Loose Ends: stability, various definition

Tomaso Poggio

9.520 Class 13

March 2010

A reminder: convergence in probability

Let $\{X_n\}$ be a sequence of bounded random variables. We say that

$$\lim_{n\to\infty} X_n = X \text{ in probability}$$

if

$$\forall \varepsilon > 0 \lim_{n\to\infty} \mathbb{P}\{|X_n - X| \geq \varepsilon\} = 0.$$

or
if for each n there exists a ε_n and a δ_n such that

$$\mathbb{P}\{|X_n - X| \geq \varepsilon_n\} \leq \delta_n,$$

with ε_n and δ_n going to zero for $n \to \infty$.

Generalization

A natural requirement for f_S is distribution independent **generalization**

$$\forall \mu, \lim_{n\to\infty} |I_S[f_S] - I[f_S]| = 0 \text{ in probability}$$

This is equivalent to saying that for each n there exists a ε_n and a δ_n such that $\forall \mu$

$$\mathbb{P}\{|I_{S_n}[f_{S_n}] - I[f_{S_n}]| \geq \varepsilon_n\} \leq \delta_n,$$

with ε_n and δ_n going to zero for $n \to \infty$.
In other words, the training error for the solution must converge to the expected error and thus be a "proxy" for it. Otherwise the solution would not be "predictive".
A desirable additional requirement is **universal consistency**

$$\forall \varepsilon > 0 \lim_{n\to\infty} \sup_{\mu} \mathbb{P}_S \left\{ I[f_S] > \inf_{f\in\mathcal{H}} I[f] + \varepsilon \right\} = 0.$$

Uniform Stability

Let us recall **notation:** S training set, $S^{i,z}$ training set obtained replacing the i-th example in S with a new point $z = (x, y)$.

Definition

We say that an algorithm $\mathcal{A}$ has **uniform stability** β (is β-stable) if

$$\forall (S, z) \in \mathcal{Z}^{n+1}, \ \forall i, \ \forall z' \ \sup_{z' \in Z} |V(f_S, z') - V(f_{S^{i,z}}, z')| \leq \beta.$$

Remarks: Uniform Stability

Uniform stability is a strong requirement: a solution has to change very little even when a very unlikely training set is drawn.

the coefficient β is a function of n, and should perhaps be written β_n.

CV_{loo} Stability

We first introduce the definition of *Cross-Validation leave-one-out stability*. **Definition:** *The learning map L is distribution-independent, CV_{loo} stable if uniformly for all probability distributions μ*

$$\lim_{n\to\infty} \sup_{i\in\{1,\ldots,n\}} |V(f_{S^i}, z_i) - V(f_S, z_i)| = 0 \quad \text{in probability},$$

where S^i denotes the training set S with the ith point removed.
CV_{loo} stability measures the difference in errors at a point z_i between a function obtained given the entire training set and one obtained given the same training set but with the point z_i left out

Theorem A: *For good loss functions the following statements are equivalent for ERM:*

- L is distribution-independent CV_{loo} stable
- ERM generalizes and is universally consistent
- $\mathcal{H}$ is uniform Glivenko-Cantelli.

Remarks: CV_{loo} Stability

CV_{loo} stability is weaker than uniform stability because a) it is in probability and b) it is true for z_i not for an arbitrary z.

the definition of stability is about difference of the error on a training point and the error on the same test point going to zero: it seems plausible that this may imply generalization.

it turns out that with some additional technical conditions CV_{loo} stability implies generalization independently of ERM.

Loose Ends: online stability

Tomaso Poggio

March 2010

Batch learning algorithms

- We consider sequentially independent and identically drawn samples from the distribution on Z. The training set S consists of n samples:

$$S = \{z_1 = (x_1, y_1), ..., z_n = (x_n, y_n)\}.$$

- The *expected error* of of a function f is defined as

$$I[f] = \int_Z V(f, z) d\mu(z) = \mathbb{E}_z V(f, z),$$

which is also the expected error of a new sample z drawn from the distribution.

