The Role of Convexity in Prediction Problems

Peter Bartlett
Division of Computer Science and Department of Statistics
UC Berkeley

Joint work with: Olivier Bousquet, Mike Jordan, Jon McAuliffe, Shahar Mendelson

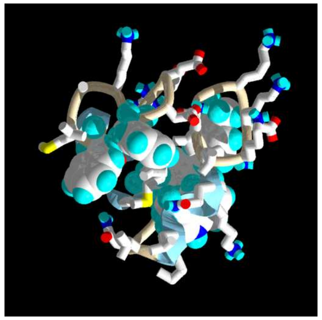
slides at http://www.cs.berkeley.edu/∼bartlett, under Talks

Handwritten Character Recognition

$X =$ grey-scale image of a character
$Y \in \{$‘a,’ ‘b,’ ‘c,’$\ldots$, ‘z’, ‘0,’ ‘1,’ $\ldots$, ‘9’$\}$

Predicting Protein Function

$X =$ amino acid sequence, protein family alignments, protein-protein interactions, gene expression profiles, $\ldots$
$Y =$ protein function (metabolism, energy, cell cycle, transcription, $\ldots$)



Prediction Problems

- Data: i.i.d. $(X, Y), (X_1, Y_1), \ldots, (X_n, Y_n)$ from $\mathcal{X} \times \mathcal{Y}$.
- Loss function: $\ell : \mathcal{Y}^2 \to \mathbb{R}^+$, $\ell(\hat{y}, y) =$ cost of mistake.
- Use data $(X_1, Y_1), \ldots, (X_n, Y_n)$ to choose $f : \mathcal{X} \to \mathcal{Y}$ with small risk,

$$R(f) = \mathbf{E}\ell(f(X), Y).$$

  Often choose f from a fixed class $\mathcal{F}$.

Three key issues:

- Approximation error
- Estimation error
- Computation

Prediction Algorithms

1. Kernel methods:
   Choose f from a reproducing kernel Hilbert space, $\mathcal{F}$.
2. Boosting methods:
   Choose f from the span of a fixed dictionary: $\mathcal{F} = \mathrm{span}(\mathcal{G})$.

Both families of prediction algorithms use a function class $\mathcal{F}$ that is

- large (typically infinite-dimensional),
- convex,
- linearly parameterized.

Outline

- Prediction algorithms:
  - Kernel methods
  - Boosting methods
- Estimation error:
  - Complexity of function class
  - Convexity
- Convexity and approximation error

Prediction Algorithms: Kernel Methods

Use a subset $\mathcal{F}$ of a *reproducing kernel Hilbert space* $\mathcal{H}$, with norm $\|\cdot\|_{\mathcal{H}}$.

Choose $f \in \mathcal{F}$ to minimize $\boxed{\underbrace{\mathbf{E}_n \ell(f(X), Y)}_{\text{empirical risk}} + \underbrace{\lambda \|f\|_{\mathcal{H}}^2}_{\text{regularization}},}$

where $\mathbf{E}_n \ell(f(X), Y) = \frac{1}{n} \sum_{i=1}^{n} \ell(f(X_i), Y_i)$.

The regularization term encourages "simple" functions.

Prediction Algorithms: Kernel Methods

Example: Linear kernel, $\mathcal{H}$ = linear functions on $\mathbb{R}^d$.

Then $f \in \mathcal{H}$ has $f(x) = \theta^T x$, and $\|f\|_{\mathcal{H}} = \|\theta\|_2$.

