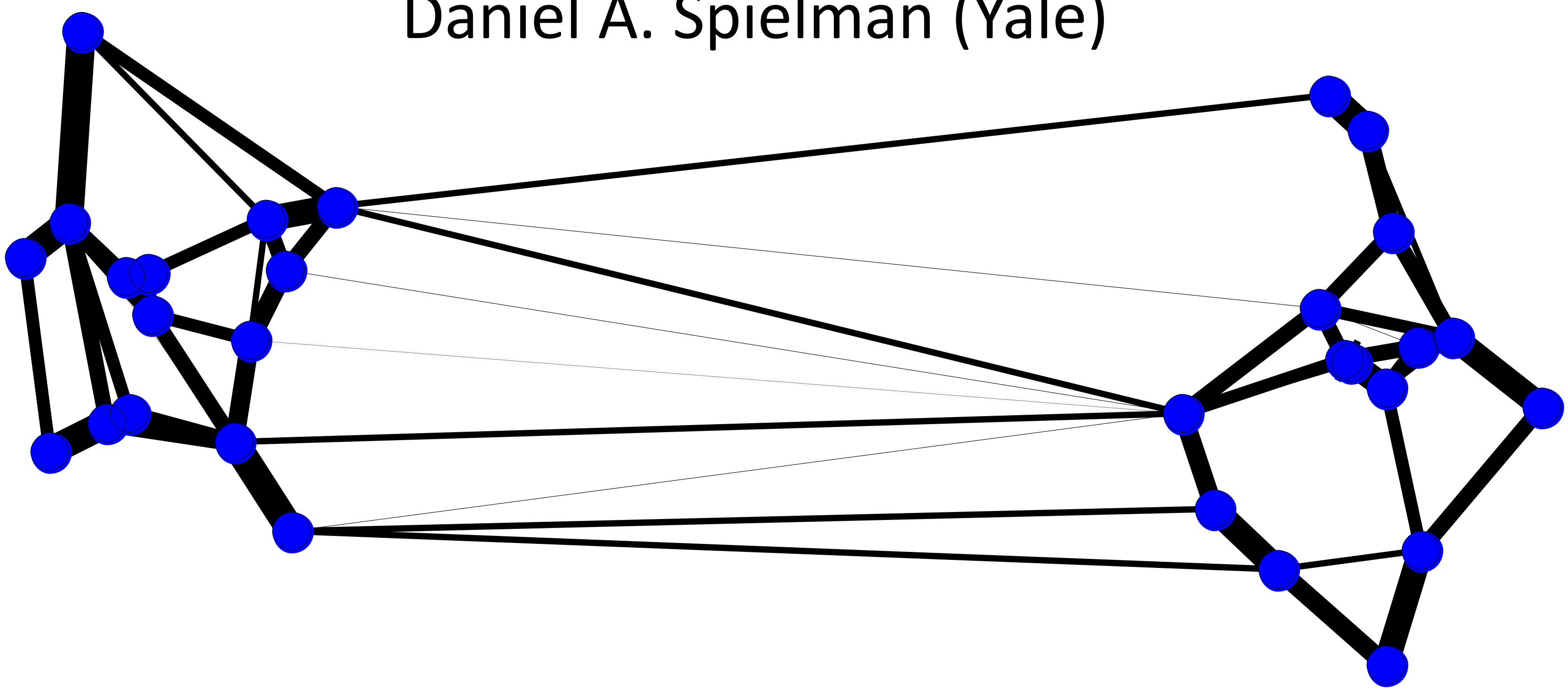
# Fitting a Graph to Vector Data

Samuel I. Daitch (Yale)
Jonathan A. Kelner (MIT)
Daniel A. Spielman (Yale)



International Conference on Machine Learning
Montreal 15 June 2009

Given a collection of vectors $x_1, \ldots, x_n \in \mathbb{R}^d$

Construct an undirected weighted graph on the vertices $x_1, \ldots, x_n$ that is

1. Helps solve ML problems on the vectors
2. Efficiently computable
3. Sparse
4. Theoretically well-motivated
5. Has interesting combinatorial properties

# Outline

Standard graphs
Motivate & define our graphs
Combinatorial properties
Use in classification, regression and clustering
Efficient computation

