# From Last Meeting

Studied Polynomial Embedding

- Toy examples
- Generalized to “kernels” (e.g. Gaussian)
- Big gains in “flexiblity”
- Magnified High Dimension Low Sample Size problems
- Motivated Support Vector Machines

# Support Vector Machines

Classical References:

Vapnik (1982) *Estimation of dependences based on empirical data*, Springer (Russian version, 1979)

Boser, Guyon & Vapnik (1992) in *Fifth Annual Workshop on Computational Learning Theory*, ACM.

Vapnik (1995) *The nature of statistical learning theory*, Springer.

Recommended tutorial:

Burges (1998) A tutorial on support vector machines for pattern recognition, *Data Mining and Knowledge Discovery*, **2**, 955-974, see also web site:
http://citeseer.nj.nec.com/burges98tutorial.html

# Support Vector Machines (cont.)

Motivation: High Dimension Low Sample Size discrimination

(e.g. from doing a nonlinear embedding)

$\exists$ a tendency towards major *over-fitting* problems

Toy Example:

In 1st dimension: Class 1: $N(2,0.8)$ Class 2: $N(-2,0.8)$

($n = 20$ of each, and threw in 4 "outliers")

In dimensions $2,...,d$: independent $N(0,1)$

Show Svm\SVMeg3p1d2m1v1.mpg

# Support Vector Machines (cont.)

Main Goal of Support Vector Machines:

Achieve a trade off between:

Discrimination quality for data at hand

vs.

Reproducibility with new data

Approaches:

1. Regularization (bound on “generaliz’n”, via “complexity”)
2. Quadratic Programming (general’n of Linear Prog.)

# Support Vector Machines (cont.)

Heuristic and Graphical Introduction:

(see Burges paper for detailed math's, and optimization)

Goal: find a hyperplane that "best separates" the data classes

Recall hyperplane in $\Re^d$ is parameterized by a "normal vector" $\underrightarrow{w}$ and an "intercept" $b$

$$\left\{\underrightarrow{x} : \left\langle \underrightarrow{x}, \underrightarrow{w} \right\rangle = b\right\}$$

# Support Vector Machines (cont.)

Case 1: “separable data”,

i.e. data can be separated by a hyperplane

Then find $\underrightarrow{w}$ and $b$ to maximize the “margin” between classes:

Show Burges98TutFig5.jpg

Statistical weakness: “driven” by just a few points

Show SVM\SVMeg2p4v1.mpg

# Support Vector Machines (cont.)

Case 2: “nonseparable data”

Then add a penalty, parametrized by $C$, to:
“points on wrong side of plane”

Show Burges98TutFig6.jpg

Solve resulting problem by quadratic programming methods

# Support Vector Machines (cont.)

Implementation: Matlab code from:

http://www.isis.ecs.soton.ac.uk/resources/svminfo/

(Caution: must use class labels $\pm 1$)

Many others web available, e.g. see:

http://www.kernel-machines.org/software.html

# Support Vector Machines (cont.)

Choice of $C$: very important

Show SVM\SVMeg2p2v1.mpg (start at log10(C) =10)

- less weight on “wrong side” points for smaller $C$
- “regresses” for bigger $C$????
- “jump” at $\log_{10} C = -4$????

# Support Vector Machines (cont.)

Choice of $C$: "regularization view"

Simpler context: :Smoothing Splines

Fit a curve $f(x)$ to data $(X_1, Y_1), \ldots, (X_n, Y_n)$

By minimizing (over curves $f$):

$$\sum_{i=1}^{n} (Y_i - f(X_i))^2 + \int (f''(x))^2 \, dx$$

# Support Vector Machines (cont.)

Note: $\lambda$ is a “smoothing parameter”

Show SiZer\SmoothingSplinesFossils.mpg

Can show: $C$ works in a similar way

Suggests that: choice of $C$ is as hard as choosing $\lambda$

Smoothing Spline References:

Eubank (1988,1999) *Nonparametric Regression and Spline Smoothing*, Dekker

Wahba (1990) *Spline Models for Observational Data*, SIAM.