Choose $\theta \in \mathbb{R}^d$ to minimize $\boxed{\underbrace{\mathbf{E}_n \ell(\theta^T X, Y)}_{\text{empirical risk}} + \underbrace{\lambda \|\theta\|^2}_{\text{regularization}}.}$

Prediction Algorithms: Kernel Methods

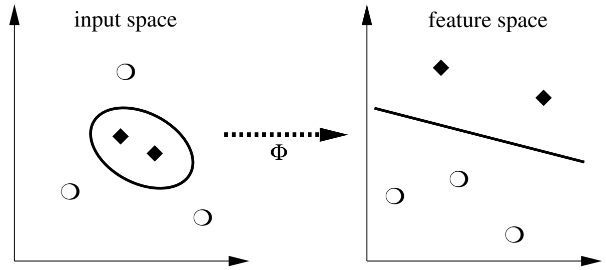
Recall: A reproducing kernel Hilbert space (RKHS) of functions $f : \mathcal{X} \to \mathbb{R}$ is a Hilbert space $\mathcal{H}$ with a "reproducing kernel," $k : \mathcal{X}^2 \to \mathbb{R}$, that is, a symmetric function satisfying

1. $k(x, \cdot) \in \mathcal{H}$,
2. $\langle k(x, \cdot), f \rangle = f(x)$ (the reproducing property)
3. $\overline{\mathrm{span}\{k(x, \cdot) : x \in \mathcal{X}\}} = \mathcal{H}$ (k spans $\mathcal{H}$)

Prediction Algorithms: Kernel Methods

Any kernel k corresponds to an inner product in a feature space,

$$k(a, b) = \langle \Phi(a), \Phi(b) \rangle.$$



picture: (Schölkopf and Smola, 2002)

Prediction Algorithms: Kernel Methods

Example: Linear kernel, $k(a, b) = a^T b$.

Example: Gaussian kernel, $k(a, b) = \exp\left(-\frac{\|a - b\|^2}{2\sigma^2}\right)$.

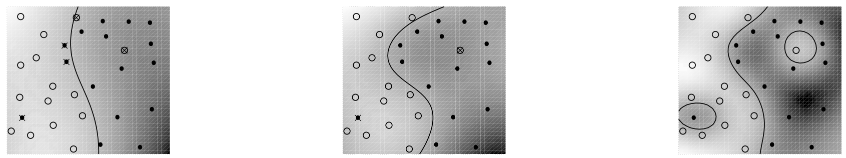
The corresponding RKHS is infinite-dimensional.

We can think of it as the set of linear combinations of Gaussian bump functions.

Prediction Algorithms: Kernel Methods

Choose $f \in \mathcal{F}$ to minimize $\boxed{\underbrace{\mathbf{E}_n \ell(f(X), Y)}_{\text{empirical risk}} + \underbrace{\lambda \|f\|^2_{\mathcal{H}}}_{\text{regularization}}.}$

The regularization term encourages "simple" functions:



picture: (Schölkopf and Smola, 2002)

Kernel Methods: Finite representation

Theorem: For any function $L : \mathbb{R}^n \to \mathbb{R}$ e.g., $(f(X_i)) \mapsto \mathbf{E}_n \ell(f(X), Y)$
and any increasing function $\Omega : \mathbb{R} \to \mathbb{R}$ e.g., *id*
if $f^* \in \mathcal{H}$ minimizes

$$J(f) = L(f(x_1), \ldots, f(x_n)) + \Omega\left(\|f\|_{\mathcal{H}}^2\right)$$

then it can be written

$$f^*(\cdot) = \sum_{i=1}^{n} \alpha_i k(x_i, \cdot)$$

for some $\alpha_1, \ldots, \alpha_n \in \mathbb{R}$.

(Kimeldorf and Wahba, 1971)

Kernel Methods: Finite representation

Because of this result, we can restrict our attention to the Gram matrix, $K \in \mathbb{R}^{n \times n}$,

$$K_{ij} = k(x_i, x_j).$$

Indeed, for $f \in \text{span}\,\{k(x_1, \cdot), \ldots, k(x_n, \cdot)\}$, we have

$$(f(x_1), \ldots, f(x_n))^T = K\alpha$$
$$\|f\|_{\mathcal{H}}^2 = \alpha^T K \alpha,$$

and we can write

$$J(\alpha) = L(K\alpha) + \Omega(\alpha^T K \alpha).$$

Kernel Methods: Finite representation

- We need only compute $k(x_i, x_j)$, we do not need to compute the transformations $\Phi(x_i)$.
- We can have a rich (even infinite-dimensional) $\mathcal{H}$, but we always have a finite-dimensional optimization problem.
- Often (if ℓ is suitably chosen), the solution vector α is sparse.