Open Questions

# Standard ways to make graphs

Choose one from each column

| Choice of edges | Weights of edges |
| --- | --- |
| k nearest neighbors | unweighted |
| neighbors if distance less than threshold $\delta$ | $\exp\left(- \left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|^2 / 2\sigma^2\right)$ |

# Motivation

Consider regression problem $y_i = f(\boldsymbol{x}_i)$
solving for $y_i$ given $\{y_j : j \neq i\}$

Given a weighted graph, natural guess is

$$\frac{1}{d_i} \sum_{j \sim i} w_{i,j} y_j$$

where $w_{i,j} = w_{j,i} \geq 0$ are the edge weights

and $d_i = \sum_{i \sim j} w_{i,j}$ are the weighted degrees

# Motivation

Try to get right on coordinate vectors,

$$\boldsymbol{x}_i = (x_i^1, \ldots, x_i^d)$$

For $x_i^k$ estimate $est_i^k = \frac{1}{d_i} \sum_j w_{i,j} x_j^k$

# Motivation

Try to get right on coordinate vectors,

$$\boldsymbol{x}_i = (x_i^1, \ldots, x_i^d)$$

For $x_i^k$ estimate $est_i^k = \frac{1}{d_i} \sum_j w_{i,j} x_j^k$

We choose weights to minimize
sum of squared errors over all coordinate vectors:

$$\sum_{k=1}^{d} \sum_{i=1}^{n} \left(x_i^k - est_i^k\right)^2$$

# Motivation

Try to get right on coordinate vectors,

$$\boldsymbol{x}_i = (x_i^1, \ldots, x_i^d)$$

For $x_i^k$ estimate $est_i^k = \frac{1}{d_i} \sum_j w_{i,j} x_j^k$

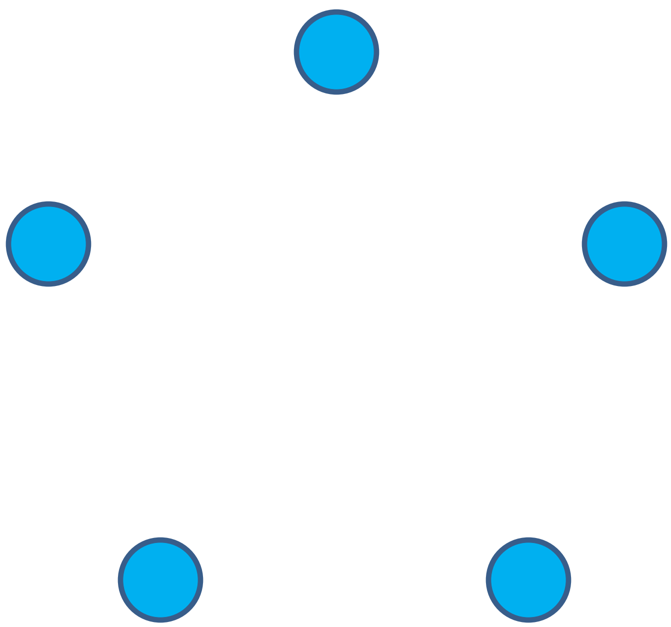
We choose weights to minimize
sum of squared errors over all coordinate vectors:

$$\sum_{k=1}^{d} \sum_{i=1}^{n} \left( x_i^k - est_i^k \right)^2$$

*But, leads to a non-convex problem, and…*

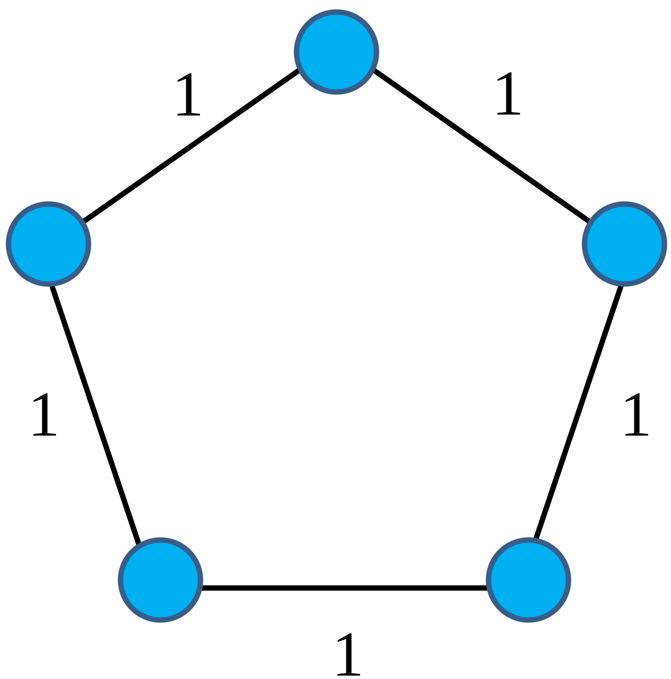
# Motivation, tricky example

Consider minimizing $$\sum_{k=1}^{d}\sum_{i=1}^{n}\left(x_i^k - est_i^k\right)^2$$



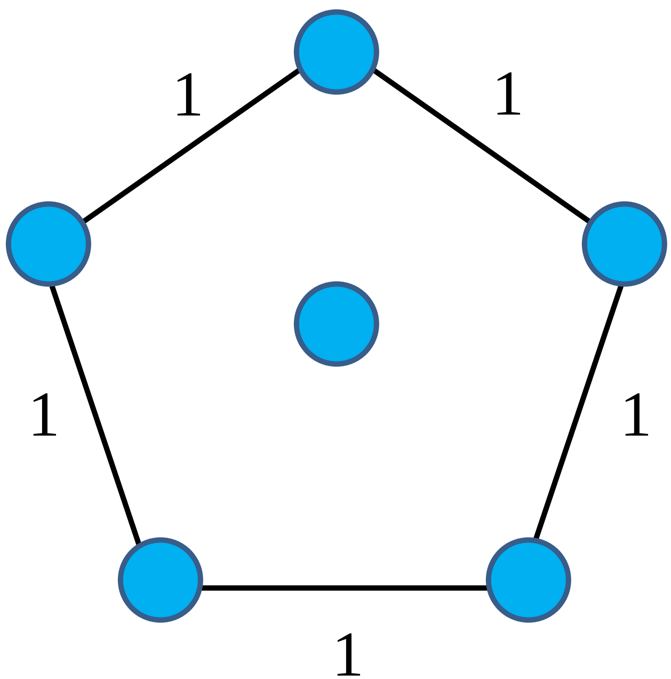
# Motivation, tricky example

Consider minimizing $$\sum_{k=1}^{d}\sum_{i=1}^{n}\left(x_i^k - est_i^k\right)^2$$



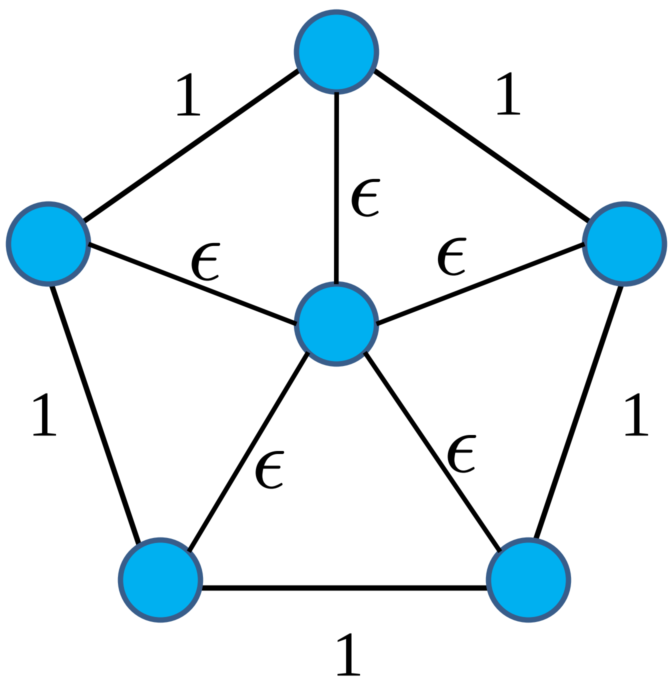
# Motivation, tricky example

Consider minimizing $$\sum_{k=1}^{d}\sum_{i=1}^{n}\left(x_i^k - est_i^k\right)^2$$



