Tangent approximation to principal manifolds and its application to regression modelling

Ludger Evers

Principal Manifolds Workshop, Leicester 2006

(joint work with Jochen Einbeck)

Overview

- Piecewise linear approximations to principal manifolds ("k-segments")
  - Motivation in the 2D case
  - Generalisation to principal manifolds
- Application to dimension reduction for supervised learning
  - Is the principal component actually the direction we are after?
  - Alternative directions
  - Example from astronomy
- Projection trees as weak learners

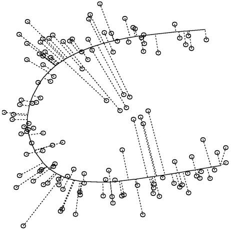
Hastie-Stuetzle principal curves

Hastie-Stuetzle principal curves: Definition

A smooth non-intersecting curve $\mathbf{m} : I \longrightarrow \mathbb{R}^p$ is called a principal curve if it is self-consistent, i.e.

$$\mathbb{E}(\mathbf{x}|\eta_{\mathbf{m}}(\mathbf{x}) = \eta) = \mathbf{m}(\eta) \qquad \text{for a.e. } \eta \in I.$$

$\eta_{\mathbf{m}}(\mathbf{x})$ is hereby the projection index of $\mathbf{x}$ onto $\mathbf{m}$.



Relationship to principal components

If the HS principal curve is linear, then it is a principal component.

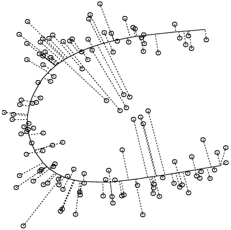
Hastie-Stuetzle principal curves

Hastie-Stuetzle principal curves: Definition

A smooth non-intersecting curve $\mathbf{m} : I \longrightarrow \mathbb{R}^p$ is called a principal curve if it is self-consistent, i.e.

$$\mathbb{E}(\mathbf{x}|\eta_{\mathbf{m}}(\mathbf{x}) = \eta) = \mathbf{m}(\eta) \qquad \text{for a.e. } \eta \in I.$$

$\eta_{\mathbf{m}}(\mathbf{x})$ is hereby the projection index of $\mathbf{x}$ onto $\mathbf{m}$.

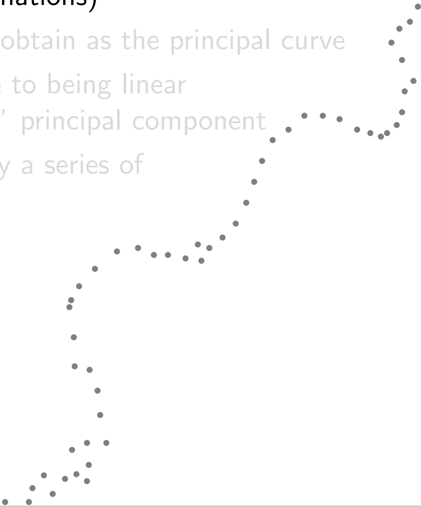


Relationship to principal components

If the HS principal curve is linear, then it is a principal component.

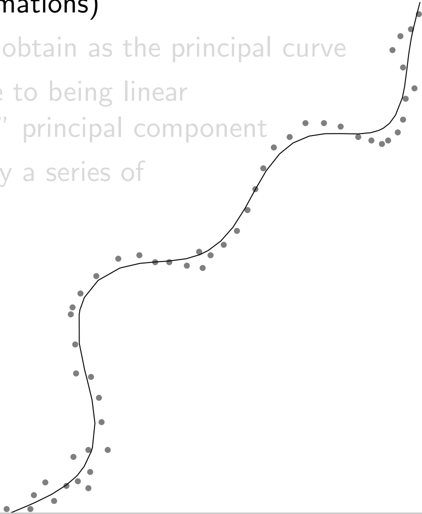
Core idea: Tangent approximations to principal curves

- Basic idea: Model tangents instead of the principal curve (Tangents are local linear approximations)
- However: Tangents as difficult to obtain as the principal curve
- Locally the principal curve is close to being linear
  $\rightsquigarrow$ not too different from a "local" principal component
- Approximate the principal curve by a series of "local" principal components (some sort of tangents)