Kernel Methods: Applications

Information retrieval: discriminating documents

Kernel, evaluated on documents 1 and 2 (d_1, d_2):

$$k(d_1, d_2) = \sum_w \Phi_w(d_1)\Phi_w(d_2),$$

where the sum is over all words w, and $\Phi_w(d_1)$ is the number of times word w appears in document d_1 (appropriately weighted).

In a variety of document categorization tasks, support vector machines with this kernel gave significant improvements over naive Bayes, k-nearest neighbour, Rocchio (a leading IR algorithm), and decision trees. (Joachims, 1998)

Kernel Methods: Applications

Bioinformatics: predicting protein function

Kernel, evaluated on proteins 1 and 2 (p_1, p_2):

$$k(p_1, p_2) = \sum_i \mu_i k_i(p_1, p_2),$$

where the μ_i are convex coefficients and each k_i is a kernel defined using different information about the protein (protein family alignments, protein-protein interactions, genetic interactions, co-participation in protein complexes, gene expression profiles, sequence comparison with known genes).

In predicting the function of yeast proteins, this approach gave significant improvements over standard methods. (Lanckriet et al, 2003)

Outline

- Prediction algorithms:
  - Kernel methods
  - Boosting methods
- Estimation error:
  - Complexity of function class
  - Convexity
- Convexity and approximation error

Prediction Algorithms: Boosting

$\mathcal{F} =$ linear combinations of functions from a fixed dictionary.

History: Schapire (1990) showed that the performance of a *weak* learning algorithm for classification (i.e., slightly better than chance) can be boosted by forming a committee of such classifiers.

Subsequently, a particular boosting algorithm—AdaBoost (Freund and Schapire, 1996)—was found to work well in practice with, e.g., decision trees.

A Generic Boosting Algorithm

Greedy convex optimization over linear combinations

set $f_0 = 0$
for $t = 1, \ldots, T$
  choose $\alpha_t \in \mathbb{R}$ and $g_t \in \mathcal{G}$ to minimize

$$A\left(f_{t-1} + \alpha_t g_t\right).$$

  set $f_t = f_{t-1} + \alpha_t g_t$
return f_T.

(A is a convex objective function).

A Generic Boosting Algorithm

Greedy convex optimization over **convex** combinations

set $f_0 = 0$
for $t = 1, \ldots, T$
  choose $\alpha_t \in [0, 1]$ and $g_t \in \mathcal{G}$ to minimize

$$A\left((1 - \alpha_t) f_{t-1} + \alpha_t g_t\right).$$

  set $f_t = (1 - \alpha_t) f_{t-1} + \alpha_t g_t$
return f_T.

(A is a convex objective function).

AdaBoost

For two-class classification ($Y \in \{\pm 1\}$), AdaBoost uses

$$\mathcal{F} = \mathrm{span}(\mathcal{G}),$$
$$A(f) = \mathbf{E}_n \exp\left(-Y f(X)\right).$$

Regularization:

- small number of steps, or
- constrain size of coefficient vector, $\|\alpha\|_1$.