# Motivation, tricky example

Consider minimizing $$\sum_{k=1}^{d}\sum_{i=1}^{n}\left(x_i^k - est_i^k\right)^2$$




Just better as $\epsilon$ goes to 0

# Motivation, fixing bad example

Instead minimize sum of squared errors, weighted by weighted degree

$$\sum_{k=1}^{d} \sum_{i=1}^{n} d_i \left(x_i^k - est_i^k\right)^2$$

# Motivation, fixing bad example

Instead minimize sum of squared errors, weighted by weighted degree

$$\sum_{k=1}^{d}\sum_{i=1}^{n} d_i \left(x_i^k - est_i^k\right)^2 = \|LX\|_F^2$$

# Motivation, fixing bad example

Instead minimize sum of squared errors, weighted by weighted degree

$$\sum_{k=1}^{d}\sum_{i=1}^{n} d_i \left(x_i^k - est_i^k\right)^2 = \|LX\|_F^2$$

To avoid degeneracy, require either

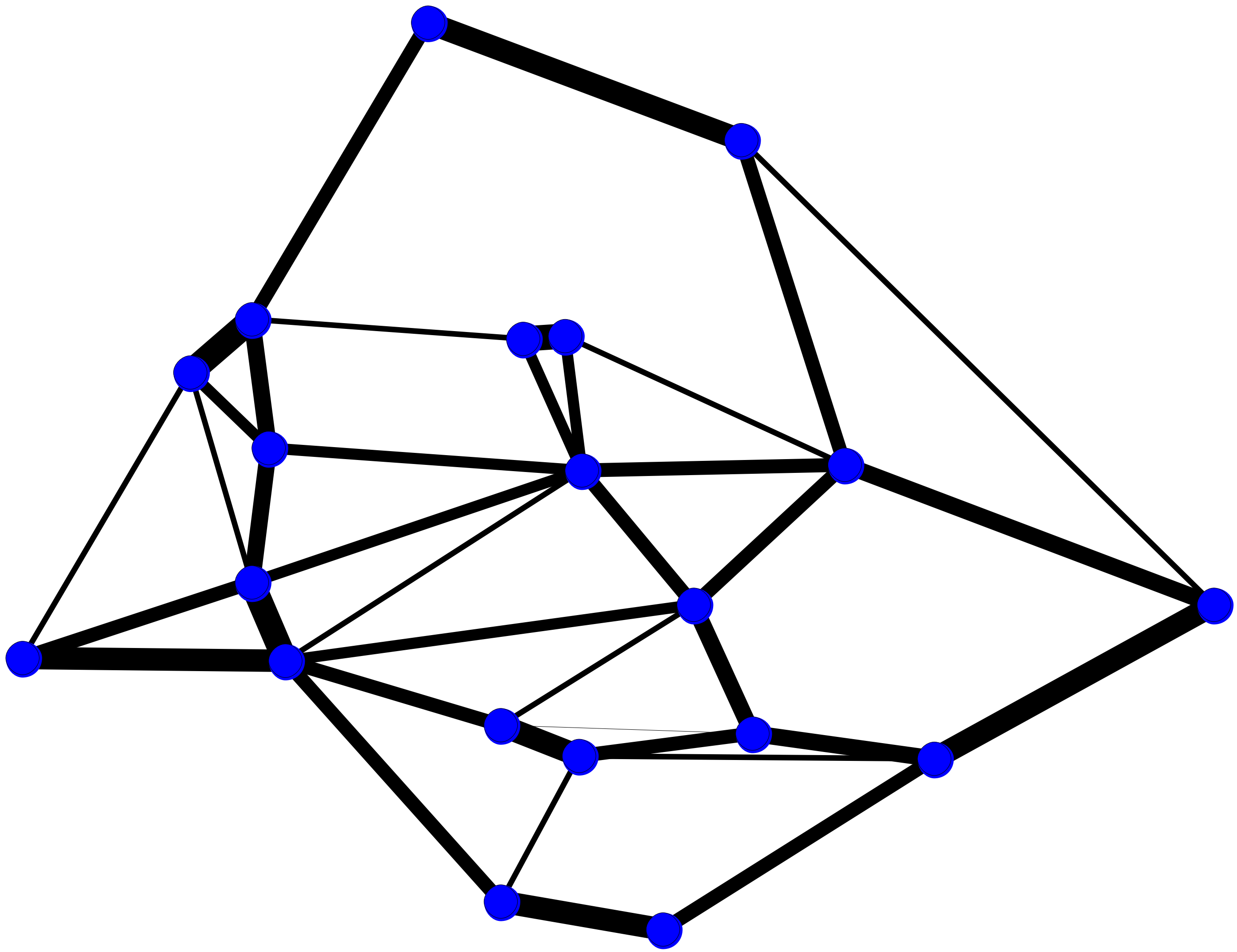
$$d_i = \sum_{j \sim i} w_{i,j} \geq 1$$ (hard graph)

