one (with ternary codes) are special cases.

Extensions

(Allwein et al., 2000)

- Matrix entries in $\{-1, 0, +1\}$:
  - examples marked with 0 disregarded during training.
  - $\longrightarrow$ one-vs-one becomes also a special case.
- Margin loss L: function of $yf(x)$, e.g., hinge loss.
  - Hamming loss:

$$h(x) = \operatorname*{argmin}_{l \in \{1,\ldots,k\}} \sum_{j=1}^{F} \frac{1 - \operatorname{sgn}\big(\mathbf{M}_{lj} f_j(x)\big)}{2}.$$

  - Margin loss:

$$h(x) = \operatorname*{argmin}_{l \in \{1,\ldots,k\}} \sum_{j=1}^{F} L\big(\mathbf{M}_{lj} f_j(x)\big).$$

Applications

- One-vs-all approach is the most widely used combination method.
- No clear empirical evidence of the superiority of other approaches (Rifkin and Klautau, 2004).
  - except perhaps on small data sets with relatively large error rate.
- Large structured multi-class problems: often treated as ranking problems (see ranking lecture).

References

- Erin L. Allwein, Robert E. Schapire and Yoram Singer. Reducing multiclass to binary: A unifying approach for margin classifiers. *Journal of Machine Learning Research*, 1:113-141, 2000.

- K. Crammer and Y. Singer. Improved output coding for classification using continuous relaxation. In Proceedings of *NIPS*, 2000.

- Koby Crammer and Yoram Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, 2:265–292, 2001.

- Koby Crammer and Yoram Singer. On the Learnability and Design of Output Codes for Multiclass Problems. *Machine Learning 47*, 2002.

- Thomas G. Dietterich, Ghulum Bakiri: Solving Multiclass Learning Problems via Error-Correcting Output Codes. *Journal of Artificial Intelligence Research (JAIR) 2*: 263-286, 1995.

- Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. Foundations of Machine Learning, the MIT Press, 2012.

- John C. Platt, Nello Cristianini, and John Shawe-Taylor. Large Margin DAGS for Multiclass Classification. In *Advances in Neural Information Processing Systems 12 (NIPS 1999)*, pp. 547-553, 2000.

References

- Ryan Rifkin. "Everything Old Is New Again: A Fresh Look at Historical Approaches in Machine Learning." Ph.D. Thesis, MIT, 2002.

- Rifkin and Klautau. "In Defense of One-Vs-All Classification." *Journal of Machine Learning Research*, 5:101-141, 2004.

- Robert E. Schapire. The boosting approach to machine learning: An overview. In D. D. Denison, M. H. Hansen, C. Holmes, B. Mallick, B. Yu, editors, *Nonlinear Estimation and Classification*. Springer, 2003.

- Robert E. Schapire, Yoav Freund, Peter Bartlett and Wee Sun Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. *The Annals of Statistics*, 26(5):1651-1686, 1998.

- Robert E. Schapire and Yoram Singer. BoosTexter: A boosting-based system for text categorization. *Machine Learning*, 39(2/3):135-168, 2000.

- Jason Weston and Chris Watkins. Support Vector Machines for Multi-Class Pattern Recognition. *Proceedings of the Seventh European Symposium On Artificial Neural Networks (ESANN '99)*, 1999.

Foundations of Machine Learning
Regression

Mehryar Mohri
Courant Institute and Google Research
mohri@cims.nyu.edu

Regression Problem

- Training data: sample drawn i.i.d. from set X according to some distribution D,

$$S=((x_1,y_1),\ldots,(x_m,y_m))\in X\times Y,$$

  with $Y\subseteq\mathbb{R}$ is a measurable subset.

- Loss function: $L\colon Y\times Y\to\mathbb{R}_+$ a measure of closeness, typically $L(y,y')=(y'-y)^2$ or $L(y,y')=|y'-y|^p$ for some $p\geq 1$.

- Problem: find hypothesis $h\colon X\to\mathbb{R}$ in H with small generalization error with respect to target f

$$R_D(h)=\mathop{\mathrm{E}}_{x\sim D}\Big[L\big(h(x),f(x)\big)\Big].$$

Notes

- Empirical error:

$$\widehat{R}_D(h) = \frac{1}{m}\sum_{i=1}^{m} L\big(h(x_i), y_i\big).$$

- In much of what follows:
  - $Y = \mathbb{R}$ or $Y = [-M, M]$ for some $M > 0$.
  - $L(y, y') = (y' - y)^2 \longrightarrow$ mean squared error.

This Lecture

- Generalization bounds
- Linear regression
- Kernel ridge regression
- Support vector regression
- Lasso

Generalization Bound - Finite H

- Theorem: let H be a finite hypothesis set, and assume that L is bounded by M. Then, for any $\delta > 0$, with probability at least $1-\delta$,

$$\forall h \in H, R(h) \leq \widehat{R}(h) + M\sqrt{\frac{\log|H| + \log\frac{2}{\delta}}{2m}}.$$

- Proof: By the union bound,

$$\Pr\left[\sup_{h\in H}\left|R(h)-\widehat{R}(h)\right|>\epsilon\right] \leq \sum_{h\in H}\Pr\left[\left|R(h)-\widehat{R}(h)\right|>\epsilon\right].$$

By Hoeffding's bound, for a fixed h,

$$\Pr\left[\left|R(h)-\widehat{R}(h)\right|>\epsilon\right] \leq 2e^{-\frac{2m\epsilon^2}{M^2}}.$$

Rademacher Complexity of Lp Loss

- Theorem: Let $p \geq 1$, $H_p = \{x \mapsto |h(x) - f(x)|^p \colon h \in H\}$. Assume that $\sup_{x \in X, h \in H} |h(x) - f(x)| \leq M$. Then, for any sample S of size m,

$$\widehat{\mathfrak{R}}_S(H_p) \leq p M^{p-1} \widehat{\mathfrak{R}}_S(H).$$

Proof

- Proof: Let $H' = \{x \mapsto h(x) - f(x) \colon h \in H\}$. Then, observe that $H_p = \{\phi \circ h \colon h \in H'\}$ with $\phi \colon x \mapsto |x|^p$.
  - ϕ is pM^{p-1} - Lipschitz over $[-M, M]$, thus
  $$\widehat{\mathfrak{R}}_S(H_p) \leq pM^{p-1} \widehat{\mathfrak{R}}_S(H').$$
  - Next, observe that:
  $$\begin{aligned}\widehat{\mathfrak{R}}_S(H') &= \frac{1}{m} \operatorname*{E}_{\boldsymbol{\sigma}} \Big[\sup_{h \in H} \sum_{i=1}^{m} \sigma_i h(x_i) + \sigma_i f(x_i) \Big] \\ &= \frac{1}{m} \operatorname*{E}_{\boldsymbol{\sigma}} \Big[\sup_{h \in H} \sum_{i=1}^{m} \sigma_i h(x_i) \Big] + \operatorname*{E}_{\boldsymbol{\sigma}} \Big[\sum_{i=1}^{m} \sigma_i f(x_i) \Big] = \widehat{\mathfrak{R}}_S(H).\end{aligned}$$

Rad. Complexity Regression Bound

- **Theorem**: Let $p \geq 1$ and assume that $\|h - f\|_\infty \leq M$ for all $h \in H$. Then, for any $\delta > 0$, with probability at least $1-\delta$, for all $h \in H$,

$$\mathrm{E}\Big[\big|h(x) - f(x)\big|^p\Big] \leq \frac{1}{m}\sum_{i=1}^{m}\big|h(x_i) - f(x_i)\big|^p + 2pM^{p-1}\mathfrak{R}_m(H) + M^p\sqrt{\frac{\log\frac{1}{\delta}}{2m}}.$$

$$\mathrm{E}\Big[\big|h(x) - f(x)\big|^p\Big] \leq \frac{1}{m}\sum_{i=1}^{m}\big|h(x_i) - f(x_i)\big|^p + 2pM^{p-1}\widehat{\mathfrak{R}}_S(H) + 3M^p\sqrt{\frac{\log\frac{2}{\delta}}{2m}}.$$

- **Proof**: Follows directly bound on Rademacher complexity and general Rademacher bound.

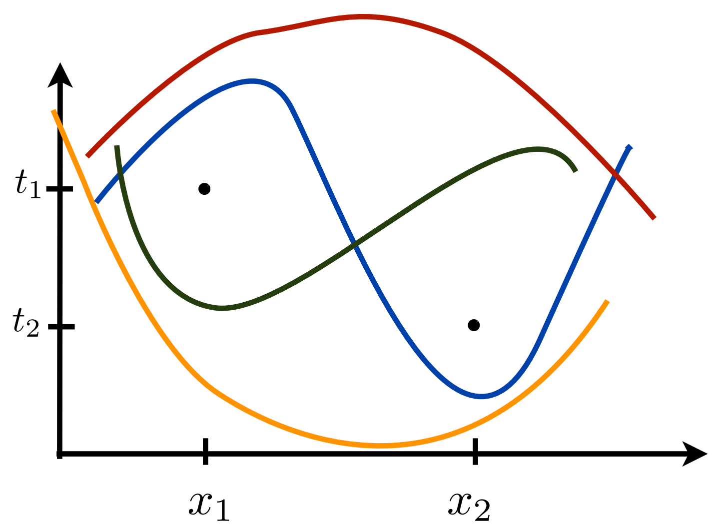
Notes

- As discussed for binary classification:
  - estimating the Rademacher complexity can be computationally hard for some Hs.
  - can we come up instead with a combinatorial measure that is easier to compute?

Shattering

- Definition: Let G be a family of functions mapping from X to $\mathbb{R}$. $A = \{x_1, \ldots, x_m\}$ is shattered by G if there exist $t_1, \ldots, t_m \in \mathbb{R}$ such that

$$\left| \left\{ \begin{bmatrix} \operatorname{sgn}\left(g(x_1) - t_1\right) \\ \vdots \\ \operatorname{sgn}\left(g(x_m) - t_m\right) \end{bmatrix} : g \in G \right\} \right| = 2^m.$$

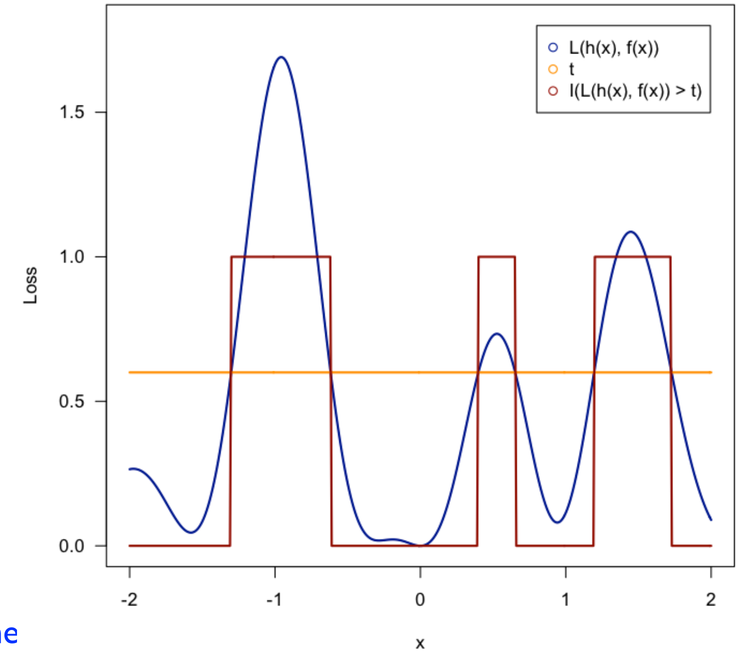


Pseudo-Dimension

(Pollard, 1984)

- **Definition**: Let G be a family of functions mapping from X to $\mathbb{R}$. The pseudo-dimension of G, $\mathrm{Pdim}(G)$, is the size of the largest set shattered by G.
- **Definition** (equivalent, see also (Vapnik, 1995)):

$$\mathrm{Pdim}(G) = \mathrm{VCdim}\Big(\big\{(x,t) \mapsto 1_{(g(x)-t)>0} \colon g \in G\big\}\Big).$$