Boosting Methods: Applications

Natural language processing (parsing, part of speech tagging):
In analysing Wall Street Journal and web text, boosting gives significant accuracy improvements over previous state-of-the-art (maximum entropy) methods. (Collins, 2000, 2002)

AT&T spoken-dialogue system:
Use text from a speech recognition system to identify the type of request, so the call can be appropriately directed. ($\mathcal{G} =$ indicator functions of keywords.) (Rochery et al, 2002)

Outline

- Prediction algorithms:
  - Kernel methods
  - Boosting methods
- Estimation error:
  - Complexity of function class
  - Convexity
- Convexity and approximation error

Estimation Error, Concentration and Complexity

Aim: Use $(X_1, Y_1), \ldots, (X_n, Y_n)$ to choose $f_n \in \mathcal{F}$ to minimize risk, $\mathbf{E}\ell(f_n(X), Y)$.

Estimation error:

$$\mathbf{E}\ell(f_n(X), Y) - \underbrace{\inf_{f \in \mathcal{F}} \mathbf{E}\ell(f(X), Y)}_{\text{Best possible in } \mathcal{F}}$$

Depends on:

- Sample size, n
- Complexity of $\mathcal{F}$
- Algorithm (in particular, regularization)

Estimation Error

Bounds on estimation error

- show how to measure complexity, and hence
- motivate regularization schemes

For example, if we know that, uniformly over $f \in \mathcal{F}$,

$$\mathbf{E}\ell(f(X), Y) \leq \mathbf{E}_n\ell(f(X), Y) + c_n\Omega(f),$$

then minimizing $\mathbf{E}_n\ell(f(X), Y) + c_n\Omega(f)$ corresponds to obtaining the best upper bound.

Can give performance guarantees ('oracle inequalities') for *complexity regularization* schemes of this kind.

Rademacher Averages

Definition: For sample size n, the Rademacher average of $\mathcal{F}$ is

$$R_n(\mathcal{F}) = \mathbf{E} \sup_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \epsilon_i f(X_i),$$

where $\epsilon_1, \ldots, \epsilon_n$ are independent uniform $\{\pm 1\}$ (Rademacher) random variables.

Intuition: Measures how well some $f \in \mathcal{F}$ can match the random direction $(\epsilon_1, \ldots, \epsilon_n) \in \{\pm 1\}^n$.

Estimation Error and Rademacher Averages

Theorem: For $\mathcal{F} \subseteq [0,1]^{\mathcal{X}}$ and for any P on $\mathcal{X}$, with probability at least $1 - e^{-x}$ over $X_1, \ldots, X_n$,

$$\sup_{f \in \mathcal{F}} (\mathbf{E}f - \mathbf{E}_n f) \leq 2R_n(\mathcal{F}) + \sqrt{\frac{x}{2n}}.$$

(Proof: Concentration of $\sup_{f \in \mathcal{F}} (\mathbf{E}f - \mathbf{E}_n f)$ about its expectation; symmetrization.)

Estimation Error and Rademacher Averages

We are usually interested in loss classes,

$$\ell_{\mathcal{F}} = \{\ell_f : f \in \mathcal{F}\} \qquad \text{with} \qquad \ell_f(x, y) = \ell(f(x), y).$$

Then $\mathbf{E}\ell_f$ is the *risk*, and $\mathbf{E}_n\ell_f$ is the *empirical risk*.

For an Lipschitz loss ℓ (with Lipschitz constant L), we have

$$R_n(\ell_{\mathcal{F}}) \le L R_n(\mathcal{F}),$$

so we can restrict attention to $R_n(\mathcal{F})$.

Estimation Error and Rademacher Averages

For kernel classes, $\mathcal{F} = \{f \in \mathcal{H} : \|f\|_{\mathcal{H}} \le \lambda\}$,

$$R_n(\mathcal{F}) \le \lambda \frac{\mathbf{E}\sqrt{\mathrm{trace}(K)}}{n},$$

where K is the Gram matrix (inner products between training data pairs).

Thus, the *empirical risk minimizer* $\hat{f} = \arg\min_{f \in \mathcal{F}} \mathbf{E}_n \ell_f$ satisfies

$$\mathbf{E}\ell_{\hat{f}} - \inf_{f \in \mathcal{F}} \mathbf{E}\ell_f \le 2L\lambda \frac{\mathbf{E}\sqrt{\mathrm{trace}(K)}}{n} + \sqrt{\frac{c}{n}}.$$

(There is an analogous data-dependent bound that holds with high probability over the data.)