- The following quantity, called *empirical error*, can be computed by a "batch" learning algorithm, given all the training data S

$$I_S[f] = \frac{1}{n} \sum_{i=1}^{n} V(f, z_i).$$

Online learning

Algorithms here take as inputs a hypothesis $f \in \mathcal{H}$ and a new example $z = x, y$ and return a new hypothesis $f' \in \mathcal{H}$. Given an input sequence $S \in Z^n$ with $S = z_1, \cdots, z_n$, the online algorithm will use z_1 and the zero hypothesis f_0 to generate the first hypothesis f_1. After seeing the whole Z^n sequence the algorithm has generated a sequence of hypothesis $f_0, \cdots, f_n$ and has "memory" only of the last example z_n.

Online learning algorithms

- We define as *training error* of an online algorithm at iteration n

$$V(f_n, z_n)$$

where the algorithm generates f_n from f_{n-1} *after* "seeing" z_n.

- We define as *average training error* of an online algorithm at iteration n

$$I^n_{emp} = \frac{1}{n} \sum_i^n V(f_i, z_i)$$

where the algorithm generates f_i from f_{i-1} *after* "seeing" z_i.

The notion of generalization is not appropriate for online algorithms

An algorithm is said to *generalize* if the function f_S selected by it satisfies for all S ($|S| = n$) and for any probability distribution μ

$$\lim_{n\to\infty} |I[f_S] - I_S[f_S]| = 0 \text{ in probability.}$$

For an online algorithm that "forgets" past data, it is not natural to define the empirical error. Generalization is *not* a natural concept for online algorithms. Consistency is meaningful for online algorithms. We recall that an algorithm is (universally) consistent if for any distribution μ and any $\varepsilon > 0$

$$\lim_{n\to\infty} \mathbb{P}\left\{ I[f_S] > \inf_{f\in\mathcal{H}} I[f] + \varepsilon \right\} = 0.$$

A class project: can stability be at the core of online learning?

CV-like:

$$\epsilon_n < [-V(f_{n+1}, z_{n+1}) + V(f_n, z_{n+1})] \leq \chi_n$$

- Notice that $V(f_n, z_{n+1})$ is the out-of-sample-error since f_n does not depend on z_{n+1} whereas $V(f_n, z_n)$ is the in-sample-error since f_n depends on z_n (and f_{n-1}). Notice that f_n *depends* on z_n: thus in $[V(f_{n+1}, z_{n+1})]$ the hypothesis f_{n+1} is a function of z_{n+1} (and of f_{n+1}). Thus this is a condition on the *cross-validation* error.
- The upper-bound above is key. It makes sure that the update of the hypothesis decreases the error on the new data point (relative to the error on that point made by the previous hypothesis that was formulated before "seeing" that point) – but *not too much*. Intuitively it guarantees that overfitting cannot occur.

A class project: can stability be at the core of online learning?

Notice that online regularization (which satisfies the condition above) ensures that $Regret = o(T)$ and this in turn ensures consistency of the online learning (Rakhlin, pers. comm.).

Conjecture *The CV-like condition is sufficient for consistency of online learning.*

Remark

If the conjecture is true, one could have algorithms which use directly stability (though they would be similar to the special case of online regularization). This may be especially interesting for biological implementations of online RL.

A note about consistency of online algorithms

For an intuition of why we need $\sum \gamma_n = \infty$ consider the differential equation $\frac{dx}{dt} + \gamma(t)x = 0$ with solution $x(t) = x_0 e^{-\int \gamma(t)dt}$. It is possible to show that the condition $\int \gamma(t)dt \to \infty$ corresponds to $\sum \gamma_n = \infty$. Conditions of this type are needed for convergence to the minimum. Consider now $\frac{dx}{dt} + \gamma(t)(x + n(t)) = 0$: we need $\gamma(t)n(t) \to 0$ to eliminate the effect of the "noise" $n(t)$, implying at least $\gamma_n \to 0$. This condition corresponds to *c-stability* which has a different motivation (generalization).