# Support Vector Machines (cont.)

Main Impact of $C$ for SVMs: High Dim'n Low Sample Size

Toy Example:

In 1st dim'n: Class 1: $N(1,1)$ Class 2: $N(-1,1)$ ($n = 20$ of each)

In dimensions $2,...,d$: independent $N(0,1)$

Fisher Linear Discrimination:

Show Svm\SVMeg3p1d3m1v1.mpg

- Gets great discrimination for higher $d$
- But finds useless "spurious" directions (nonrepeatable)

# Support Vector Machines (cont.)

SVM, $C = 1000$ (default)

Show Svm\SVMeg3p1d3m2v1.mpg

- Also has poor performance
- But in “different direction” (tries to max “gap” between)

SVM, $C = 10^{12}$

Show Svm\SVMeg3p1d3m3v1.mpg

- very similar to $C = 1000$

flip back and forth with Svm\SVMeg3p1d3m2v1.mpg

# Support Vector Machines (cont.)

SVM, $C = 10^{-6}$

Show Svm\SVMeg3p1d3m4v1.mpg

- Performance much improved
- Since very small weight given to “wrong side” points
- Found direction stays close to MLE
- Should be much more “repeatable” (for new data)

# Support Vector Machines (cont.)

Related Toy Example: Class 1: $N(3,0.3)$ Class 2: $N(-3,0.3)$

"populations farther apart"

- FLD still nonrepeatable for high $d$

Show Svm\SVMeg3p1d1m1v1.mpg

- SVM $C = 1000$ seems better, but still affected

Show Svm\SVMeg3p1d1m2v1.mpg

- SVM $C = 10^{-6}, 10^{12}$ both much better????

Show Svm\ SVMeg3p1d1m3v1.mpg & SVMeg3p1d1m4v1.mpg

# Support Vector Machines (cont.)

Caution:  SVM is not robust, instead “feels outliers”

Show SVM\SVMeg2p1v1.mpg

- reason is “higher penalty for data farther from plane”
- note “jumping effect” – nonlinear min’ing artifact????

Can get strange results in “indeterminate case”:

Show SVM\SVMeg2p3v1.mpg

- generally good, stable answer
- but hardly inside data at “crossing point”?

# Support Vector Machines (cont.)

Possible weakness:  can be “strongly driven by a few points”

Again show SVM\SVMeg2p4v1.mpg

- huge “range of chosen hyperplanes”
- but all are “pretty good discriminators”
- only happens when “whole range is OK”????

# Support Vector Machines (cont.)

Revisit toy examples (from Polynomial Embedding):

E.g. Parallel Clouds:

Show PolyEmbed\Peod1FLDcombine.pdf and Peod1SVMcombine.pdf

- SVM and FLD very comparable

E.g. Two Clouds:

Show PolyEmbed\PEtclFLDcombine.pdf and PEtclSVMcombine.pdf

- SVM better in linear case
- since doesn't "miss with covariance assumption"

# Support Vector Machines (cont.)

E.g. Split X:

Show PolyEmbed\Pexd3FLDcombine.pdf and Pexd3SVMcombine.pdf

- fairly comparable
- SVM had worse overfitting at cubic (could fix via C????)

E.g. Split X, parallel to axes (e.g. after ICA):

Show PolyEmbed\Pexd4FLDcombine.pdf and Pexd4SVMcombine.pdf

- fairly comparable
- SVM gives better “cutoffs” (since non-elliptical data)

# Support Vector Machines (cont.)

## E.g. Donut:

Show PolyEmbed\PedonFLDcombine.pdf and PedonSVMcombine.pdf

- fairly comparable
- SVM gives better “cutoffs” at higher degrees
- since non-elliptical data, in high degree embedded space

## E.g. Checkerboard – Kernel embedding

Show PolyEmbed\PEchbFLDe7.ps, PEchbSVMe7.ps & PEchbGLRe7.ps

- SVM gives better boundaries than FLD
- But not so good as GLR

# General Conclusion about Discrimination

"There Ain't No Such Thing As a Free Lunch"

I.e. each method can be:

- Great
- Very Poor

Depending on context, and data set at hand.

Thus useful to understand, and to have a big bag of tricks.

# Validation for Discrimination

How “well” does a method work?