(Bartlett, Bousquet, Mendelson, 2002)

Estimation Error and Rademacher Averages

For boosting algorithms, with

$$\mathcal{F} = \left\{ \sum_{i=1}^{k} \alpha_i g_i : \|\alpha\|_1 \le \lambda,\ g_i \in \mathcal{G} \right\},$$

we have

$$R_n(\mathcal{F}) = \lambda R_n(\mathcal{G}).$$

(Koltchinskii and Panchenko, 2002)

Estimation Error and Rademacher Averages

For sigmoid neural networks, with

$$\mathcal{F} = \left\{ x \mapsto \sum_{i=1}^{k} w_i \sigma(v_i^T x) : \|w\|_1 \le B,\ \|v_i\|_1 \le B \right\},$$

where $\sigma : \mathbb{R} \to [-1, 1]$ is Lipschitz (with constant 1),

$$R_n(\mathcal{F}) \le B^2 \sqrt{\frac{2\log(\dim(\mathcal{X}))}{n}}.$$

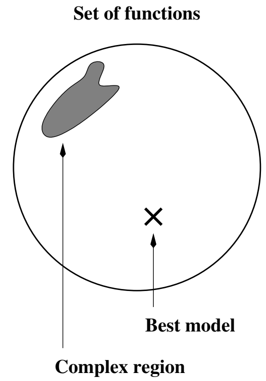
Outline

- Prediction algorithms:
  - Kernel methods
  - Boosting methods
- Estimation error:
  - Complexity of function class
  - Convexity
- Convexity and approximation error

Convexity and Estimation Error

A bound on $\sup_{f \in \mathcal{F}} (\mathbf{E}\ell_f - \mathbf{E}_n \ell_f)$ ensures that functions with small empirical risk ($\mathbf{E}_n \ell_f$) have small risk ($\mathbf{E}\ell_f$). But this can be conservative.

Some part of the space of functions might be complex, but if it is far from the functions of interest (ie: those that fit the data well), it is unlikely to be important.



Convexity and Estimation Error

Consider instead

$$\begin{aligned} \mathcal{L}_{\mathcal{F}} &= \{\ell_f - \ell_{f^*} : f \in \mathcal{F}\} \\ &= \{(x, y) \mapsto \ell(f(x), y) - \ell(f^*(x), y) : f \in \mathcal{F}\}, \end{aligned}$$

where $f^* = \arg\min_{f \in \mathcal{F}} \mathbf{E}\ell_f$.

If

- ℓ is Lipschitz, uniformly convex (say, has quadratic modulus of convexity)
- $\mathcal{F}$ is convex,

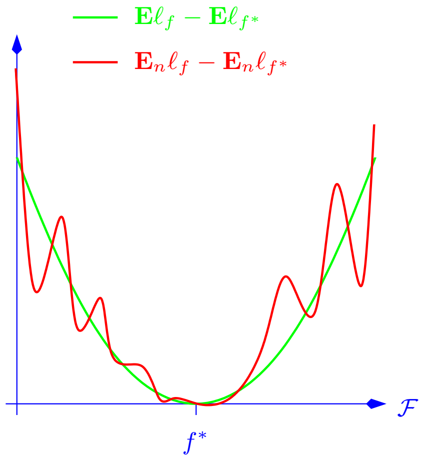
then for all $f \in \mathcal{F}$, $\quad \mathrm{var}(\ell_f - \ell_{f^*}) \leq c\mathbf{E}(\ell_f - \ell_{f^*})$.