Loose Ends...

Lorenzo Rosasco

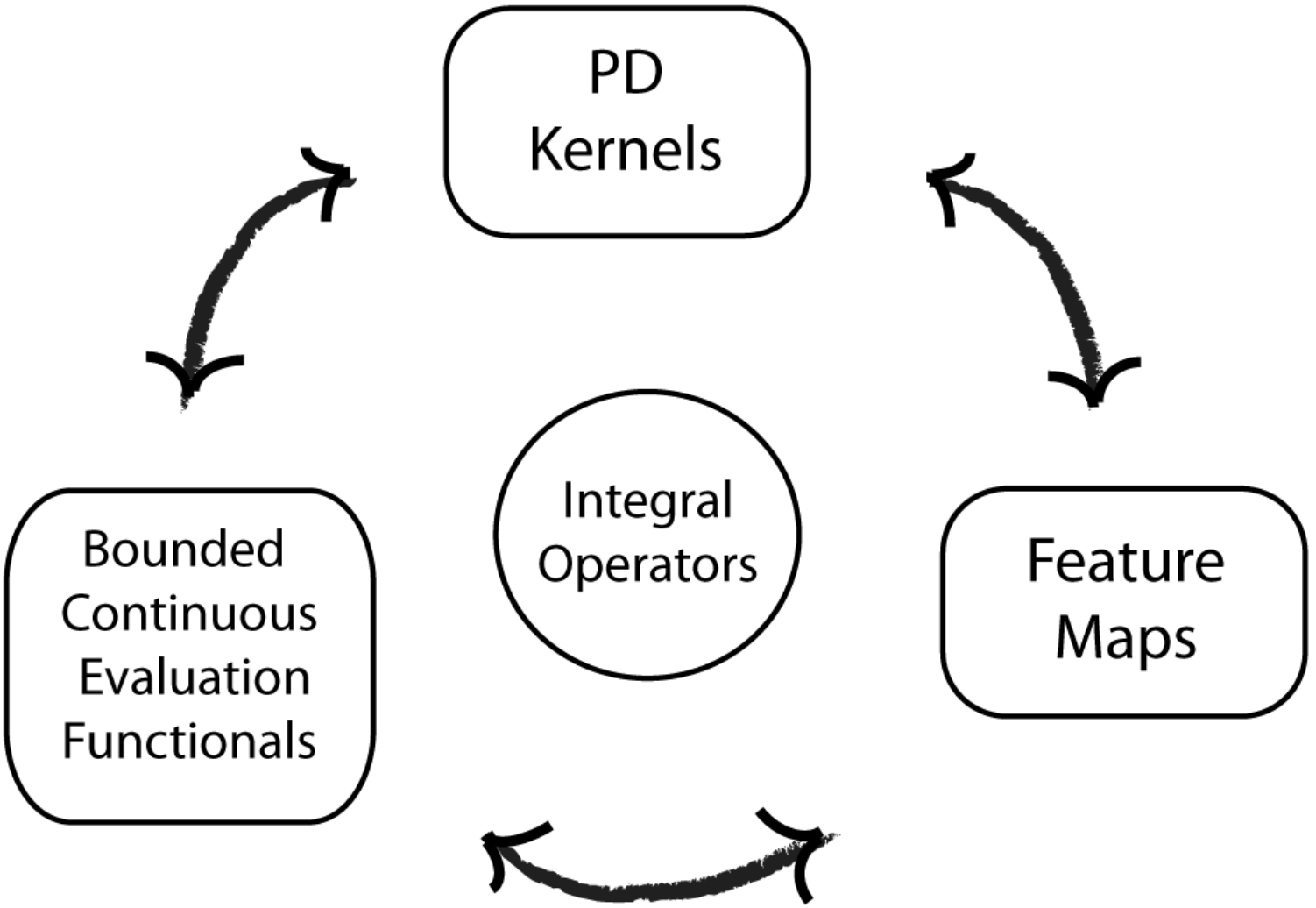
9.520 Class 13

March 17, 2009

Plan

- Mercer Theorem
- Elastic Net

Some Other Facts on RKH Spaces



Integral Operator

RKH space can be characterized using the integral operator

$$L_K f(s) = \int_X K(x,s) f(x) p(x) dx$$

where $p(x)$ is the probability density on X.