Pseudo-Dimension - Properties

- Theorem: Pseudo-dimension of hyperplanes.

$$\mathrm{Pdim}(\mathbf{x} \mapsto \mathbf{w} \cdot \mathbf{x} + b \colon \mathbf{w} \in \mathbb{R}^N, b \in \mathbb{R}) = N + 1.$$

- Theorem: Pseudo-dimension of a vector space of real-valued functions H:

$$\mathrm{Pdim}(H) = \dim(H).$$

Generalization Bounds

Classification ⟶ Regression

- **Lemma** (Lebesgue integral): for $f \geq 0$ measurable,

$$\mathop{\mathrm{E}}_{D}[f(x)] = \int_0^\infty \Pr_{D}[f(x) > t]\,dt.$$

- Assume that the loss function L is bounded by M.

$$\begin{aligned}
|R(h) - \widehat{R}(h)| &= \left| \int_0^M \Big(\Pr_{x\sim D}[L(h(x), f(x)) > t] - \Pr_{x\sim S}[L(h(x), f(x)) > t] \Big)\, dt \right| \\
&\leq M \sup_{t\in[0,M]} \left| \Pr_{x\sim D}[L(h(x), f(x)) > t] - \Pr_{x\sim S}[L(h(x), f(x)) > t] \right| \\
&= M \sup_{t\in[0,M]} \left| \mathop{\mathrm{E}}_{x\sim D}[1_{L(h(x),f(x))>t}] - \mathop{\mathrm{E}}_{x\sim S}[1_{L(h(x),f(x))>t}] \right|.
\end{aligned}$$

$$\Pr\left[\sup_{h\in H} |R(h) - \widehat{R}(h)| > \epsilon\right] \boxed{\leq \Pr\left[\sup_{\substack{h\in H\\ t\in[0,M]}} \left| R(1_{L(h,f)>t}) - \widehat{R}(1_{L(h,f)>t}) \right| > \frac{\epsilon}{M}\right]}.$$

Standard classification generalization bound.

Generalization Bound - Pdim

- **Theorem**: Let H be a family of real-valued functions. Assume that $\mathrm{Pdim}(\{L(h,f)\colon h \in H\}) = d < \infty$ and that the loss L is bounded by M. Then, for any $\delta > 0$, with probability at least $1-\delta$, for any $h \in H$,

$$R(h) \leq \widehat{R}(h) + M\sqrt{\frac{2d \log \frac{em}{d}}{m}} + M\sqrt{\frac{\log \frac{1}{\delta}}{2m}}.$$

- **Proof**: follows observation of previous slide and VCDim bound for indicator functions of lecture 3.

Notes

- Pdim bounds in unbounded case modulo assumptions: existence of an envelope function or moment assumptions.
- Other relevant capacity measures:
  - covering numbers.
  - packing numbers.
  - fat-shattering dimension.

This Lecture

- Generalization bounds
- Linear regression
- Kernel ridge regression
- Support vector regression
- Lasso

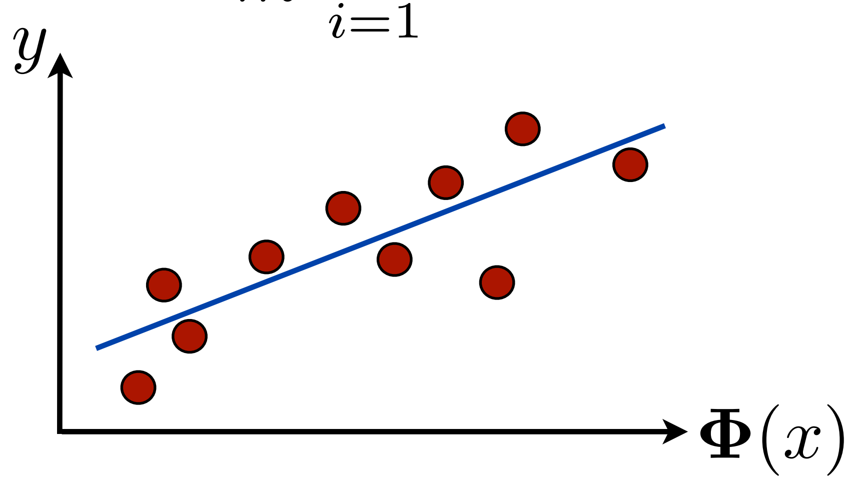
Linear Regression

- Feature mapping $\mathbf{\Phi}\colon X \to \mathbb{R}^N$.
- Hypothesis set: linear functions.

$$\{x \mapsto \mathbf{w} \cdot \mathbf{\Phi}(x) + b\colon \mathbf{w} \in \mathbb{R}^N, b \in \mathbb{R}\}.$$

- Optimization problem: empirical risk minimization.

$$\min_{\mathbf{w}, b} F(\mathbf{w}, b) = \frac{1}{m} \sum_{i=1}^{m} (\mathbf{w} \cdot \mathbf{\Phi}(x_i) + b - y_i)^2 .$$



Linear Regression - Solution

- Rewrite objective function as $F(\mathbf{W}) = \frac{1}{m}\|\mathbf{X}^\top \mathbf{W} - \mathbf{Y}\|^2$, $\mathbf{X} = \left[\begin{smallmatrix} \Phi(x_1) & \ldots & \Phi(x_m) \\ 1 & \ldots & 1 \end{smallmatrix}\right] \in \mathbb{R}^{(N+1)\times m}$

$$\text{with } \mathbf{X}^\top = \begin{bmatrix} \mathbf{\Phi}(x_1)^\top & 1 \\ \vdots & \\ \mathbf{\Phi}(x_m)^\top & 1 \end{bmatrix} \quad \mathbf{W} = \begin{bmatrix} w_1 \\ \vdots \\ w_N \\ b \end{bmatrix} \quad \mathbf{Y} = \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix}.$$

- Convex and differentiable function.

$$\nabla F(\mathbf{W}) = \frac{2}{m}\mathbf{X}(\mathbf{X}^\top \mathbf{W} - \mathbf{Y}).$$

$$\nabla F(\mathbf{W}) = 0 \Leftrightarrow \mathbf{X}(\mathbf{X}^\top \mathbf{W} - \mathbf{Y}) = 0 \Leftrightarrow \mathbf{X}\mathbf{X}^\top \mathbf{W} = \mathbf{X}\mathbf{Y}.$$

Linear Regression - Solution

- Solution:

$$\mathbf{W} = \begin{cases} (\mathbf{X}\mathbf{X}^\top)^{-1}\mathbf{X}\mathbf{Y} & \text{if } \mathbf{X}\mathbf{X}^\top \text{ invertible.} \\ (\mathbf{X}\mathbf{X}^\top)^{\dagger}\mathbf{X}\mathbf{Y} & \text{in general.} \end{cases}$$

  - Computational complexity: $O(mN + N^3)$ if matrix inversion in $O(N^3)$.
  - Poor guarantees in general, no regularization.
  - For output labels in $\mathbb{R}^p$, $p > 1$, solve p distinct linear regression problems.

This Lecture

- Generalization bounds
- Linear regression
- Kernel ridge regression
- Support vector regression
- Lasso

Mean Square Bound - Kernel-Based Hypotheses

- **Theorem:** Let $K\colon X \times X \to \mathbb{R}$ be a PDS kernel and let $\Phi\colon X \to \mathbb{H}$ be a feature mapping associated to K. Let $H = \left\{\mathbf{x} \mapsto \mathbf{w} \cdot \Phi(x) \colon \|\mathbf{w}\|_{\mathbb{H}} \leq \Lambda\right\}$. Assume $K(x, x) \leq R^2$ and $|f(x)| \leq \Lambda R$ for all $x \in X$. Then, for any $\delta > 0$, with probability at least $1 - \delta$, for any $h \in H$,

$$R(h) \leq \widehat{R}(h) + \frac{8R^2\Lambda^2}{\sqrt{m}}\left(1 + \frac{1}{2}\sqrt{\frac{\log\frac{1}{\delta}}{2}}\right)$$

$$R(h) \leq \widehat{R}(h) + \frac{8R^2\Lambda^2}{\sqrt{m}}\left(\sqrt{\frac{\mathrm{Tr}[\mathbf{K}]}{mR^2}} + \frac{3}{4}\sqrt{\frac{\log\frac{2}{\delta}}{2}}\right).$$

Mean Square Bound - Kernel-Based Hypotheses

- Proof: direct application of the Rademacher Complexity Regression Bound (this lecture) and bound on the Rademacher complexity of kernel-based hypotheses (lecture 5):

$$\widehat{\mathfrak{R}}_S(H) \leq \frac{\Lambda\sqrt{\mathrm{Tr}[\mathbf{K}]}}{m} \leq \sqrt{\frac{R^2\Lambda^2}{m}}.$$

Ridge Regression

(Hoerl and Kennard, 1970)

- Optimization problem:

$$\min_{\mathbf{w}} F(\mathbf{w}, b) = \lambda \|\mathbf{w}\|^2 + \sum_{i=1}^{m} \left(\mathbf{w} \cdot \mathbf{\Phi}(x_i) + b - y_i\right)^2,$$

where $\lambda \geq 0$ is a (regularization) parameter.

  - directly based on generalization bound.
  - generalization of linear regression.
  - closed-form solution.
  - can be used with kernels.

Ridge Regression - Solution

- Assume $b=0$: often constant feature used (but not equivalent to the use of original offset!).
- Rewrite objective function as

$$F(\mathbf{W}) = \lambda\|\mathbf{W}\|^2 + \|\mathbf{X}^\top\mathbf{W} - \mathbf{Y}\|^2.$$

- Convex and diferentiable function.

$$\nabla F(\mathbf{W}) = 2\lambda\mathbf{W} + 2\mathbf{X}(\mathbf{X}^\top\mathbf{W} - \mathbf{Y}).$$

$$\nabla F(\mathbf{W}) = 0 \Leftrightarrow (\mathbf{X}\mathbf{X}^\top + \lambda\mathbf{I})\mathbf{W} = \mathbf{X}\mathbf{Y}.$$

- Solution: $$\boxed{\mathbf{W} = \underbrace{(\mathbf{X}\mathbf{X}^\top + \lambda\mathbf{I})}_{\text{always invertible.}}{}^{-1}\mathbf{X}\mathbf{Y}}.$$

Ridge Regression - Equivalent Formulations

- Optimization problem:

$$\min_{\mathbf{w}, b} \sum_{i=1}^{m} (\mathbf{w} \cdot \mathbf{\Phi}(x_i) + b - y_i)^2$$

$$\text{subject to: } \|\mathbf{w}\|^2 \leq \Lambda^2.$$

- Optimization problem:

$$\min_{\mathbf{w}, b} \sum_{i=1}^{m} \xi_i^2$$

$$\text{subject to: } \xi_i = \mathbf{w} \cdot \mathbf{\Phi}(x_i) + b - y_i$$

$$\|\mathbf{w}\|^2 \leq \Lambda^2.$$

Ridge Regression Equations

- **Lagrangian**: assume $b = 0$. For all $\xi, \mathbf{w}, \boldsymbol{\alpha}', \lambda \geq 0$,

$$L(\xi, \mathbf{w}, \boldsymbol{\alpha}', \lambda) = \sum_{i=1}^{m} \xi_i^2 + \sum_{i=1}^{m} \alpha_i'(y_i - \xi_i - \mathbf{w} \cdot \mathbf{\Phi}(x_i)) + \lambda(\|\mathbf{w}\|^2 - \Lambda^2).$$