(Bartlett, Jordan, McAuliffe, 2003)

Convexity and Estimation Error

Even if $\mathbf{E}_n \ell_f$ fluctuates a lot, if $\mathbf{E}\ell_f$ is close to the minimal $\mathbf{E}\ell_{f^*}$, the fluctuations of $\mathbf{E}_n \ell_f$ and $\mathbf{E}_n \ell_{f^*}$ are strongly correlated.

Thus, $\hat{f} = \arg\min_{f \in \mathcal{F}} \mathbf{E}_n \ell_f$ has risk $\mathbf{E}\ell_{\hat{f}}$ converging quickly to $\mathbf{E}\ell_{f^*}$.



Convexity and Estimation Error

Theorem: For

- bounded, Lipschitz ℓ with modulus of convexity $\delta(\epsilon) \geq c\epsilon^2$,
- convex $\mathcal{F}$,

if $\hat{f} \in \mathcal{F}$ minimizes $\mathbf{E}_n \ell_f$, then

$$\mathbf{E}\ell_{\hat{f}} \leq \mathbf{E}\ell_{f^*} + \epsilon^* + \frac{c}{n},$$

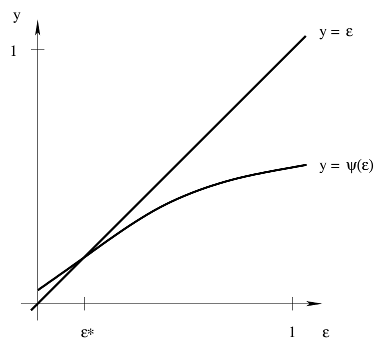
provided

$$\epsilon^* \geq c \underbrace{R_n \{f \in \mathcal{F} : \mathbf{E}\ell_f \leq \mathbf{E}\ell_{f^*} + \epsilon^*\}}_{\text{local Rademacher averages}} = \Psi(\epsilon^*).$$

(Bartlett, Bousquet, Mendelson, 2002); see also (Koltchinskii and Panchenko, 2000), (Massart, 2000), (Lugosi and Wegkamp, 2003)

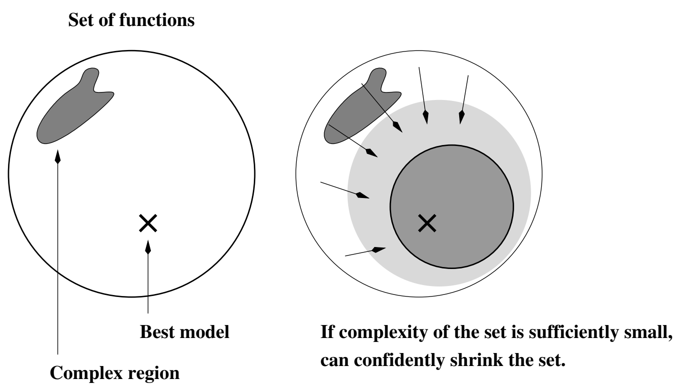
Convexity and Estimation Error

The local Rademacher averages can be *much* smaller than the Rademacher averages of the full class ($\Psi(1)$).



Convexity and Estimation Error

The intuition:



Convexity and Estimation Error

For kernel classes, $\mathcal{F} = \{f \in \mathcal{H} : \|f\|_{\mathcal{H}} \leq \lambda\}$, we can choose

$$\epsilon^* \geq c\lambda \sqrt{\frac{\sum_{i=1}^{n} \min(\mu_i/n, \epsilon^*)}{n}},$$

where μ_i are the eigenvalues of the Gram matrix K (the matrix of inner products between training data pairs).

For rapidly decreasing eigenvalues (e.g., Gaussian kernel), this gives a faster convergence rate.

Convexity and Estimation Error

For boosting algorithms, with

$$\mathcal{F} = \left\{ \sum_{i=1}^{k} \alpha_i g_i : \|\alpha\|_1 \le \lambda,\ g_i \in \mathcal{G} \right\}$$

and $\dim(\mathcal{G}) = d$, we can choose

$$\epsilon^* = c(\lambda + 1) n^{-(d+2)/(2d+2)}.$$

Again, we get a faster rate of convergence of $\mathbf{E}\ell_{\hat{f}}$ to $\mathbf{E}\ell_{f^*}$.