or

$$\sum_{i:d_i<1} (1-d_i)^2 \cdot \quad \alpha n$$ ($\alpha$-soft graph)

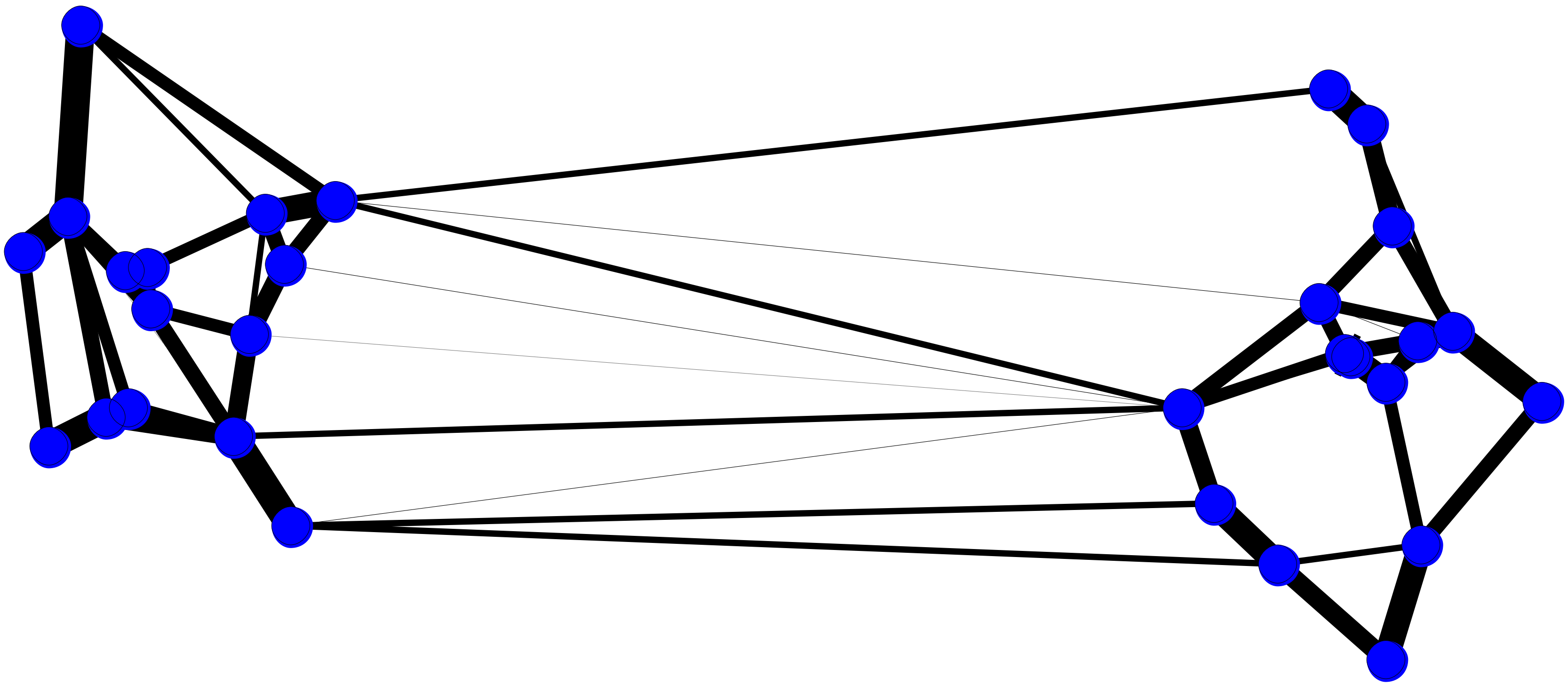
# Example

Random points in 2 dimensions (0.1-soft)



# Example

## Two Clusters (0.1-soft)



# Related to

Locally linear embeddings [Roweis-Saul 2000]

Chooses allowable neighbors, then chooses asymmetric weights to minimize the same objective function.