- **KKT conditions**:

$$\nabla_{\mathbf{w}} L = -\sum_{i=1}^{m} \alpha_i' \mathbf{\Phi}(x_i) + 2\lambda \mathbf{w} = 0 \iff \boxed{\mathbf{w} = \frac{1}{2\lambda} \sum_{i=1}^{m} \alpha_i' \mathbf{\Phi}(x_i).}$$

$$\nabla_{\xi_i} L = 2\xi_i - \alpha_i' = 0 \iff \boxed{\xi_i = \alpha_i'/2.}$$

$$\boxed{\begin{array}{l} \forall i \in [1, m], \alpha_i'(y_i - \xi_i - \mathbf{w} \cdot \mathbf{\Phi}(x_i)) = 0 \\ \lambda(\|\mathbf{w}\|^2 - \Lambda^2) = 0. \end{array}}$$

Moving to The Dual

- Plugging in the expression of $\mathbf{w}$ and ξ_is gives

$$L = \sum_{i=1}^{m} \frac{{\alpha'_i}^2}{4} + \sum_{i=1}^{m} \alpha'_i y_i - \sum_{i=1}^{m} \frac{{\alpha'_i}^2}{2} - \frac{1}{2\lambda} \sum_{i,j=1}^{m} \alpha'_i \alpha'_j \mathbf{\Phi}(x_i)^\top \mathbf{\Phi}(x_j) + \lambda \Big(\frac{1}{4\lambda^2} \| \sum_{i=1}^{m} \alpha'_i \mathbf{\Phi}(x_i) \|^2 - \Lambda^2 \Big).$$

- Thus,

$$\begin{aligned} L &= -\frac{1}{4} \sum_{i=1}^{m} {\alpha'_i}^2 + \sum_{i=1}^{m} \alpha'_i y_i - \frac{1}{4\lambda} \sum_{i,j=1}^{m} \alpha'_i \alpha'_j \mathbf{\Phi}(x_i)^\top \mathbf{\Phi}(x_j) - \lambda \Lambda^2 \\ &= -\lambda \sum_{i=1}^{m} \alpha_i^2 + 2 \sum_{i=1}^{m} \alpha_i y_i - \sum_{i,j=1}^{m} \alpha_i \alpha_j \mathbf{\Phi}(x_i)^\top \mathbf{\Phi}(x_j) - \lambda \Lambda^2, \end{aligned}$$

with $\alpha'_i = 2\lambda\alpha_i$.

RR - Dual Optimization Problem

- Optimization problem:

$$\max_{\boldsymbol{\alpha} \in \mathbb{R}^m} -\lambda \boldsymbol{\alpha}^\top \boldsymbol{\alpha} + 2\boldsymbol{\alpha}^\top \mathbf{y} - \boldsymbol{\alpha}^\top (\mathbf{X}^\top \mathbf{X}) \boldsymbol{\alpha}$$

or $$\max_{\boldsymbol{\alpha} \in \mathbb{R}^m} -\boldsymbol{\alpha}^\top (\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}) \boldsymbol{\alpha} + 2\boldsymbol{\alpha}^\top \mathbf{y}.$$

- Solution:

$$h(x) = \sum_{i=1}^{m} \alpha_i \boldsymbol{\Phi}(\mathbf{x}_i) \cdot \boldsymbol{\Phi}(x),$$

with $\boldsymbol{\alpha} = (\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{y}.$

Direct Dual Solution

- Lemma: The following matrix identity always holds.

$$(\mathbf{X}\mathbf{X}^\top + \lambda\mathbf{I})^{-1}\mathbf{X} = \mathbf{X}(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})^{-1}.$$

- Proof: Observe that $(\mathbf{X}\mathbf{X}^\top + \lambda\mathbf{I})\mathbf{X} = \mathbf{X}(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})$. Left-multiplying by $(\mathbf{X}\mathbf{X}^\top + \lambda\mathbf{I})^{-1}$ and right-multiplying by $(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})^{-1}$ yields the statement.

- Dual solution: $\boldsymbol{\alpha}$ such that

$$\mathbf{W} = \sum_{i=1}^m \alpha_i K(x_i, \cdot) = \sum_{i=1}^m \alpha_i \mathbf{\Phi}(x_i) = \mathbf{X}\boxed{\boldsymbol{\alpha}}.$$

By lemma, $\mathbf{W} = (\mathbf{X}\mathbf{X}^\top + \lambda\mathbf{I})^{-1}\mathbf{X}\mathbf{Y} = \mathbf{X}\boxed{(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{Y}}$.

This gives $\boxed{\boldsymbol{\alpha} = (\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{Y}}$.

Computational Complexity

	Solution	Prediction
Primal	$O(mN^2 + N^3)$	$O(N)$
Dual	$O(\kappa m^2 + m^3)$	$O(\kappa m)$

Kernel Ridge Regression

(Saunders et al., 1998)

- Optimization problem:

$$\max_{\boldsymbol{\alpha} \in \mathbb{R}^m} -\lambda \boldsymbol{\alpha}^\top \boldsymbol{\alpha} + 2\boldsymbol{\alpha}^\top \mathbf{y} - \boldsymbol{\alpha}^\top \mathbf{K} \boldsymbol{\alpha}$$

or $$\max_{\boldsymbol{\alpha} \in \mathbb{R}^m} -\boldsymbol{\alpha}^\top (\mathbf{K} + \lambda \mathbf{I}) \boldsymbol{\alpha} + 2\boldsymbol{\alpha}^\top \mathbf{y}.$$

- Solution:

$$h(x) = \sum_{i=1}^{m} \alpha_i K(x_i, x),$$

with $$\boldsymbol{\alpha} = (\mathbf{K} + \lambda \mathbf{I})^{-1} \mathbf{y}.$$

Notes

- Advantages:
  - strong theoretical guarantees.
  - generalization to outputs in $\mathbb{R}^p$: single matrix inversion (Cortes et al., 2007).
  - use of kernels.
- Disadvantages:
  - solution not sparse.
  - training time for large matrices: low-rank approximations of kernel matrix, e.g., Nyström approx., partial Cholesky decomposition.

This Lecture

- Generalization bounds
- Linear regression
- Kernel ridge regression
- Support vector regression
- Lasso

Support Vector Regression

(Vapnik, 1995)

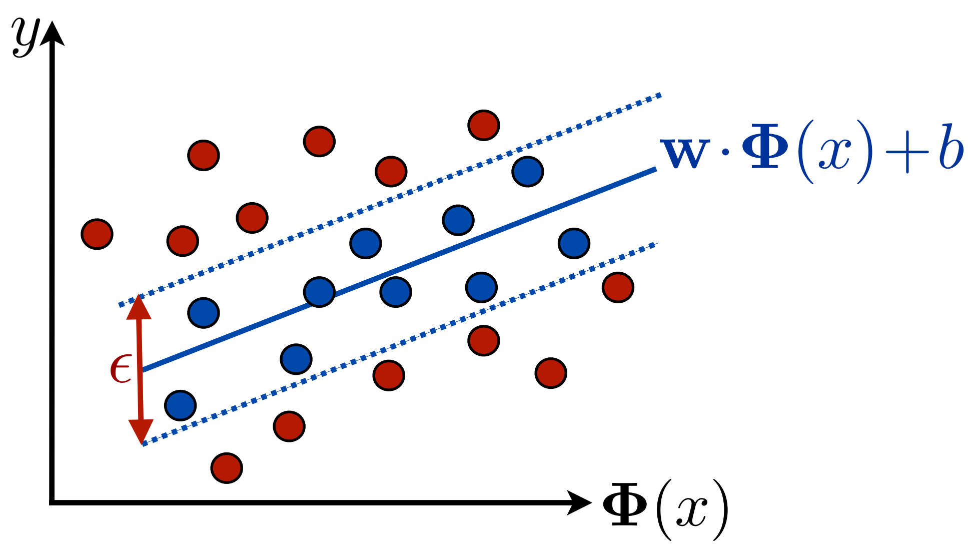
- Hypothesis set:

$$\{x \mapsto \mathbf{w} \cdot \mathbf{\Phi}(x) + b \colon \mathbf{w} \in \mathbb{R}^N, b \in \mathbb{R}\}.$$

- Loss function: ϵ-insensitive loss.

$$L(y, y') = |y' - y|_\epsilon = \max(0, |y' - y| - \epsilon).$$

Fit 'tube' with width ϵ to data.



Support Vector Regression (SVR)

(Vapnik, 1995)

- Optimization problem: similar to that of SVM.

$$\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m} \big|y_i - (\mathbf{w} \cdot \mathbf{\Phi}(x_i) + b)\big|_{\epsilon}.$$

- Equivalent formulation:

$$\min_{\mathbf{w},\boldsymbol{\xi},\boldsymbol{\xi}'} \ \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m}(\xi_i + \xi_i')$$

$$\begin{aligned}\text{subject to } & (\mathbf{w} \cdot \mathbf{\Phi}(x_i) + b) - y_i \leq \epsilon + \xi_i \\ & y_i - (\mathbf{w} \cdot \mathbf{\Phi}(x_i) + b) \leq \epsilon + \xi_i' \\ & \xi_i \geq 0, \xi_i' \geq 0.\end{aligned}$$

SVR - Dual Optimization Problem

- Optimization problem:

$$\max_{\boldsymbol{\alpha},\boldsymbol{\alpha}'} \; -\epsilon(\boldsymbol{\alpha}'+\boldsymbol{\alpha})^\top\mathbf{1}+(\boldsymbol{\alpha}'-\boldsymbol{\alpha})^\top\mathbf{y}-\frac{1}{2}(\boldsymbol{\alpha}'-\boldsymbol{\alpha})^\top\mathbf{K}(\boldsymbol{\alpha}'-\boldsymbol{\alpha})$$

$$\text{subject to: } (\mathbf{0}\leq\boldsymbol{\alpha}\leq\mathbf{C})\wedge(\mathbf{0}\leq\boldsymbol{\alpha}'\leq\mathbf{C})\wedge((\boldsymbol{\alpha}'-\boldsymbol{\alpha})^\top\mathbf{1}=0)\,.$$

- Solution:

$$h(x)=\sum_{i=1}^{m}(\alpha'_i-\alpha_i)K(\mathbf{x}_i,\mathbf{x})+b$$

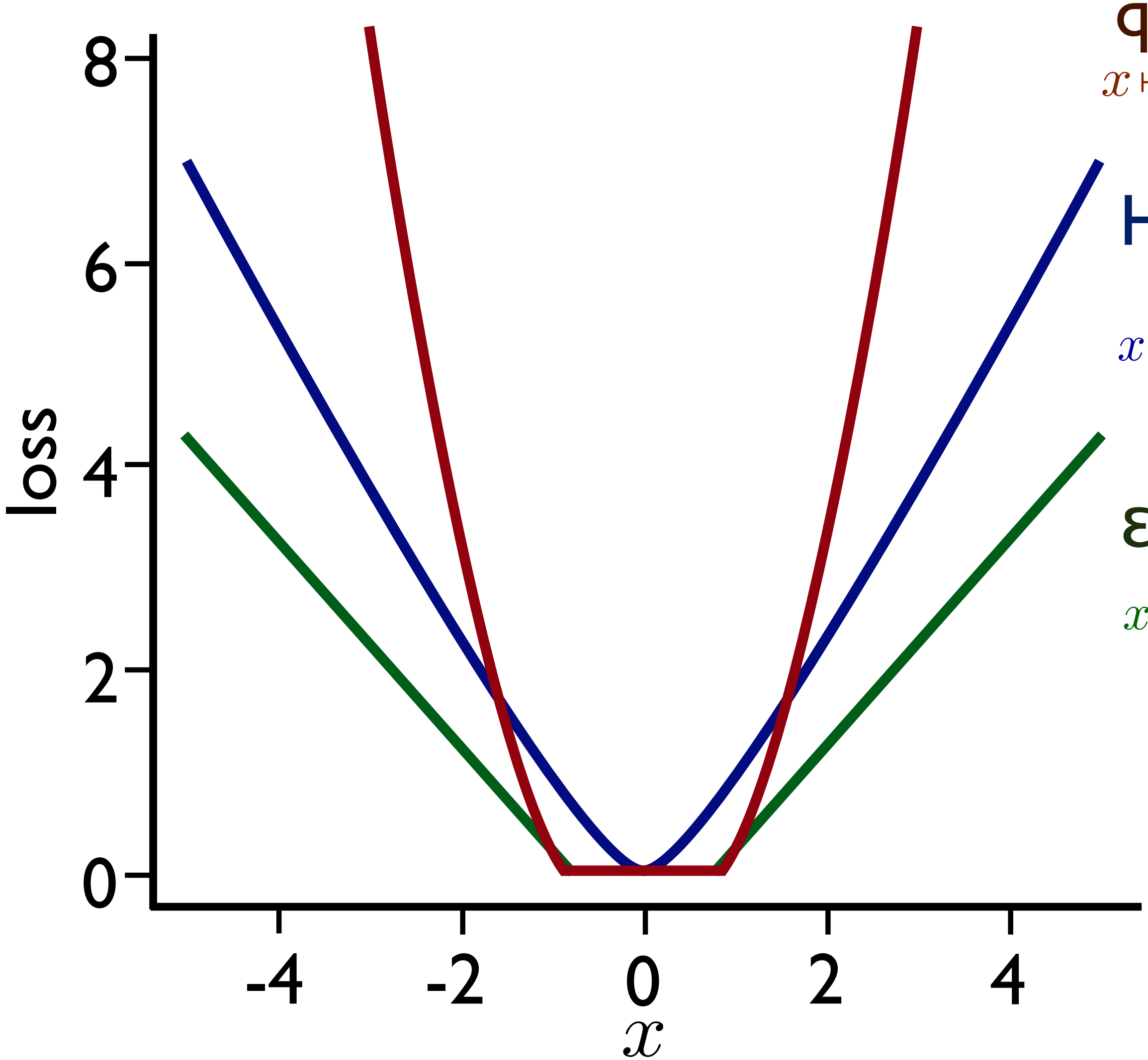
with $b=\begin{cases}-\sum_{i=1}^m(\alpha'_j-\alpha_j)K(x_j,x_i)+y_i+\epsilon & \text{when } 0<\alpha_i<C\\ -\sum_{i=1}^m(\alpha'_j-\alpha_j)K(x_j,x_i)+y_i-\epsilon & \text{when } 0<\alpha'_i<C.\end{cases}$

- Support vectors: points strictly outside the tube.

Notes

- Advantages:
  - strong theoretical guarantees (for that loss).
  - sparser solution.
  - use of kernels.
- Disadvantages:
  - selection of two parameters: C and ϵ. Heuristics:
    - search C near maximum y, ϵ near average difference of ys, measure of no. of SVs.
  - large matrices: low-rank approximations of kernel matrix.

Alternative Loss Functions



quadratic ε-insensitive

$$x \mapsto \max(0, |x| - \epsilon)^2$$

Huber

$$x \mapsto \begin{cases} x^2 & \text{if } |x| \leq c \\ 2c|x| - c^2 & \text{otherwise.} \end{cases}$$

ε-insensitive

$$x \mapsto \max(0, |x| - \epsilon)$$

SVR - Quadratic Loss

- Optimization problem:

$$\max_{\boldsymbol{\alpha},\boldsymbol{\alpha}'} \; -\epsilon(\boldsymbol{\alpha}'+\boldsymbol{\alpha})^\top \mathbf{1} + (\boldsymbol{\alpha}'-\boldsymbol{\alpha})^\top \mathbf{y} - \frac{1}{2}(\boldsymbol{\alpha}'-\boldsymbol{\alpha})^\top \left(\mathbf{K}+\frac{1}{C}\mathbf{I}\right)(\boldsymbol{\alpha}'-\boldsymbol{\alpha})$$

$$\text{subject to: } (\boldsymbol{\alpha} \geq \mathbf{0}) \wedge (\boldsymbol{\alpha} \geq \mathbf{0}) \wedge (\boldsymbol{\alpha}'-\boldsymbol{\alpha})^\top \mathbf{1} = 0) \,.$$

- Solution:

$$h(x) = \sum_{i=1}^{m} (\alpha'_i - \alpha_i) K(\mathbf{x}_i, \mathbf{x}) + b$$

with $b = \begin{cases} -\sum_{i=1}^{m} (\alpha'_j - \alpha_j) K(x_j, x_i) + y_i + \epsilon & \text{when } 0 < \alpha_i \wedge \xi_i = 0 \\ -\sum_{i=1}^{m} (\alpha'_j - \alpha_j) K(x_j, x_i) + y_i - \epsilon & \text{when } 0 < \alpha'_i \wedge \xi'_i = 0. \end{cases}$

- Support vectors: points strictly outside the tube.
- For $\epsilon = 0$, coincides with KRR.

ε-Insensitive Bound - Kernel-Based Hypotheses

- Theorem: Let $K\colon X \times X \to \mathbb{R}$ be a PDS kernel and let $\Phi\colon X \to H$ be a feature mapping associated to K. Let $H = \{\mathbf{x} \mapsto \mathbf{w} \cdot \Phi(x) \colon \|\mathbf{w}\|_H \leq \Lambda\}$. Assume $K(x, x) \leq R^2$ and $|f(x)| \leq \Gamma R$ for all $x \in X$. Then, for any $\delta > 0$, with probability at least $1 - \delta$, for any $h \in H$,

$$\mathrm{E}[|h(x) - f(x)|_\epsilon] \leq \widehat{\mathrm{E}}[|h(x) - f(x)|_\epsilon] + \frac{R\Lambda}{\sqrt{m}}\left[2 + \left(\frac{\Gamma}{\Lambda} + 1\right)\sqrt{\frac{\log \frac{1}{\delta}}{2}}\right].$$

$$\mathrm{E}[|h(x)-f(x)|_\epsilon] \leq \widehat{\mathrm{E}}[|h(x)-f(x)|_\epsilon]+\frac{\Lambda R}{\sqrt{m}}\left[2\sqrt{\frac{\mathrm{Tr}[\mathbf{K}]/R^2}{m}}+3\left(\frac{\Gamma}{\Lambda} + 1\right)\sqrt{\frac{\log \frac{2}{\delta}}{2}}\right].$$

ε-Insensitive Bound - Kernel-Based Hypotheses

- Proof: Let $H_\epsilon = \{x \mapsto |h(x) - f(x)|_\epsilon : h \in H\}$ and let H' be defined by $H' = \{x \mapsto h(x) - f(x) : h \in H\}$.
  - The function $\Phi_\epsilon : x \mapsto |x|_\epsilon$ is 1-Lipschitz and $\Phi_\epsilon(0) = 0$. Thus, by the contraction lemma,
  $$\widehat{\mathfrak{R}}_S(H_\epsilon) \leq \widehat{\mathfrak{R}}_S(H').$$
  - Since $\widehat{\mathfrak{R}}_S(H') = \widehat{\mathfrak{R}}_S(H)$ (see proof for Rademacher Complexity of L_p Loss), this shows that $\widehat{\mathfrak{R}}_S(H_\epsilon) \leq \widehat{\mathfrak{R}}_S(H)$.
  - The rest is a direct application of the Rademacher Complexity Regression Bound (this lecture).

On-line Regression

- On-line version of batch algorithms:
  - stochastic gradient descent.
  - primal or dual.
- Examples:
  - Mean squared error function: Widrow-Hoff (or LMS) algorithm (Widrow and Hoff, 1995).
  - SVR ε-insensitive (dual) linear or quadratic function: on-line SVR.

Widrow-Hoff

(Widrow and Hoff, 1988)

$\textsc{WidrowHoff}(\mathbf{w}_0)$

$\mathbf{w}_1 \leftarrow \mathbf{w}_0 \qquad \triangleright$ typically $\mathbf{w}_0 = \mathbf{0}$

for $t \leftarrow 1$ **to** T **do**

$\quad \textsc{Receive}(\mathbf{x}_t)$

$\quad \widehat{y}_t \leftarrow \mathbf{w}_t \cdot \mathbf{x}_t$

$\quad \textsc{Receive}(y_t)$

$\quad \mathbf{w}_{t+1} \leftarrow \mathbf{w}_t + 2\eta(\mathbf{w}_t \cdot \mathbf{x}_t - y_t)\mathbf{x}_t \qquad \triangleright\, \eta > 0$

return $\mathbf{w}_{T+1}$

Dual On-Line SVR

$(b=0)$ (Vijayakumar and Wu, 1988)

$\textsc{DualSVR}()$

$\boldsymbol{\alpha} \leftarrow \mathbf{0}$
$\boldsymbol{\alpha}' \leftarrow \mathbf{0}$
for $t \leftarrow 1$ **to** T **do**
  $\textsc{Receive}(x_t)$
  $\widehat{y}_t \leftarrow \sum_{s=1}^{T} (\alpha'_s - \alpha_s) K(x_s, x_t)$
  $\textsc{Receive}(y_t)$
  $\alpha'_{t+1} \leftarrow \alpha'_t + \min(\max(\eta(y_t - \widehat{y}_t - \epsilon), -\alpha'_t), C - \alpha'_t)$
  $\alpha_{t+1} \leftarrow \alpha_t + \min(\max(\eta(\widehat{y}_t - y_t - \epsilon), -\alpha_t), C - \alpha_t)$
return $\sum_{t=1}^{T} \alpha_t K(x_t, \cdot)$

This Lecture

- Generalization bounds
- Linear regression
- Kernel ridge regression
- Support vector regression
- Lasso

LASSO

(Tibshirani, 1996)

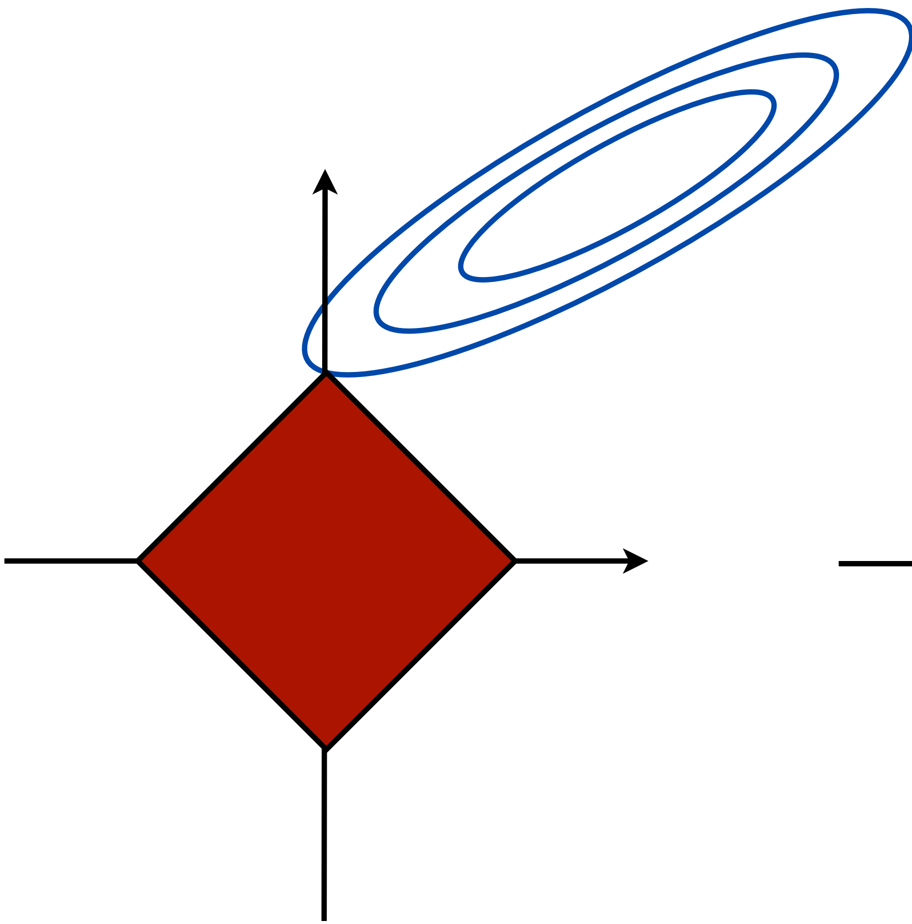
- Optimization problem: 'least absolute shrinkage and selection operator'.