Convexity and Estimation Error

Summary:

If
- ℓ is strictly convex,
- $\mathcal{F}$ is convex,

then $\mathbf{E}\ell_{\hat{f}} - \mathbf{E}\ell_{f^*}$ decreases rapidly,

where $\hat{f} = \arg\min_{f\in\mathcal{F}} \mathbf{E}_n \ell_f$, and $f^* = \arg\min_{f\in\mathcal{F}} \mathbf{E}\ell_f$.

In particular, the rate can be faster than $n^{-1/2}$ under these conditions.

There are converse results.

Convexity and Estimation Error: Converse Result

Theorem: If
- $\ell(\hat{y}, y) = (\hat{y} - y)^2$, and
- $\mathcal{F}$ is *not convex* (i.e., for some $P_{\mathcal{X}}$ on $\mathcal{X}$, the closure of $\mathcal{F}$ in $L_2(P_{\mathcal{X}})$ is not convex)

then
- for some $B > 0$,
- for every algorithm for choosing $\hat{f}$
- for every n,

there is a probability distribution P on $\mathcal{X} \times [-B, B]$ with

$$\mathbf{E}\ell_{\hat{f}} - \mathbf{E}\ell_{f^*} \ge \frac{c}{\sqrt{n}}.$$

(Lee, Bartlett, Williamson, 1998)

Outline

- Prediction algorithms:
  - Kernel methods
  - Boosting methods
- Estimation error:
  - Complexity of function class
  - Convexity
- Convexity and approximation error

Convexity and Approximation

Thus, if ℓ is convex, $\mathcal{F}$ is non-convex, then replacing $\mathcal{F}$ with $\mathrm{co}(\mathcal{F})$ leads to

- Computational cost never much worse.
- Approximation error at least as good.
- Estimation error not much worse (e.g., ℓ quadratic, $\mathcal{F}$ simple).

What about a non-convex cost function ℓ?

Large Margin Classifiers

- In two-class classification ($y \in \{\pm 1\}$), we have

$$\ell(\hat{y}, y) = \mathbf{1}\left[\mathrm{sign}(\hat{y}) \neq y\right].$$

- Minimizing empirical risk is often intractable.
- We replace the 0-1 loss, ℓ, with a convex surrogate, ϕ.
- Can efficiently minimize empirical average over many natural function classes.
- This is the approach used by AdaBoost, support vector machines, logistic regression, sigmoid neural networks, ...
- What is the impact of this computational convenience?

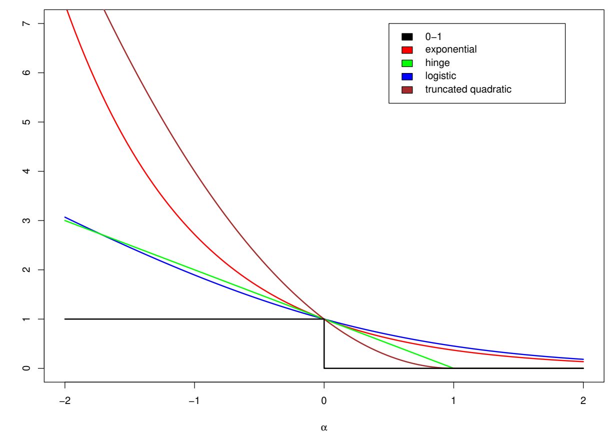
Large Margin Classifiers

- Consider the margins, $Yf(X)$.
- Define a margin cost function $\phi : \mathbb{R} \to \mathbb{R}^+$.
- Define the **ϕ-risk** of $f : \mathcal{X} \to \mathbb{R}$ as $R_\phi(f) = \mathbf{E}\phi(Yf(X))$.
- Choose $f \in \mathcal{F}$ to minimize ϕ-risk.
  (e.g., use data, $(X_1, Y_1), \ldots, (X_n, Y_n)$, to minimize **empirical ϕ-risk**,

$$\hat{R}_\phi(f) = \hat{\mathbf{E}}\phi(Yf(X)) = \frac{1}{n}\sum_{i=1}^{n}\phi(Y_i f(X_i)),$$

  or a regularized version.)