Theoretical Answer: for a random point from the underlying distributions, what is the probability of “correct classification”

Naïve Empirical Answer: proportion of training data correctly classified

Problem 1: tendency towards “too optimistic”

Problem 2: Way off for overfitting (e.g. HDLSS)

Again show Svm\SVMeg3p1d2m1v1.mpg

# Validation for Discrimination (cont.)

Better empirical answers:  Cross-Validation

Simplest version:

- Use ½ the data to “train”, i.e. construct the discrim’n rule
- Use the other ½ to “assess”, i.e. compute error rate
- Unbiased est. of prob. of correct classification
- But get error rate for “wrong sample size”

# Validation for Discrimination (cont.)

Cross-validation (cont.)

More sophisticated version:  Leave – One - Out

- Train with all but one data point
- Use that for assessment
- Repeat for each data point
- Still unbiased est. of prob. of correct classification
- Much closer to correct sample size

# Validation for Discrimination (cont.)

E.g. In 1st dimension: Class 1: $N(\mu, 0.8)$ Class 2: $N(-\mu, 0.8)$

In dimensions $2, \ldots, d$: independent $N(0,1)$

- Bayes Rule: best discriminator using unknown dist'ns

(i.e. choose Class 1 for $X_1 > 0$)

# Validation for Discrimination (cont.)

$\mu = 1$: Small separation (hard discrimination)

show HDLSS\ HDLSSdiscCVm2.ps

- All are good in 1-d (better than Bayes by "luck")
- Performance degrades for higher dim. (except Bayes)
- SVMs better than FLD or GLR for higher dim.

(benefit of regularization)

- SVMs all pretty similar???? (unlike above)

  Again show SVM\SVMeg3p1d3m1v1.mpg, SVMeg3p1d3m2v1.mpg, SVMeg3p1d3m3v1.mpg, SVMeg3p1d3m4v1.mpg

- Since "data are separated", so no "wrong side data"????

# Validation for Discrimination (cont.)

$\mu = 1$: Revisit "Naïve Correct Class. Rate" vs. Cross Validation

show HDLSS\HDLSSegCVm2.ps

For linear methods (FLD, SVM):

- Naïve goes up quickly, and is too high
- CV seems "unbiased", with "some sampling variability"

Nonlinear GLR:

- feels sampling variability much more strongly?

# Validation for Discrimination (cont.)

$\mu = 2$: More separation (easy discrimination)

show HDLSS\ HDLSSdiscCVm3.ps

- Overall correct class'n rates much higher
- All methods perfect in lower dim'ns
- FLD and GLR feel over-fitting mosts strongly (high dim.)
- All SVMs same as each other (no "wrong side" data)
- SVMs as good or better than FLD or GLR everywhere
- FLD and GLR improve for higher dim'n?????

# Validation for Discrimination (cont.)

$\mu = 4$: Very wide separation (very easy discrimination)

show HDLSS\ HDLSSdiscCVm4.ps

- All methods nearly always perfect
- FLD and GLR have high dim'al problems (overfitting)
- GLR improves for highest dim'n????

# Validation for Discrimination (cont.)

$\mu = 0$: No separation (hardest discrimination)

show HDLSS\ HDLSSdiscCVm1.ps

- All methods have rate of correct class'n ~ ½
- I.e. as good as "classification by coin tossing"
- With "sampling variability" present
- Sometimes better than Bayes rule (just luck)
- SVMs most different from each other
- Since $C$ constraints are most "active"

# Validation for Discrimination (cont.)

Variations on “good performance”:

1. Stability: how much change for new data?

   Useful tool: bootstrap, either “quantitative” or “visual”

   E.g. Corpus Callosum data, Schizophrenics vs. Controls

Show CorpColl\CCFrawSs3.mpg, CCFrawCs3.mpg

- Fisher Linear Discrimination found a useless direction

Show CorpColl\CCFfldSCs3.mpg, CCFfldSCs3mag.mpg, CCFfldSCs3VisStab.mpg

- Orthogonal Subspace Projection found something

Show CorpColl\CCFospSCs3RS11o2.mpg, CCFospSCs3RS12o1.mpg, CCFospSCs3RS11o2VS.mpg, CCFospSCs3RS12o1VS.mpg

# Validation for Discrimination (cont.)

Variations on “good performance” (cont.)

2. Significant effect: Is there really something there?

   Useful tool: Permutation

   Use random relabellings of classes

E.g. Corpus Callosum data, Schizophrenics vs. Controls

Show lower left of CorpColl\CCFospSCs3RS1stab.ps