$$\min_{\mathbf{w}} F(\mathbf{w}, b) = \lambda \|\mathbf{w}\|_1 + \sum_{i=1}^{m} (\mathbf{w} \cdot \mathbf{x}_i + b - y_i)^2 \,,$$

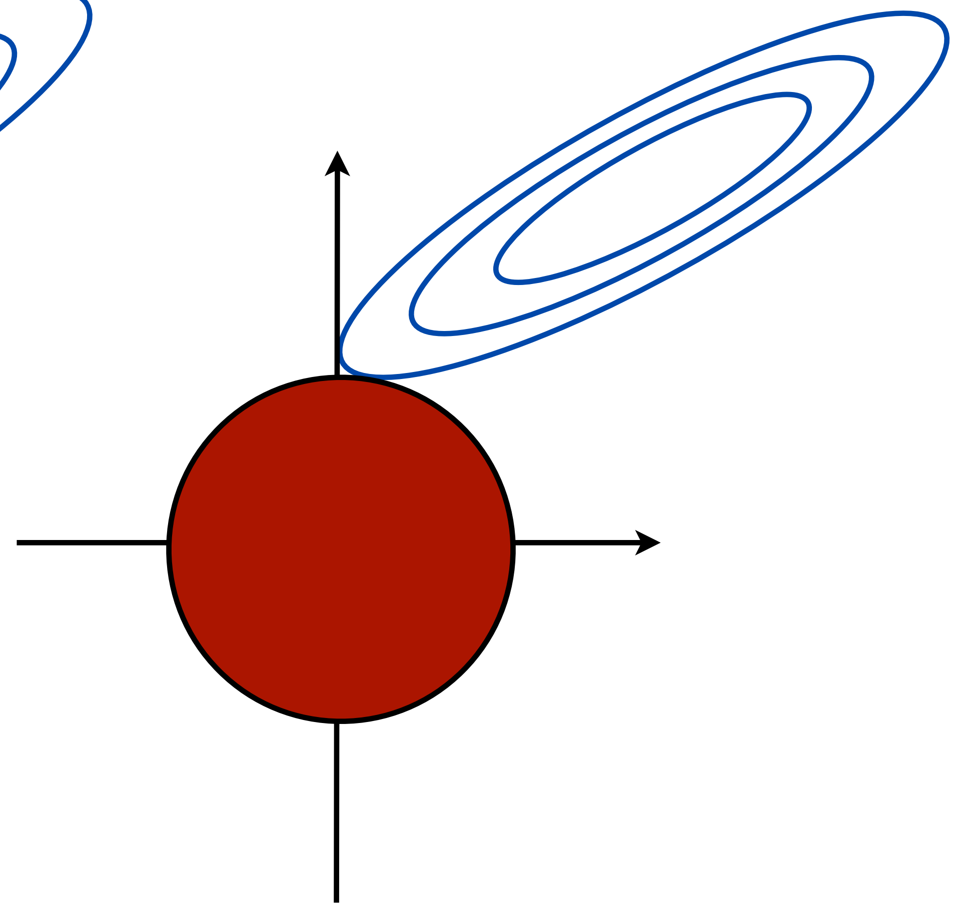
  where $\lambda \geq 0$ is a (regularization) parameter.

- Solution: equiv. convex quadratic program (QP).
  - general: standard QP solvers.
  - specific algorithm: LARS (least angle regression procedure), entire path of solutions.

Sparsity of L1 regularization



L1 regularization



L2 regularization

Sparsity Guarantee

- Rademacher complexity of L1-norm bounded linear hypotheses:

$$
\begin{aligned}
\widehat{\mathfrak{R}}_S(H) &= \frac{1}{m} \operatorname*{E}_{\boldsymbol{\sigma}} \left[\sup_{\|\mathbf{w}\|_1 \le \Lambda_1} \sum_{i=1}^m \sigma_i \mathbf{w} \cdot \mathbf{x}_i \right] \\
&= \frac{\Lambda_1}{m} \operatorname*{E}_{\boldsymbol{\sigma}} \left[\left\| \sum_{i=1}^m \sigma_i \mathbf{x}_i \right\|_\infty \right] && \text{(by definition of the dual norm)} \\
&= \frac{\Lambda_1}{m} \operatorname*{E}_{\boldsymbol{\sigma}} \left[\max_{j \in [1,N]} \left| \sum_{i=1}^m \sigma_i x_{ij} \right| \right] && \text{(by definition of } \| \cdot \|_\infty) \\
&= \frac{\Lambda_1}{m} \operatorname*{E}_{\boldsymbol{\sigma}} \left[\max_{j \in [1,N]} \max_{s \in \{-1,+1\}} s \sum_{i=1}^m \sigma_i x_{ij} \right] && \text{(by definition of } | \cdot |) \\
&= \frac{\Lambda_1}{m} \operatorname*{E}_{\boldsymbol{\sigma}} \left[\sup_{\mathbf{z} \in A} \sum_{i=1}^m \sigma_i z_i \right] \le r_\infty \Lambda_1 \sqrt{\frac{2 \log(2N)}{m}}. && \text{(Massart's lemma)}
\end{aligned}
$$

Notes

- Advantages:
  - theoretical guarantees.
  - sparse solution.
  - feature selection.
- Drawbacks:
  - no natural use of kernels.
  - no closed-form solution (not necessary, but can be convenient for theoretical analysis).

Regression

- Many other families of algorithms: including
  - neural networks.
  - decision trees (see multi-class lecture).
  - boosting trees for regression.

References

- Corinna Cortes, Mehryar Mohri, and Jason Weston. A General Regression Framework for Learning String-to-String Mappings. In *Predicting Structured Data*. The MIT Press, 2007.
- Efron, B., Johnstone, I., Hastie, T. and Tibshirani, R. (2002). Least angle regression. *Annals of Statistics* 2003.
- Arthur Hoerl and Robert Kennard. Ridge Regression: biased estimation of nonorthogonal problems. *Technometrics*, 12:55-67, 1970.
- C. Saunders and A. Gammerman and V. Vovk, Ridge Regression Learning Algorithm in Dual Variables, In *ICML '98*, pages 515--521, 1998.
- Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of Royal Statistical Society*, pages B. 58:267-288, 1996.
- David Pollard. *Convergence of Stochastic Processes*. Springer, New York, 1984.
- David Pollard. *Empirical Processes: Theory and Applications*. Institute of Mathematical Statistics, 1990.

References

- Sethu Vijayakumar and Si Wu. Sequential support vector classifiers and regression. In Proceedings of the International Conference on Soft Computing (SOCO'99), 1999.

- Vladimir N. Vapnik. *Estimation of Dependences Based on Empirical Data*. Springer, Basederlin, 1982.

- Vladimir N. Vapnik. *The Nature of Statistical Learning Theory.* Springer, 1995.

- Vladimir N. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, New York, 1998.

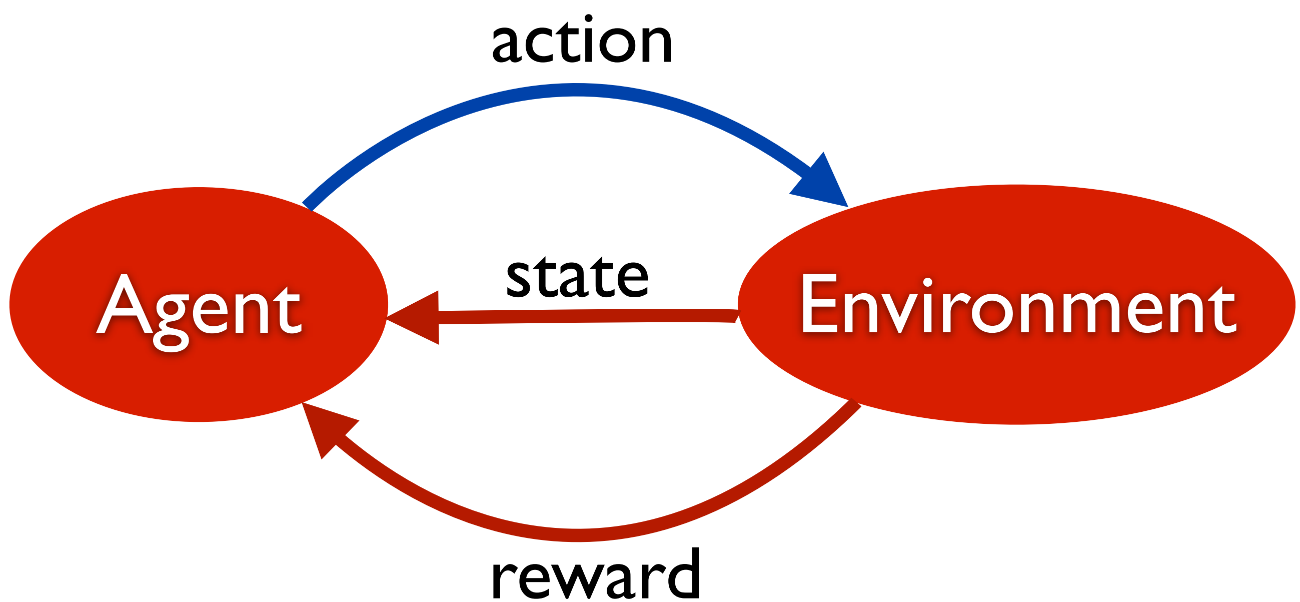
- Bernard Widrow and Ted Hoff. Adaptive Switching Circuits. *Neurocomputing: foundations of research*, pages 123-134, MIT Press, 1988.

Foundations of Machine Learning
Reinforcement Learning

Mehryar Mohri
Courant Institute and Google Research
mohri@cims.nyu.edu

Reinforcement Learning

- Agent exploring environment.
- Interactions with environment:



- Problem: find action policy that maximizes cumulative reward over the course of interactions.

Key Features

- Contrast with supervised learning:
  - no explicit labeled training data.
  - distribution defined by actions taken.
- Delayed rewards or penalties.
- RL trade-off:
  - exploration (of unknown states and actions) to gain more reward information; vs.
  - exploitation (of known information) to optimize reward.

Applications

- Robot control e.g., Robocup Soccer Teams (Stone et al., 1999), helicopter flight, autonomous driving.
- Board games, e.g., TD-Gammon (Tesauro, 1995), Go (Silver et al., 2016).
- Elevator scheduling (Crites and Barto, 1996).
- Ads placement, patient treatment.
- Telecommunications.
- Inventory management.
- Dynamic radio channel assignment.

This Lecture

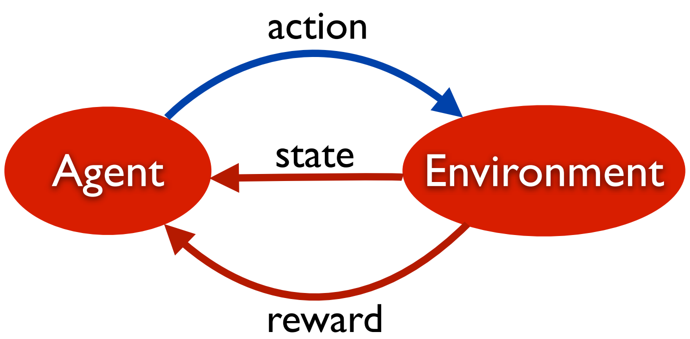
- Markov Decision Processes (MDPs)
- Planning
- Learning
- Multi-armed bandit problem

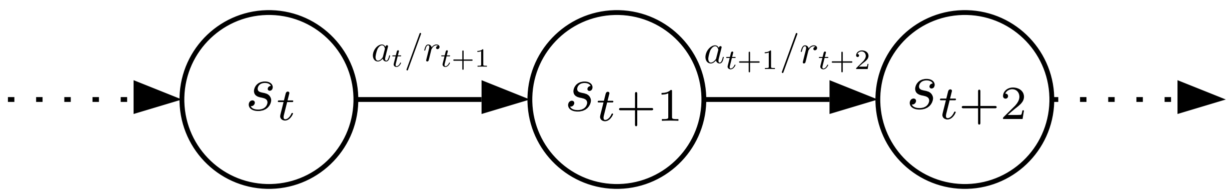
Markov Decision Process (MDP)

- Definition: a Markov Decision Process is defined by:
  - a set of decision epochs $\{0, \ldots, T\}$.
  - a set of states S, possibly infinite.
  - a start state or initial state s_0;
  - a set of actions A, possibly infinite.
  - a transition probability $\Pr[s'|s, a]$: distribution over destination states $s' = \delta(s, a)$.
  - a reward probability $\Pr[r'|s, a]$: distribution over rewards returned $r' = r(s, a)$.

