2007 projects (*if good results then conference paper*)

Pr. 1.1 Experiment with RLS, measuring training, testing and LOO performance as a function of λ, σ on UCI data and possibly synthetic data (to be able to compare to expected error) [RIF]

Pr 1.3 Implement and test a "large-scale" nonlinear RLS, using and expanding the ideas discussed in class. [RIF]

Pr 1.4 When does overfitting occur? Describe some specific examples (such as the leukemia data). Analyze. Could learning a kernel overfit? Do theoretical analysis (difficult) or empirical experiments. [RIF+TP]

Pr 1.5 (a) Why Reproducing Kernel Hilbert Spaces are a natural set of hypothesis spaces for supervised learning? Give a general, mathematical argument.

(b) Use Sobolev emebedding lemma to argue that the requirement above implies smoothness of the h ypothesis space

Pr 1.6 Analyze theoretically uniform stability for Gaussian kernels as a function of λ, σ. [RIF]

Pr 1.7 Laplacian RLS: stability wrt unlabeled data, performance as a function of n, number of unlabeled data points. [Lorenzo, RIF]

2007: other projects

1. Sparsity: representation (or reconstruction or interpretation) and generalization? Is sparsity "good" for learning, namely for generalization? Study the connections between sparsity and generalization using the tools of stability. Alternatively describe possible connections between a) sparse representations, b) information theory (compression, see Warmuth paper) c) capacity constraints on hypothesis space (VC-dimension etc.) d) NMF [JAKE, Lorenzo,TP]

2. Describe learning of parameters (possibly including dimensionality reduction using Laplacian eigenfunctions and clustering) for a stochastic dynamical system using the framework of Coiffman's diffusion maps.

2007: other projects

1. Derive and test in simulations data-mining bound: Roughly speaking, if you try k models on l (validation set) points, then for all k models uniformly with probability $1 - \eta$: TestError $\leq$ ValidationError $+ \frac{\sqrt{((log(K) - log(\eta))}}{l)}$ Clearly this holds also for the one of the k machines you choose (one would choose the one with the smallest validation error).

2. Discuss ideas for algorithms based on maximizing stability of the algorithm at the predicted point and minimizing empirical error [TP]

3. (suggested by Steve Smale) Approximate indicator functions with kernels from a RKHS with very little smoothness. Calculate approx and sample error using bounds such as Cucker Smale etc.. Verify with computer simulations. [TP+RIF]

4. Exploit fast algoritm for NN (Indyk's algorithms, improved fast Gauss transforms ....) [RIF]

5. Variable selection. Measure effect of noise variables... [RIF,JAKE?]

2007: Computational Neuroscience-type projects

1. Describe a few plausible neural circuit for Gaussian tuning

2. Do specific experiments in object recognition (eg on Catlech 256,...) with the model described in Class 18, 19 and in Serre, T., L. Wolf, S. Bileschi, M. Riesenhuber and T. Poggio. Object Recognition with Cortex-like Mechanisms, IEEE Transactions on Pattern Analysis and Machine Intelligence, 29, 3, 411-426, 2007 (software available on the Web). We may provide the output of the top level of the model (before the classifier) and you may try different classifiers and different supervised and semisupervised schemes. [Serre, tp]

3. Use the software provided by Jim Mutch to experiment with biologically inspired systems for object recognition by changing parameters in the basic implementation above (eg number of layers, form of the 2 basic functions, etc.) [Mutch, tp]

2007: Review-type projects

1. Review: Vector valued RKHS: Micchelli and Pontil, describe applications

2. Review: recent approaches to prediction of time series (advice: avoid financial time series). Review approaches based on combination of classifiers for time series prediction – such as mixture of Gaussians (see Gerschefeld Nature paper, January 28, 1999)

3. Review: techniques to transform a variable length input vector into a fixed length one. What is an acceptable set of measurements? Consider in particular time series.

4. Review: Core Vector Machines: review very fast SVM-like algorithms.

5. Review/analyze use of GPUs for SVM implementations.