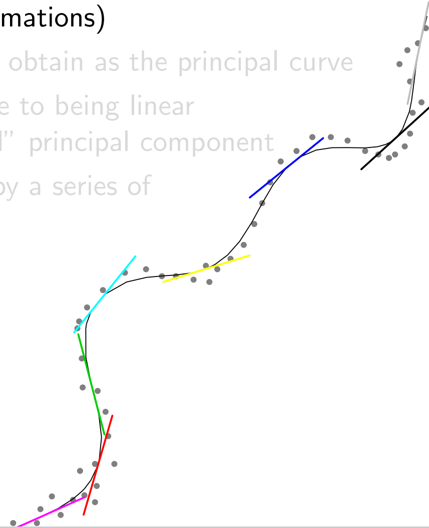
Core idea: Tangent approximations to principal curves

- Basic idea: Model tangents instead of the principal curve (Tangents are local linear approximations)
- However: Tangents as difficult to obtain as the principal curve
- Locally the principal curve is close to being linear
  $\rightsquigarrow$ not too different from a "local" principal component
- Approximate the principal curve by a series of "local" principal components (some sort of tangents)



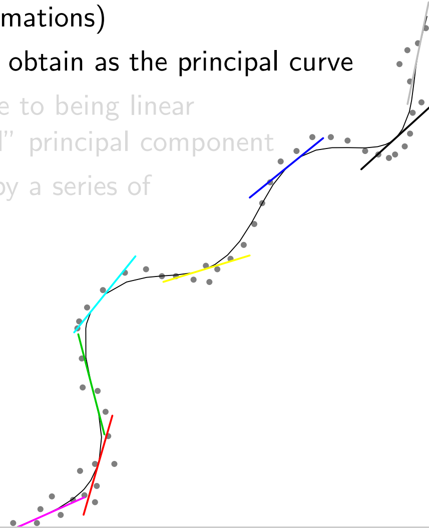
Core idea: Tangent approximations to principal curves

- Basic idea: Model tangents instead of the principal curve (Tangents are local linear approximations)
- However: Tangents as difficult to obtain as the principal curve
- Locally the principal curve is close to being linear
  $\rightsquigarrow$ not too different from a “local” principal component
- Approximate the principal curve by a series of “local” principal components (some sort of tangents)



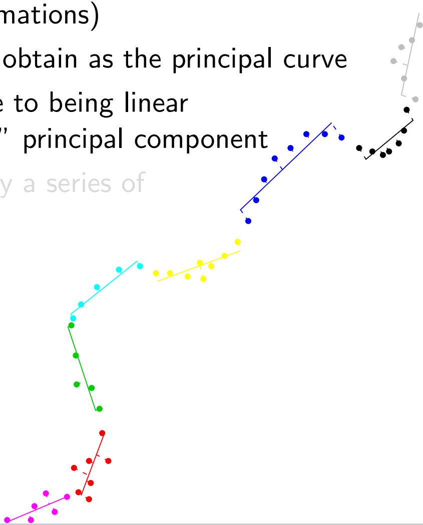
Core idea: Tangent approximations to principal curves

- Basic idea: Model tangents instead of the principal curve (Tangents are local linear approximations)
- However: Tangents as difficult to obtain as the principal curve
- Locally the principal curve is close to being linear $\rightsquigarrow$ not too different from a "local" principal component
- Approximate the principal curve by a series of "local" principal components (some sort of tangents)



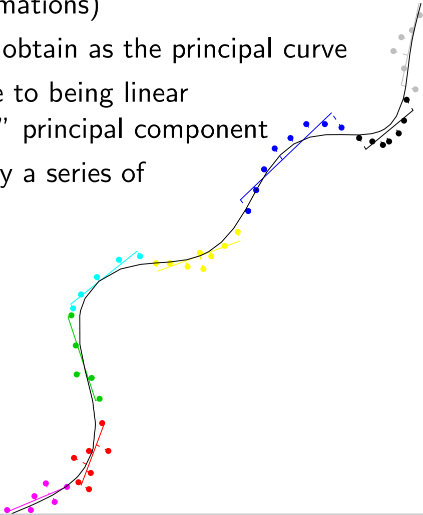
Core idea: Tangent approximations to principal curves