Model

- State observed at time t : $s_t \in S$.
- Action taken at time t : $a_t \in A$.
- State reached $s_{t+1} = \delta(s_t, a_t)$.
- Reward received: $r_{t+1} = r(s_t, a_t)$.

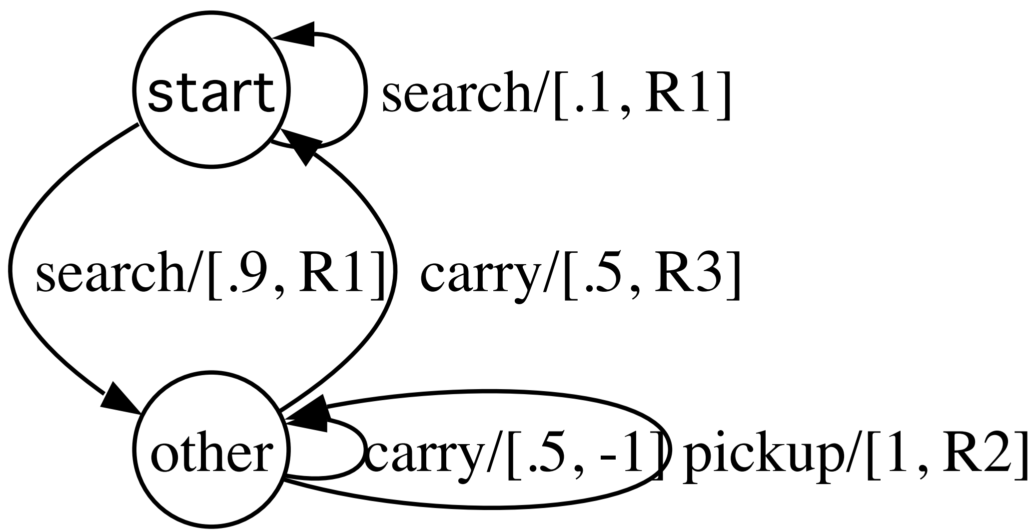




MDPs - Properties

- Finite MDPs: A and S finite sets.
- Finite horizon when $T < \infty$.
- Reward $r(s, a)$: often deterministic function.

Example - Robot Picking up Balls



Policy

- **Definition**: a policy is a mapping $\pi\colon S \to A$.
- **Objective**: find policy π maximizing expected return.
  - finite horizon return: $\sum_{t=0}^{T-1} r\big(s_t, \pi(s_t)\big)$.
  - infinite horizon return: $\sum_{t=0}^{+\infty} \gamma^t r\big(s_t, \pi(s_t)\big)$.
- **Theorem**: for any finite MDP, there exists an optimal policy (for any start state).

Policy Value

- **Definition**: the value of a policy π at state s is
  - finite horizon:

$$V_\pi(s) = \mathrm{E}\left[\sum_{t=0}^{T-1} r\big(s_t, \pi(s_t)\big) \,\Big|\, s_0 = s\right].$$

  - infinite horizon: discount factor $\gamma \in [0, 1)$,

$$V_\pi(s) = \mathrm{E}\left[\sum_{t=0}^{+\infty} \gamma^t r\big(s_t, \pi(s_t)\big) \,\Big|\, s_0 = s\right].$$

- **Problem**: find policy π with maximum value for all states.

Policy Evaluation

- Analysis of policy value:

$$
\begin{aligned}
V_\pi(s) &= \mathrm{E}\left[\sum_{t=0}^{+\infty} \gamma^t r\big(s_t, \pi(s_t)\big) \,\Big|\, s_0 = s\right]. \\
&= \mathrm{E}[r(s, \pi(s))] + \gamma\, \mathrm{E}\left[\sum_{t=0}^{+\infty} \gamma^t r\big(s_{t+1}, \pi(s_{t+1})\big) \,\Big|\, s_0 = s\right] \\
&= \mathrm{E}[r(s, \pi(s)] + \gamma\, \mathrm{E}[V_\pi(\delta(s, \pi(s)))].
\end{aligned}
$$

- Bellman equations (system of linear equations):

$$
\boxed{V_\pi(s) = \mathrm{E}[r(s, \pi(s)] + \gamma \sum_{s'} \Pr[s'|s, \pi(s)] V_\pi(s').}
$$

Bellman Equation - Existence and Uniqueness

- Notation:
  - transition probability matrix $\mathbf{P}_{s,s'} = \Pr[s'|s, \pi(s)]$.
  - value column matrix $\mathbf{V} = V_\pi(s)$.
  - expected reward column matrix: $\mathbf{R} = \mathrm{E}[r(s, \pi(s)]$.
- Theorem: for a finite MDP, Bellman's equation admits a unique solution given by

$$\mathbf{V}_0 = (\mathbf{I} - \gamma \mathbf{P})^{-1} \mathbf{R}.$$

Bellman Equation - Existence and Uniqueness

- Proof: Bellman's equation rewritten as

$$\mathbf{V} = \mathbf{R} + \gamma \mathbf{P}\mathbf{V}.$$

  - $\mathbf{P}$ is a stochastic matrix, thus,

$$\|\mathbf{P}\|_\infty = \max_s \sum_{s'} |\mathbf{P}_{ss'}| = \max_s \sum_{s'} \Pr[s'|s, \pi(s)] = 1.$$

  - This implies that $\|\gamma \mathbf{P}\|_\infty = \gamma < 1$. The eigenvalues of $\gamma \mathrm{P}$ are all less than one and $(\mathbf{I} - \gamma \mathbf{P})$ is invertible.

- Notes: general shortest distance problem (MM, 2002).

Optimal Policy

- Definition: policy π^* with maximal value for all states $s \in S$.
  - value of π^* (optimal value):
$$\forall s \in S, V_{\pi^*}(s) = \max_{\pi} V_{\pi}(s).$$
  - optimal state-action value function: expected return for taking action a at state s and then following optimal policy.
$$\begin{aligned} Q^*(s,a) &= \mathrm{E}[r(s,a)] + \gamma\, \mathrm{E}[V^*(\delta(s,a))] \\ &= \mathrm{E}[r(s,a)] + \gamma \sum_{s' \in S} \Pr[s' \mid s,a] V^*(s'). \end{aligned}$$

Optimal Values - Bellman Equations

- Property: the following equalities hold:

$$\forall s \in S,\ V^*(s) = \max_{a \in A} Q^*(s, a).$$

- Proof: by definition, for all s, $V^*(s) \leq \max_{a \in A} Q^*(s, a)$.
  - If for some s we had $V^*(s) < \max_{a \in A} Q^*(s, a)$, then maximizing action would define a better policy.

- Thus,

$$\boxed{V^*(s) = \max_{a \in A} \Big\{ \mathrm{E}[r(s, a)] + \gamma \sum_{s' \in S} \Pr[s'|s, a] V^*(s') \Big\}.}$$

This Lecture

- Markov Decision Processes (MDPs)
- Planning
- Learning
- Multi-armed bandit problem

Known Model

- Setting: environment model known.
- Problem: find optimal policy.
- Algorithms:
  - value iteration.
  - policy iteration.
  - linear programming.

Value Iteration Algorithm

$$\mathbf{\Phi}(\mathbf{V})(s) = \max_{a \in A} \Big\{ \mathrm{E}[r(s,a)] + \gamma \sum_{s' \in S} \Pr[s'|s,a] V(s') \Big\}.$$

$$\mathbf{\Phi}(\mathbf{V}) = \max_{\pi} \{ \mathbf{R}_\pi + \gamma \mathbf{P}_\pi \mathbf{V} \}.$$

$\textsc{ValueIteration}(\mathbf{V}_0)$

1 $\quad \mathbf{V} \leftarrow \mathbf{V}_0 \quad \triangleright\, \mathbf{V}_0$ arbitrary value

2 $\quad$ **while** $\|\mathbf{V} - \mathbf{\Phi}(\mathbf{V})\| \geq \frac{(1-\gamma)\epsilon}{\gamma}$ **do**

3 $\qquad\quad \mathbf{V} \leftarrow \mathbf{\Phi}(\mathbf{V})$

4 $\quad$ **return** $\mathbf{\Phi}(\mathbf{V})$

VI Algorithm - Convergence

- Theorem: for any initial value $\mathbf{V}_0$, the sequence defined by $\mathbf{V}_{n+1} = \mathbf{\Phi}(\mathbf{V}_n)$ converge to $\mathbf{V}^*$.
- Proof: we show that $\mathbf{\Phi}$ is γ-contracting for $\|\cdot\|_\infty$

→ existence and uniqueness of fixed point for $\mathbf{\Phi}$.

  - for any $s \in S$, let $a^*(s)$ be the maximizing action defining $\mathbf{\Phi}(\mathbf{V})(s)$. Then, for $s \in S$ and any $\mathbf{U}$,

$$\begin{aligned}
\mathbf{\Phi}(\mathbf{V})(s) - \mathbf{\Phi}(\mathbf{U})(s) &\leq \mathbf{\Phi}(\mathbf{V})(s) - \Big(\mathrm{E}[r(s, a^*(s))] + \gamma \sum_{s' \in S} \Pr[s' \mid s, a^*(s)] \mathbf{U}(s') \Big) \\
&= \gamma \sum_{s' \in S} \Pr[s' | s, a^*(s)] [\mathbf{V}(s') - \mathbf{U}(s')] \\
&\leq \gamma \sum_{s' \in S} \Pr[s' | s, a^*(s)] \|\mathbf{V} - \mathbf{U}\|_\infty = \gamma \|\mathbf{V} - \mathbf{U}\|_\infty .
\end{aligned}$$

Complexity and Optimality

- **Complexity**: convergence in $O(\log \frac{1}{\epsilon})$. Observe that

$$\|\mathbf{V}_{n+1} - \mathbf{V}_n\|_\infty \leq \gamma \|\mathbf{V}_n - \mathbf{V}_{n-1}\|_\infty \leq \gamma^n \|\mathbf{\Phi}(\mathbf{V}_0) - \mathbf{V}_0\|_\infty.$$

Thus, $\gamma^n \|\mathbf{\Phi}(\mathbf{V}_0) - \mathbf{V}_0\|_\infty \leq \dfrac{(1-\gamma)\epsilon}{\gamma} \Rightarrow n = O\Big(\log \dfrac{1}{\epsilon}\Big).$