K-means

If restrict graph to be a union of complete graphs, one for each cluster, are minimizing same objective function.

# Unique?

Not necessarily:

Not unique on highly symmetric point sets.

But, almost always unique.

Conjecture:

Unique with probability 1 after infinitesimally small random perturbation.

# Sparsity

Theorem:

For n vectors in d dimensions,

always is a graph with at most (d+1)n edges.

i.e. average (combinatorial) degree 2(d+1).

# Degrees on real data

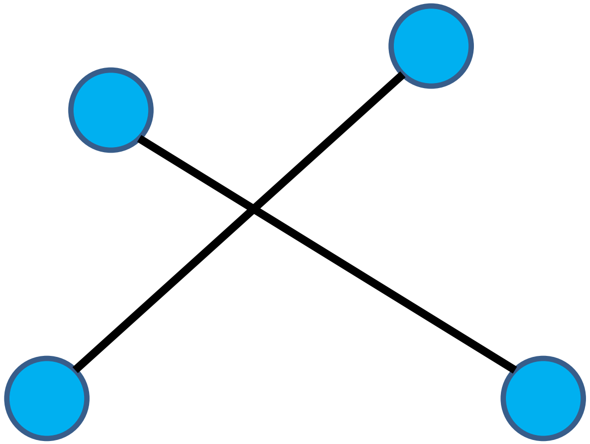
| Data set | Type | n | dim | ave deg (hard) | ave deg (0.1-soft) |
|---|---|---|---|---|---|
| abalone | regression | 4177 | 8 | 13.1 | 12.7 |
| glass | 6 classes | 214 | 9 | 8.9 | 8.6 |
| heart | 2 classes | 270 | 13 | 11.1 | 11 |
| housing | regression | 506 | 13 | 8.8 | 10.1 |
| ionosphere | 2 classes | 351 | 34 | 13 | 11.9 |
| iris | 3 classes | 150 | 4 | 7 | 7 |
| machine | regression | 209 | 6 | 8.9 | 8 |
| mpg | regression | 392 | 7 | 7.6 | 8.8 |
| pima | 2 classes | 768 | 8 | 11.2 | 10.9 |
| sonar | 2 classes | 208 | 60 | 12.7 | 13 |
| vehicle | 4 classes | 846 | 18 | 11.8 | 11.5 |
| vowel | 11 classes | 990 | 10 | 10.4 | 10.1 |
| wine | 3 classes | 178 | 13 | 9.1 | 9.9 |

# Degrees on real data

| Data set | Type | n | dim | ave deg (hard) | ave deg (0.1-soft) |
|---|---|---|---|---|---|
| abalone | regression | 4177 | 8 | 13.1 | 12.7 |
| glass | 6 classes | 214 | 9 | 8.9 | 8.6 |
| heart | 2 classes | 270 | 13 | 11.1 | 11 |
| housing | regression | 506 | 13 | 8.8 | 10.1 |
| ionosphere | 2 classes | 351 | 34 | 13 | 11.9 |
| iris | 3 classes | 150 | 4 | 7 | 7 |
| machine | regression | 209 | 6 | 8.9 | 8 |
| mpg | regression | 392 | 7 | 7.6 | 8.8 |
| pima | 2 classes | 768 | 8 | 11.2 | 10.9 |
| sonar | 2 classes | 208 | 60 | 12.7 | 13 |
| vehicle | 4 classes | 846 | 18 | 11.8 | 11.5 |
| vowel | 11 classes | 990 | 10 | 10.4 | 10.1 |
| wine | 3 classes | 178 | 13 | 9.1 | 9.9 |