- Basic idea: Model tangents instead of the principal curve (Tangents are local linear approximations)
- However: Tangents as difficult to obtain as the principal curve
- Locally the principal curve is close to being linear
  $\rightsquigarrow$ not too different from a "local" principal component
- Approximate the principal curve by a series of "local" principal components (some sort of tangents)



Core idea: Tangent approximations to principal curves

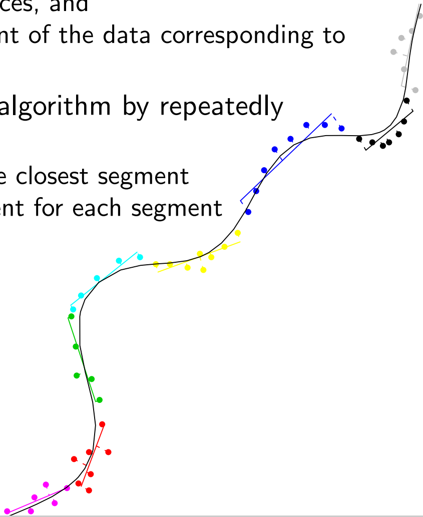
- Basic idea: Model tangents instead of the principal curve (Tangents are local linear approximations)
- However: Tangents as difficult to obtain as the principal curve
- Locally the principal curve is close to being linear
  $\rightsquigarrow$ not too different from a "local" principal component
- Approximate the principal curve by a series of "local" principal components (some sort of tangents)



Core idea: some more details

- We have to . . .
  1. split the principal curve into pieces, and
  2. compute the principal component of the data corresponding to each part
- Can be done with a k-means-like algorithm by repeatedly iterating . . .
  1. Allocate each observation to the closest segment
  2. Compute the principal component for each segment
- Important questions:
  - How many segments?
  - Which initial values?

⇝ Start with global principal component and recursively split (and combine) partitions.



The algorithm

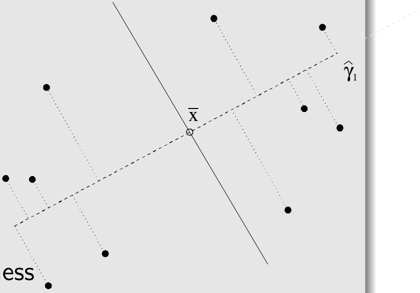
Algorithm

Iterate until convergence . . .

1. If necessary split some of the partitions.
2. If possible combine neighbouring partitions.
3. Update the partitions and local principal components by iterating
    1. Allocate each observation to the closest segment
    2. Compute the principal component for each segment

Splitting partitions (step 1)

- Very much like CARTs the algorithm tries at every stage to split every partition.
- Enough to split each partition P in the middle, i.e.
  $L := \{i \in P : (\mathbf{x}_i - \bar{\mathbf{x}}_P)'\hat{\gamma}_1^P \leq 0\}$
  $R := \{i \in P : (\mathbf{x}_i - \bar{\mathbf{x}}_P)'\hat{\gamma}_1^P > 0\}$
  ($\hat{\gamma}_1^P$ is the first principal component in P, $\bar{\mathbf{x}}_P$ the centroid)



- Only retain splits for with the goodness

$$\frac{|L| \cdot \frac{\hat{\lambda}_1^L}{\hat{\lambda}_1^L + \hat{\lambda}_2^L} + |R| \cdot \frac{\hat{\lambda}_1^R}{\hat{\lambda}_1^R + \hat{\lambda}_2^R}}{|P| \cdot \frac{\hat{\lambda}_1^P}{\hat{\lambda}_1^P + \hat{\lambda}_2^P}} > G_S$$

($\frac{\hat{\lambda}_1^P}{\hat{\lambda}_1^P + \hat{\lambda}_2^P}$ is the variance proportion of the first principal component)

Combining neighbouring partitions (step 2)

- Check for each neighbouring partition whether they can be combined. (Two partitions L and R are neighbours if at least one element of L has R as second-closest segment (and vice versa).)
- Use the criterion $g_{L,R}$ from above. Combine partitions with $g_{L,R} < G_C$.