Large Margin Classifiers



Consequences of Using a Convex Cost

$$R(f) = \Pr\left(\text{sign}(f(X)) \neq Y\right) \qquad R^* = \inf_f R(f) \qquad \text{(Bayes risk)}$$

$$R_\phi(f) = \mathbf{E}\phi(Yf(X)) \qquad R^*_\phi = \inf_f R_\phi(f) \qquad \text{(optimal } \phi\text{-risk)}$$

Theorem: For any ϕ, there is a function ψ such that for all P and f,

$$\psi(R(f) - R^*) \leq R_\phi(f) - R^*_\phi.$$

(Bartlett, Jordan, McAuliffe, 2003)

Consequences of Using a Convex Cost

For convex ϕ, the ψ-transform is given by

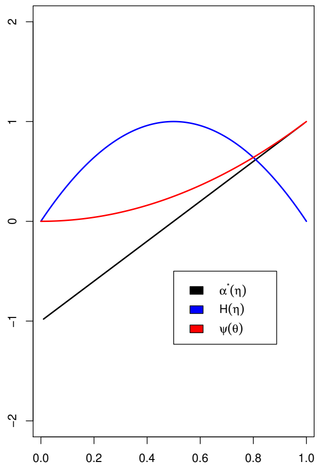
$$\psi(\theta) = \inf_{\alpha \leq 0}\left(\frac{1+\theta}{2}\phi(\alpha) + \frac{1-\theta}{2}\phi(-\alpha)\right) - \inf_{\alpha}\left(\frac{1+\theta}{2}\phi(\alpha) + \frac{1-\theta}{2}\phi(-\alpha)\right).$$

- This gives the best possible upper bound (it cannot be improved anywhere).
- For convex ϕ, $R_\phi(f_n) \to R^*_\phi$ always implies $R(f_n) \to R^*$ if and only if $\phi'(0)$ exists and is negative.

ψ-transform: Example

$$\psi(\theta) = \inf_{\alpha \leq 0}\left(\frac{1+\theta}{2}\phi(\alpha) + \frac{1-\theta}{2}\phi(-\alpha)\right) - \inf_{\alpha}\left(\frac{1+\theta}{2}\phi(\alpha) + \frac{1-\theta}{2}\phi(-\alpha)\right).$$

e.g., two-norm soft-margin SVM:
$\phi(\alpha) = (\max(1-\alpha, 0))^2$.
first term = 1.
second term = $H((1+\theta)/2)$, with
$H(\eta) = 4\eta(1-\eta)$.



Complexity, Concentration, and Convexity

- Prediction algorithms:
  Choose $\hat{f} \in \mathcal{F}$ to minimize empirical risk, $\mathbf{E}_n \ell_f$.
  - Kernel methods
  - Boosting methods

  Convex, infinite-dimensional $\mathcal{F}$; convex ℓ.
- Estimation error: $\mathbf{E}\ell_{\hat{f}} - \inf_{f \in \mathcal{F}} \mathbf{E}\ell_f$.
  - Bound using Rademacher averages
  - Implies complexity penalty:
    - norm in RKHS, kernel matrix eigenvalues,
    - norm of boosting coefficients.

Complexity, Concentration, and Convexity

- If $\mathcal{F}$ and ℓ are convex, local Rademacher averages give better error estimates.
- Benefits of replacing a nonconvex $\mathcal{F}$ with its convex hull.
- Effect of replacing a nonconvex ℓ with a convex surrogate: large margin classifiers.