The operator has domain and range in $L^2(X, p(x)dx)$ the space of functions $f : X \to \mathbb{R}$ such that

$$\langle f, f \rangle_2 = \int_X |f(x)|^2 p(x) dx < \infty$$

Mercer Theorem

If X is a compact subset in $\mathbb{R}^d$ and K continuous, symmetric (and PD) then L_K is a **compact, positive** and **self-adjoint** operator.

- There is a decreasing sequence $(\sigma_i)_i \geq 0$ such that $\lim_{i\to\infty} \sigma_i = 0$ and

$$L_K \phi_i(x) = \int_X K(x,s)\phi_i(s)p(s)ds = \sigma_i \phi_i(x),$$

  where ϕ_i is an orthonormal basis in $L^2(X, p(x)dx)$.
- The action of L_K can be written as

$$L_K f = \sum_{i\geq 1} \sigma_i \langle f, \phi_i \rangle_2 \phi_i.$$

Mercer Theorem (cont.)

- The kernel function have the following representation

$$K(x, s) = \sum_{i \geq 1} \sigma_i \phi_i(x) \phi_i(s).$$

A symmetric, positive definite *and* continuous Kernel is called a *Mercer* kernel.

- The above decomposition allows to look at the kernel as a dot product in some *feature space*.

Different Definition of RKHS

It is possible to prove that:

- $$\mathcal{H} = \{f \in L^2(X, p(x)dx) | \sum_{i \geq 1} \frac{\langle f, \phi_i \rangle_2^2}{\sigma_i} < \infty\}.$$

- The scalar product in $\mathcal{H}$ is

$$\langle f, g \rangle_{\mathcal{H}} = \sum_{i \geq 1} \frac{\langle f, \phi_i \rangle_2 \langle g, \phi_i \rangle_2}{\sigma_i}.$$

A different proof of the representer theorem can be given using Mercer theorem.

Plan

- Mercer Theorem
- Elastic Net

Some Remarks

$$\min_{\beta \in \mathbb{R}^p} \|Y - \beta X\|^2 + \lambda \|\beta\|_1 .$$

- **About Uniqueness**: the solution of ℓ_1 regularization is not unique. Note that the various solution have the **same prediction properties** but **different selection properties**.
- **Correlated Variables**: If we have a group of correlated variables the algorithm is going to select just one of them. This can be bad for interpretability but maybe good for compression.

Elastic Net Regularization

One possible way to cope with the previous problems is to consider

$$\min_{\beta \in \mathbb{R}^p} \|Y - \beta X\|^2 + \lambda(\alpha \|\beta\|_1 + (1-\alpha)\|\beta\|_2^2).$$