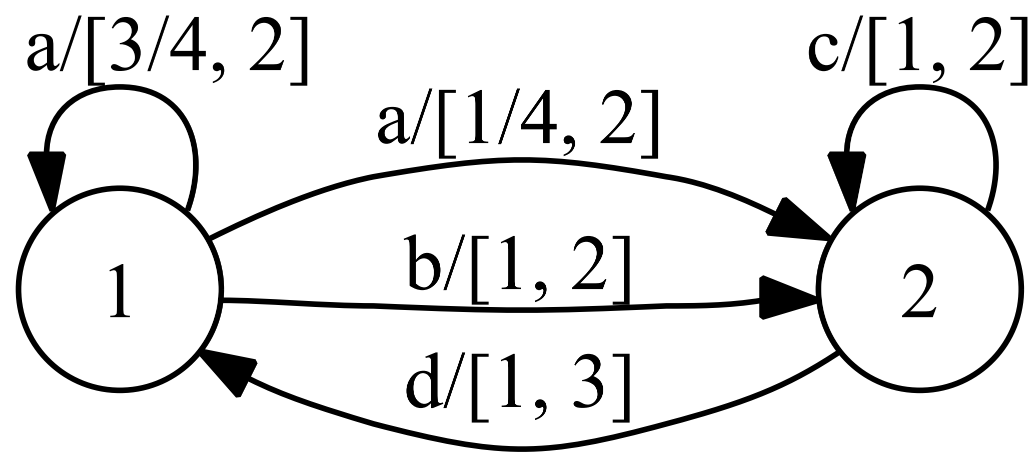
- ϵ**-Optimality**: let $\mathbf{V}_{n+1}$ be the value returned. Then,

$$\begin{aligned}\|\mathbf{V}^* - \mathbf{V}_{n+1}\|_\infty &\leq \|\mathbf{V}^* - \mathbf{\Phi}(\mathbf{V}_{n+1})\|_\infty + \|\mathbf{\Phi}(\mathbf{V}_{n+1}) - \mathbf{V}_{n+1}\|_\infty \\ &\leq \gamma \|\mathbf{V}^* - \mathbf{V}_{n+1}\|_\infty + \gamma \|\mathbf{V}_{n+1} - \mathbf{V}_n\|_\infty.\end{aligned}$$

Thus,

$$\|\mathbf{V}^* - \mathbf{V}_{n+1}\|_\infty \leq \frac{\gamma}{1-\gamma} \|\mathbf{V}_{n+1} - \mathbf{V}_n\|_\infty \leq \epsilon.$$

VI Algorithm - Example



$$\mathbf{V}_{n+1}(1) = \max\left\{2 + \gamma\Big(\frac{3}{4}\mathbf{V}_n(1) + \frac{1}{4}\mathbf{V}_n(2)\Big), 2 + \gamma\mathbf{V}_n(2)\right\}$$

$$\mathbf{V}_{n+1}(2) = \max\left\{3 + \gamma\mathbf{V}_n(1), 2 + \gamma\mathbf{V}_n(2)\right\}.$$

For $\mathbf{V}_0(1) = -1, \mathbf{V}_0(2) = 1, \gamma = 1/2$, $\mathbf{V}_1(1) = \mathbf{V}_1(2) = 5/2.$

But, $\mathbf{V}^*(1) = 14/3, \mathbf{V}^*(2) = 16/3.$

Policy Iteration Algorithm

PolicyIteration(π_0)

$\pi \leftarrow \pi_0$ $\quad \triangleright$ π_0 arbitrary policy
$\pi' \leftarrow$ NIL
while $(\pi \neq \pi')$ **do**
  $\mathbf{V} \leftarrow \mathbf{V}_\pi$ $\quad \triangleright$ policy evaluation: solve $(\mathbf{I} - \gamma \mathbf{P}_\pi)\mathbf{V} = \mathbf{R}_\pi$.
  $\pi' \leftarrow \pi$
  $\pi \leftarrow \operatorname{argmax}_\pi \{\mathbf{R}_\pi + \gamma \mathbf{P}_\pi \mathbf{V}\}$ $\quad \triangleright$ greedy policy improvement.
return π

PI Algorithm - Convergence

- **Theorem**: let $(\mathbf{V}_n)_{n \in \mathbb{N}}$ be the sequence of policy values computed by the algorithm, then,

$$\mathbf{V}_n \leq \mathbf{V}_{n+1} \leq \mathbf{V}^*.$$

- **Proof**: let π_{n+1} be the policy improvement at the nth iteration, then, by definition,

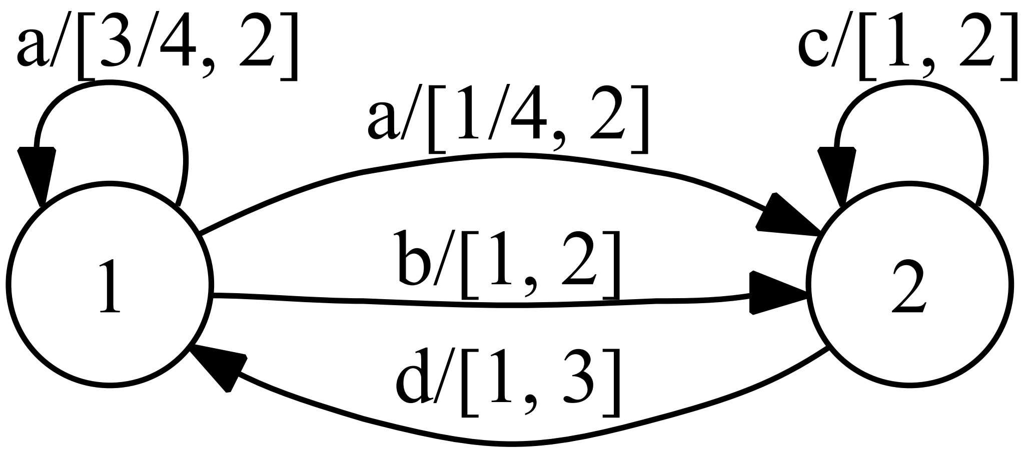
$$\mathbf{R}_{\pi_{n+1}} + \gamma \mathbf{P}_{\pi_{n+1}} \mathbf{V}_n \geq \mathbf{R}_{\pi_n} + \gamma \mathbf{P}_{\pi_n} \mathbf{V}_n = \mathbf{V}_n.$$

  - therefore, $\mathbf{R}_{\pi_{n+1}} \geq (\mathbf{I} - \gamma \mathbf{P}_{\pi_{n+1}}) \mathbf{V}_n$.
  - note that $(\mathbf{I} - \gamma \mathbf{P}_{\pi_{n+1}})^{-1}$ preserves ordering:
    $\mathbf{X} \geq \mathbf{0} \Rightarrow (\mathbf{I} - \gamma \mathbf{P}_{\pi_{n+1}})^{-1} \mathbf{X} = \sum_{k=0}^{\infty} (\gamma \mathbf{P}_{\pi_{n+1}})^k \mathbf{X} \geq \mathbf{0}$.
  - thus, $\mathbf{V}_{n+1} = (\mathbf{I} - \gamma \mathbf{P}_{\pi_{n+1}})^{-1} \mathbf{R}_{\pi_{n+1}} \geq \mathbf{V}_n$.

Notes

- Two consecutive policy values can be equal only at last iteration.
- The total number of possible policies is $|A|^{|S|}$, thus, this is the maximal possible number of iterations.
  - best upper bound known $O(\frac{|A|^{|S|}}{|S|})$.

PI Algorithm - Example



Initial policy: $\pi_0(1) = b, \pi_0(2) = c.$

Evaluation: $V_{\pi_0}(1) = 1 + \gamma V_{\pi_0}(2)$

$$V_{\pi_0}(2) = 2 + \gamma V_{\pi_0}(2).$$

Thus, $V_{\pi_0}(1) = \dfrac{1+\gamma}{1-\gamma} \quad V_{\pi_0}(2) = \dfrac{2}{1-\gamma}.$

VI and PI Algorithms - Comparison

- Theorem: let $(\mathbf{U}_n)_{n\in\mathbb{N}}$ be the sequence of policy values generated by the VI algorithm, and $(\mathbf{V}_n)_{n\in\mathbb{N}}$ the one generated by the PI algorithm. If $\mathbf{U}_0 = \mathbf{V}_0$, then,
$$\forall n \in \mathbb{N}, \; \mathbf{U}_n \leq \mathbf{V}_n \leq \mathbf{V}^*.$$
- Proof: we first show that $\mathbf{\Phi}$ is monotonic. Let $\mathbf{U}$ and $\mathbf{V}$ be such that $\mathbf{U} \leq \mathbf{V}$ and let π be the policy such that $\mathbf{\Phi}(\mathbf{U}) = \mathbf{R}_\pi + \gamma \mathbf{P}_\pi \mathbf{U}$. Then,

$$\mathbf{\Phi}(\mathbf{U}) \leq \mathbf{R}_\pi + \gamma \mathbf{P}_\pi \mathbf{V} \leq \max_{\pi'} \{\mathbf{R}'_\pi + \gamma \mathbf{P}'_\pi \mathbf{V}\} = \mathbf{\Phi}(\mathbf{V}).$$

VI and PI Algorithms - Comparison

- The proof is by induction on n. Assume $\mathbf{U}_n \leq \mathbf{V}_n$, then, by the monotonicity of $\mathbf{\Phi}$,

$$\mathbf{U}_{n+1} = \mathbf{\Phi}(\mathbf{U}_n) \leq \mathbf{\Phi}(\mathbf{V}_n) = \max_{\pi}\{\mathbf{R}_\pi + \gamma \mathbf{P}_\pi \mathbf{V}_n\}.$$

- Let π_{n+1} be the maximizing policy:

$$\pi_{n+1} = \operatorname*{argmax}_{\pi}\{\mathbf{R}_\pi + \gamma \mathbf{P}_\pi \mathbf{V}_n\}.$$

- Then,

$$\mathbf{\Phi}(\mathbf{V}_n) = \mathbf{R}_{\pi_{n+1}} + \gamma \mathbf{P}_{\pi_{n+1}} \mathbf{V}_n \leq \mathbf{R}_{\pi_{n+1}} + \gamma \mathbf{P}_{\pi_{n+1}} \mathbf{V}_{n+1} = \mathbf{V}_{n+1}.$$

Notes

- The PI algorithm converges in a smaller number of iterations than the VI algorithm due to the optimal policy.
- But, each iteration of the PI algorithm requires computing a policy value, i.e., solving a system of linear equations, which is more expensive to compute that an iteration of the VI algorithm.

Primal Linear Program

- LP formulation: choose $\alpha(s) > 0$, with $\sum_s \alpha(s) = 1$.

$$\min_{\mathbf{V}} \quad \sum_{s \in S} \alpha(s) V(s)$$

$$\text{subject to} \quad \forall s \in S, \forall a \in A, V(s) \geq \mathrm{E}[r(s,a)] + \gamma \sum_{s' \in S} \Pr[s'|s,a] V(s').$$

- Parameters:
  - number rows: $|S||A|$.
  - number of columns: $|S|$.

Dual Linear Program

- LP formulation:

$$\max_{\mathbf{x}} \quad \sum_{s \in S, a \in A} \mathrm{E}[r(s,a)]\, x(s,a)$$

$$\text{subject to} \quad \forall s \in S, \sum_{a \in A} x(s',a) = \alpha(s') + \gamma \sum_{s \in S, a \in A} \Pr[s'|s,a]\, x(s',a)$$

$$\forall s \in S, \forall a \in A, x(s,a) \geq 0.$$

- Parameters: more favorable number of rows.
  - number rows: $|S|$.
  - number of columns: $|S||A|$.

This Lecture

- Markov Decision Processes (MDPs)
- Planning
- Learning
- Multi-armed bandit problem

Problem

- Unknown model:
  - transition and reward probabilities not known.
  - realistic scenario in many practical problems, e.g., robot control.
- Training information: sequence of immediate rewards based on actions taken.
- Learning approches:
  - model-free: learn policy directly.
  - model-based: learn model, use it to learn policy.

Learning Approaches

- Two broad families:
  - model-based approaches: use samples based on interactions to learn P and r explicitly; next, use value iteration to learn policy.
  - model-free approaches: do not seek to learn model; instead, use samples to learn Q function; policy readily derived from Q.

Problem

- How do we estimate reward and transition probabilities?
  - use equations derived for policy value and Q-functions.
  - but, equations given in terms of some expectations.
  - $\longrightarrow$ instance of a stochastic approximation problem.