# Planarity

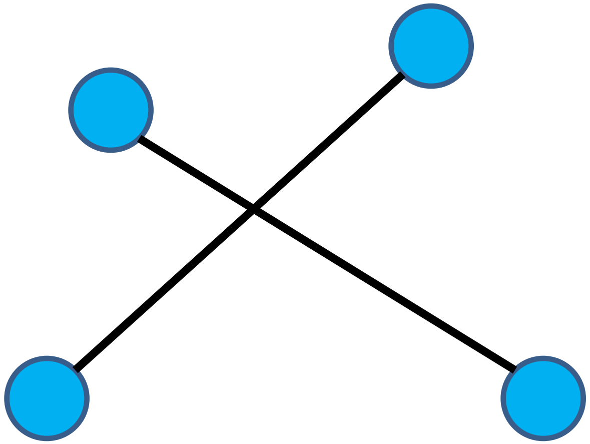
For every set of vectors in 2 dimensions, there is a hard and $\alpha$-soft graph that is planar



no crossings

# Planarity

For every set of vectors in 2 dimensions, there is a hard and $\alpha$-soft graph that is planar



no crossings

In general dimension, if consider each clique as simplex, then is a simplicial complex.

# Classification and Regression Experiments

Given a set $S$ of vectors with labels, $y_i \in \mathbb{R}$
compute $z$ minimizing

$$\sum_{i \sim j} w_{i,j}(z_i - z_j)^2 = z^T L z$$

subject to $z_i = y_i, \text{ for } i \in S$ [ZGL ‘03]

With 2 classes, use an indicator vector for one.
With k classes, use k indicator vectors.

# Classification Experiments

Used 10-fold cross validation

We had no parameters to train.

For other means of choosing graphs,
chose their parameters by gridding over choices,
and evaluating by leave-one-out cross-validation.

Repeated 100 times

# Classification experiments

| Data set | n | dim | classes | 0.1-soft | knn | thresh |
|---|---|---|---|---|---|---|
| glass | 214 | 9 | 6 | 28.3 | **26.92** | 33.3 |
| heart | 270 | 13 | 2 | 17.81 | **16.05** | 16.1 |
| ionosphere | 351 | 34 | 2 | **5.57** | 18.5 | 6.34 |
| iris | 150 | 4 | 3 | **4.21** | 4.46 | 6.2 |
| pima | 768 | 8 | 2 | 26.61 | **24.54** | 26.45 |
| sonar | 208 | 60 | 2 | **8.64** | 13.8 | 14.94 |
| vehicle | 846 | 18 | 4 | **22.47** | 27.7 | 29.98 |
| vowel | 990 | 10 | 11 | **0.95** | 2.62 | 0.98 |
| wine | 178 | 13 | 3 | **2.62** | 2.86 | 3.64 |

tried unweighted, and exponential weights varying $\sigma$
varied k for knn, and $\delta$ for threshold

# Classification experiments

| Data set | n | dim | classes | 0.1-soft | knn | thresh | libsvm |
|---|---|---|---|---|---|---|---|
| glass | 214 | 9 | 6 | 28.3 | **26.92** | 33.3 | 31.44 |
| heart | 270 | 13 | 2 | 17.81 | **16.05** | 16.1 | 17.01 |
| ionosphere | 351 | 34 | 2 | **5.57** | 18.5 | 6.34 | 6.2 |
| iris | 150 | 4 | 3 | 4.21 | 4.46 | 6.2 | **3.87** |
| pima | 768 | 8 | 2 | 26.61 | 24.54 | 26.45 | **23.24** |
| sonar | 208 | 60 | 2 | **8.64** | 13.8 | 14.94 | 11.71 |
| vehicle | 846 | 18 | 4 | 22.47 | 27.7 | 29.98 | **14.87** |
| vowel | 990 | 10 | 11 | 0.95 | 2.62 | 0.98 | **0.64** |
| wine | 178 | 13 | 3 | 2.62 | 2.86 | 3.64 | **2.57** |

For libsvm [ChangLin],
chose params by 10-fold CV on training dat

# Regression Error

| Data set | n | dim | hard | 0.1-soft | knn | thresh | epsilon-svr |
|---|---|---|---|---|---|---|---|
| abalone | 4177 | 8 | **0.479** | 0.482 | 0.492 | 0.657 | Too long |
| housing | 506 | 13 | **0.136** | 0.138 | 0.224 | 0.507 | 0.138 |
| machine | 209 | 6 | 0.17 | 0.185 | **0.164** | 0.608 | 0.394 |
| mpg | 392 | 7 | 0.12 | **0.118** | 0.137 | 0.145 | 0.128 |