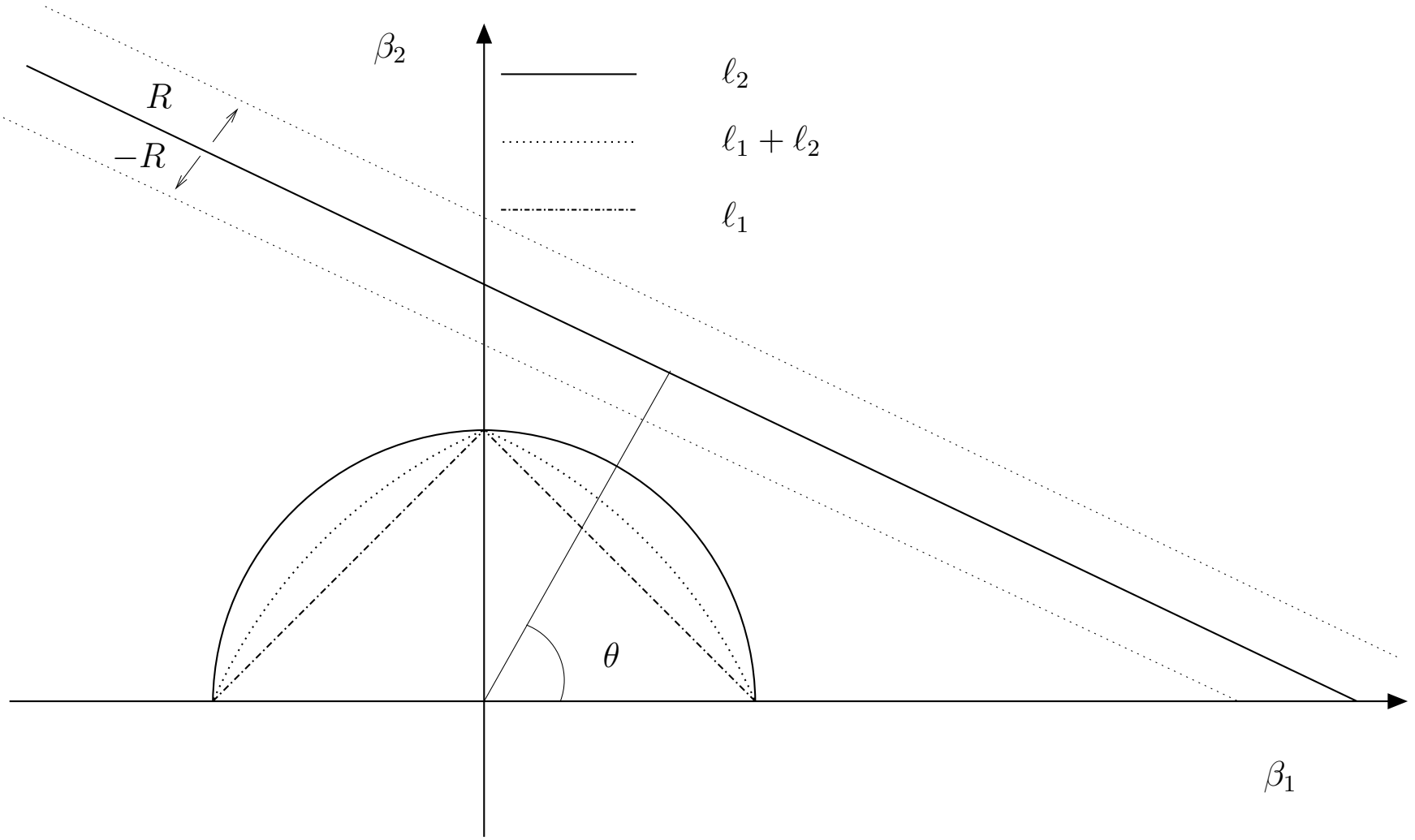
- λ is the regularization parameter.
- α controls the amount of sparsity and correlation.

(Zhu. Hastie '05; De Mol, De Vito, Rosasco '07)

Elastic Net Regularization (cont.)

- The ℓ_1 term promotes sparsity and the ℓ_2 term smoothness.
- The functional is strictly convex: the solution is unique.
- A whole group of correlated variables is selected rather than just one variable in the group.

Geometry of the Problem



ℓ_q regularization?

Consider a more general penalty of the form

$$\|\beta\|_q = (\sum_{i=1}^{p} |\beta^i|^q)^{1/q}$$

(called bridge regression in statistics).
It can be proved that:

- $\lim_{q\to 0} \|\beta\|_q \to \|\beta\|_0$,
- for $0 < q < 1$ the norm is **not** a convex map,
- for $q = 1$ the norm **is** a convex map and is **strictly** convex for $q > 1$.

Caveat

Learning algorithms based on sparsity usually suffer from an excessive shrinkage effect of the coefficients.
For this reason in practice a two-step procedure is usually used:

- Use Lasso (or Elastic Net) to select the relevant components
- Use ordinary least squares (in fact usually Tikhonov with λ small...) on the selected variables.