Stochastic Approximation

- Problem: find solution of $\mathbf{x} = H(\mathbf{x})$ with $\mathbf{x} \in \mathbb{R}^N$ while
  - $H(\mathbf{x})$ cannot be computed, e.g., H not accessible;
  - i.i.d. sample of noisy observations $H(\mathbf{x}_i) + \mathbf{w}_i$, available, $i \in [1, m]$, with $\mathrm{E}[\mathbf{w}] = 0$.
- Idea: algorithm based on iterative technique:

$$\begin{aligned}\mathbf{x}_{t+1} &= (1 - \alpha_t)\mathbf{x}_t + \alpha_t[H(\mathbf{x}_t) + \mathbf{w}_t] \\ &= \mathbf{x}_t + \alpha_t[H(\mathbf{x}_t) + \mathbf{w}_t - \mathbf{x}_t].\end{aligned}$$

  - more generally $\mathbf{x}_{t+1} = \mathbf{x}_t + \alpha_t D(\mathbf{x}_t, \mathbf{w}_t)$.

Mean Estimation

- Theorem: Let X be a random variable taking values in $[0, 1]$ and let $x_0, \ldots, x_m$ be i.i.d. values of X. Define the sequence $(\mu_m)_{m \in \mathbb{N}}$ by

$$\mu_{m+1} = (1 - \alpha_m)\mu_m + \alpha_m x_m \quad \text{with } \mu_0 = x_0.$$

Then, for $\alpha_m \in [0, 1]$, with $\sum_{m \geq 0} \alpha_m = +\infty$ and $\sum_{m \geq 0} \alpha_m^2 < +\infty$,

$$\boxed{\mu_m \xrightarrow{\text{a.s}} \mathrm{E}[X].}$$

Proof

- Proof: By the independence assumption, for $m \geq 0$,

$$\begin{aligned}\mathrm{Var}[\mu_{m+1}] &= (1-\alpha_m)^2 \mathrm{Var}[\mu_m] + \alpha_m^2 \mathrm{Var}[x_m] \\ &\leq (1-\alpha_m)\mathrm{Var}[\mu_m] + \alpha_m^2.\end{aligned}$$

  - We have $\alpha_m \to 0$ since $\sum_{m \geq 0} \alpha_m^2 < +\infty$.
  - Let $\epsilon > 0$ and suppose there exists $N \in \mathbb{N}$ such that for all $m \geq N$, $\mathrm{Var}[\mu_m] \geq \epsilon$. Then, for $m \geq N$,

$$\mathrm{Var}[\mu_{m+1}] \leq \mathrm{Var}[\mu_m] - \alpha_m \epsilon + \alpha_m^2,$$

which implies $\mathrm{Var}[\mu_{m+N}] \leq \underbrace{\mathrm{Var}[\mu_N] - \epsilon \sum_{n=N}^{m+N} \alpha_n + \sum_{n=N}^{m+N} \alpha_n^2}_{\to -\infty \text{ when } m \to \infty}$,

contradicting $\mathrm{Var}[\mu_{m+N}] \geq 0$.

Mean Estimation

- Thus, for all $N \in \mathbb{N}$ there exists $m_0 \geq N$ such that $\mathrm{Var}[\mu_{m_0}] < \epsilon$. Choose N large enough so that $\forall m \geq N, \alpha_m \leq \epsilon$. Then,
$$\mathrm{Var}[\mu_{m_0+1}] \leq (1 - \alpha_{m_0})\epsilon + \epsilon\alpha_{m_0} = \epsilon.$$
- Therefore, $\mu_m \leq \epsilon$ for all $m \geq m_0$ (L_2 convergence).

Notes

- special case: $\alpha_m = \frac{1}{m}$.
  - Strong law of large numbers.
- Connection with stochastic approximation.

TD(0) Algorithm

- Idea: recall Bellman's linear equations giving V

$$\begin{aligned} V_\pi(s) &= \mathrm{E}[r(s,\pi(s)] + \gamma \sum_{s'} \Pr[s'|s,\pi(s)]V_\pi(s') \\ &= \mathop{\mathrm{E}}_{s'}\big[r(s,\pi(s)) + \gamma V_\pi(s')|s\big]. \end{aligned}$$

- Algorithm: temporal difference (TD).
  - sample new state s'.
  - update: α depends on number of visits of s.

$$\begin{aligned} V(s) &\leftarrow (1-\alpha)V(s) + \alpha[r(s,\pi(s)) + \gamma V(s')] \\ &= V(s) + \alpha[\underbrace{r(s,\pi(s)) + \gamma V(s') - V(s)}_{\text{temporal difference of } V \text{ values}}]. \end{aligned}$$

TD(0) Algorithm

TD(0)()

$\mathbf{V} \leftarrow \mathbf{V}_0$ ▷ initialization.
for $t \leftarrow 0$ **to** T **do**
  $s \leftarrow \textsc{SelectState}()$
  for each step of epoch t **do**
    $r' \leftarrow \textsc{Reward}(s, \pi(s))$
    $s' \leftarrow \textsc{NextState}(\pi, s)$
    $V(s) \leftarrow (1 - \alpha)V(s) + \alpha[r' + \gamma V(s')]$
    $s \leftarrow s'$
return $\mathbf{V}$

Q-Learning Algorithm

- Idea: assume deterministic rewards.

$$\begin{aligned} Q^*(s, a) &= \mathrm{E}[r(s, a)] + \gamma \sum_{s' \in S} \Pr[s' \mid s, a] V^*(s') \\ &= \mathop{\mathrm{E}}_{s'}[r(s, a) + \gamma \max_{a \in A} Q^*(s', a)] \end{aligned}$$

- Algorithm: $\alpha \in [0, 1]$ depends on number of visits.
  - sample new state s'.
  - update:

$$Q(s, a) \leftarrow (1 - \alpha) Q(s, a) + \alpha[r(s, a) + \gamma \max_{a' \in A} Q(s', a')].$$

Q-Learning Algorithm

(Watkins, 1989; Watkins and Dayan 1992)

Q-LEARNING(π)

$Q \leftarrow Q_0$ $\quad \triangleright$ initialization, e.g., $Q_0 = 0$.
for $t \leftarrow 0$ **to** T **do**
  $s \leftarrow$ SELECTSTATE()
  for each step of epoch t **do**
    $a \leftarrow$ SELECTACTION(π, s) $\triangleright$ policy π derived from Q, e.g., ϵ-greedy.
    $r' \leftarrow$ REWARD(s, a)
    $s' \leftarrow$ NEXTSTATE(s, a)
    $Q(s, a) \leftarrow Q(s, a) + \alpha\big[r' + \gamma \max_{a'} Q(s', a') - Q(s, a)\big]$
    $s \leftarrow s'$
return Q

Notes

- Can be viewed as a stochastic formulation of the value iteration algorithm.
- Convergence for any policy so long as states and actions visited infinitely often and parameter chosen as in mean estimation theorem.
- How to choose the action at each iteration? Maximize reward? Explore other actions?
- Q-learning is an off-policy method: no control over the policy; estimates and evaluates policy using experience from following different policy.

Policies

- Epsilon-greedy strategy:
  - with probability $1-\epsilon$ greedy action from s ;
  - with probability ϵ random action.
- Epoch-dependent strategy (Boltzmann exploration):

$$p_t(a|s,Q) = \frac{e^{\frac{Q(s,a)}{\tau_t}}}{\sum_{a' \in A} e^{\frac{Q(s,a')}{\tau_t}}},$$

  - $\tau_t \rightarrow 0$: greedy selection.
  - larger τ_t : random action.

Convergence of Q-Learning

- Theorem: consider a finite MDP. Assume that for all $s \in S$ and $a \in A$, $\sum_{t=0}^{\infty} \alpha_t(s,a) = \infty, \sum_{t=0}^{\infty} \alpha_t^2(s,a) < \infty$ with $\alpha_t(s,a) \in [0,1]$. Then, the Q-learning algorithm converges to the optimal value Q^* (with probability one).
    - note: the conditions on $\alpha_t(s,a)$ impose that each state-action pair is visited infinitely many times.

This Lecture

- Markov Decision Processes (MDPs)
- Planning
- Learning
- Multi-armed bandit problem

Multi-Armed Bandit Problem

(Robbins, 1952)

- Problem: gambler must decide which arm of a N-slot machine to pull to maximize his total reward in a series of trials.
  - stochastic setting: N lever reward distributions.
  - adversarial setting: reward selected by adversary aware of all the past.

Applications

- Clinical trials.
- Adaptive routing.
- Ads placement on pages.
- Games.

Multi-Armed Bandit Game

- For $t = 1$ to T do
  - adversary determines outcome $y_t \in Y$.
  - player selects probability distribution p_t and pulls lever $I_t \in \{1, \ldots, N\}$, $I_t \sim p_t$.
  - player incurs loss $L(I_t, y_t)$ (adversary is informed of p_t and I_t.
- Objective: minimize regret

$$\text{Regret}(T) = \sum_{t=1}^{T} L(I_t, y_t) - \min_{i=1,\ldots,N} \sum_{t=1}^{T} L(i, y_t).$$

Notes

- Player is informed only of the loss (or reward) corresponding to his own action.
- Adversary knows past but not action selected.
- Stochastic setting: loss $(L(1, y_t), \ldots, L(N, y_t))$ drawn according to some distribution $D = D_1 \otimes \cdots \otimes D_N$. Regret definition modified by taking expectations.
- Exploration/Exploitation trade-off: playing the best arm found so far versus seeking to find an arm with a better payoff.

Notes

- Equivalent views:
  - special case of learning with partial information.
  - one-state MDP learning problem.
- Simple strategy: ϵ-greedy: play arm with best empirical reward with probability $1-\epsilon_t$, random arm with probability ϵ_t.

Exponentially Weighted Average

- **Algorithm**: Exp3, defined for $\eta, \gamma > 0$ by

$$p_{i,t} = (1-\gamma) \frac{\exp\big(-\eta \sum_{s=1}^{t-1} \widehat{l}_{i,t}\big)}{\sum_{i=1}^{N} \exp\big(-\eta \sum_{s=1}^{t-1} \widehat{l}_{i,t}\big)} + \frac{\gamma}{N},$$

with $\forall i \in [1, N]$, $\widehat{l}_{i,t} = \frac{L(I_t, y_t)}{p_{I_t, t}} 1_{I_t = i}$.

- **Guarantee**: expected regret of

$$O(\sqrt{NT \log N}).$$

Exponentially Weighted Average

- Proof: similar to the one for the Exponentially Weighted Average with the additional observation that:

$$\mathrm{E}[\widehat{l}_{i,t}] = \sum_{i=1}^{N} p_{i,t} \frac{L(I_t, y_t)}{p_{I_t,t}} 1_{I_t = i} = L(i, y_t).$$

References

- Dimitri P. Bertsekas. Dynamic Programming and Optimal Control. 2 vols. Belmont, MA: Athena Scientific, 2007.
- Mehryar Mohri. Semiring Frameworks and Algorithms for Shortest-Distance Problems. *Journal of Automata, Languages and Combinatorics*, 7(3):321-350, 2002.
- Martin L. Puterman *Markov decision processes: discrete stochastic dynamic programming*. Wiley-Interscience, New York, 1994.
- Robbins, H. (1952), "Some aspects of the sequential design of experiments", Bulletin of the American Mathematical Society 58 (5): 527–535.
- Sutton, Richard S., and Barto, Andrew G. *Reinforcement Learning: An Introduction*. MIT Press, 1998.

References

- Gerald Tesauro. *Temporal Difference Learning and TD-Gammon*. Communications of the ACM 38 (3), 1995.

- Watkins, Christopher J. C. H. *Learning from Delayed Rewards*. Ph.D. thesis, Cambridge University, 1989.

- Christopher J. C. H. Watkins and Peter Dayan. *Q-learning*. Machine Learning, Vol. 8, No. 3-4, 1992.