Data normalized to have variance 1.
2-fold CV on abalone, 10-fold otherwise

# Clustering Experiments (0.1-soft)

| Data set | n | dim | k | k-means | NJW | 0.1-soft Nvec = k |
|---|---|---|---|---|---|---|
| glass | 214 | 9 | 6 | **0.41** | 0.43 | 0.45 |
| heart | 270 | 13 | 2 | 0.41 | 0.42 | **0.19** |
| ionosphere | 351 | 34 | 2 | **0.29** | 0.33 | 0.33 |
| iris | 150 | 4 | 3 | **0.11** | 0.33 | 0.17 |
| pima | 768 | 8 | 2 | **0.34** | 0.35 | 0.35 |
| sonar | 208 | 60 | 2 | 0.45 | 0.47 | **0.40** |
| vehicle | 846 | 18 | 4 | **0.55** | 0.62 | 0.56 |
| Vowel | 990 | 10 | 11 | 0.66 | 0.76 | **0.65** |
| wine | 178 | 13 | 3 | 0.30 | 0.33 | **0.03** |

NJW: Ng-Jordan-Weiss, as implemented in Spider.
Use graph, run k-means in Nvec eigenvectors.

# Clustering Experiments (0.1-soft)

| Data set | n | dim | k | k-means | NJW | 0.1-soft Nvec = k | 0.1-soft Nvec chosen |
|---|---|---|---|---|---|---|---|
| glass | 214 | 9 | 6 | **0.41** | 0.43 | 0.45 | 0.45 (12) |
| heart | 270 | 13 | 2 | 0.41 | 0.42 | **0.19** | 0.19 (2) |
| ionosphere | 351 | 34 | 2 | 0.29 | 0.33 | 0.33 | **0.09** (15) |
| iris | 150 | 4 | 3 | 0.11 | 0.33 | 0.17 | **0.09** (8) |
| pima | 768 | 8 | 2 | **0.34** | 0.35 | 0.35 | 0.35 (12) |
| sonar | 208 | 60 | 2 | 0.45 | 0.47 | 0.40 | **0.35** (35) |
| vehicle | 846 | 18 | 4 | 0.55 | 0.62 | 0.56 | **0.53** (6) |
| Vowel | 990 | 10 | 11 | 0.66 | 0.76 | 0.65 | **0.60** (30) |
| wine | 178 | 13 | 3 | 0.30 | 0.33 | 0.03 | **0.03** (3) |

NJW: Ng, Jordan, Weiss, as implemented in Spider.
Use graph, run k-means in Nvec eigenvectors.

# Computing the Graphs

Hard graphs: quadratic program
Soft graphs: non-neg linear least squares

Know solution is sparse,
so look for solution among a subset of edges.

Given solution restricted to subset of edges
check gradient of obj func wrt unused edges
if all positive, finished
otherwise, include some and solve again.

# Computation Time (secs)

| Data set | Type | n | dim | soft time | hard time |
|---|---|---|---|---|---|
| abalone | regression | 4177 | 8 | 1582 | 49986 |
| glass | 6 classes | 214 | 9 | 8 | 22 |
| heart | 2 classes | 270 | 13 | 12 | 36 |
| housing | regression | 506 | 13 | 23 | 114 |
| ionosphere | 2 classes | 351 | 34 | 72 | 148 |
| iris | 3 classes | 150 | 4 | 4 | 17 |
| machine | regression | 209 | 6 | 11 | 26 |
| mpg | regression | 392 | 7 | 16 | 43 |
| pima | 2 classes | 768 | 8 | 111 | 529 |
| sonar | 2 classes | 208 | 60 | 38 | 50 |
| vehicle | 4 classes | 846 | 18 | 159 | 889 |
| vowel | 11 classes | 990 | 10 | 85 | 706 |
| wine | 3 classes | 178 | 13 | 6 | 15 |

# Open Questions

Are there other natural and useful graphs?

Are there interesting ways to
modify/parameterize our construction?

Are our graphs connected?
Are our graphs unique (under perturbation)?

Do there exist sparse approximate solutions in
all dimensions?

How to interpolate off the graph, or add points?