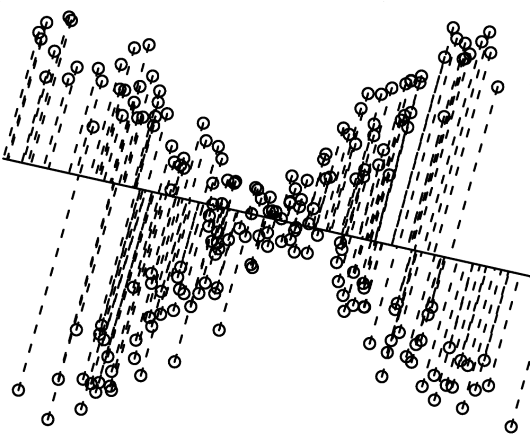
It might be beneficial to "enforce" a certain number of splits. (Not much of a problem as they can be undone by the algorithm by combining partitions)

Combining neighbouring partitions (step 2)

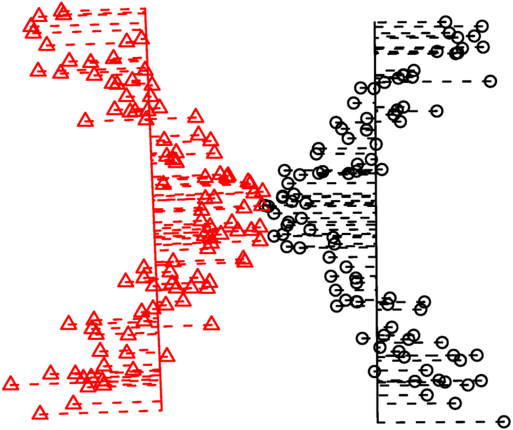
- Check for each neighbouring partition whether they can be combined. (Two partitions L and R are neighbours if at least one element of L has R as second-closest segment (and vice versa).)
- Use the criterion $g_{L,R}$ from above. Combine partitions with $g_{L,R} < G_C$.

It might be beneficial to "enforce" a certain number of splits.
(Not much of a problem as they can be undone by the algorithm by combining partitions)

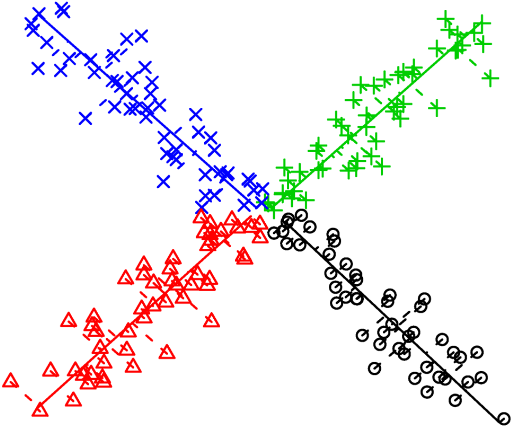
A simple example



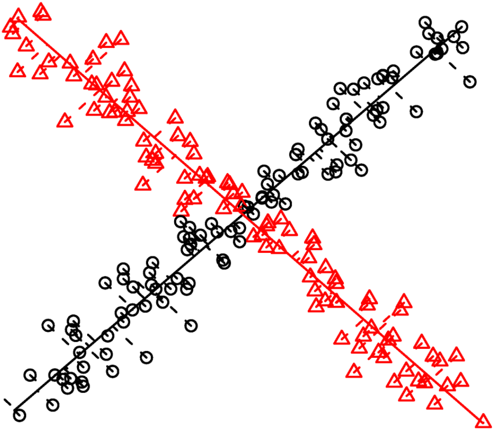
A simple example



A simple example



A simple example



Pros and Cons

Pros

- Simple and fast algorithm
- Straightforward generalisation to r-dimensional manifolds in p-dimensional space
- Discontinuity allows avoiding the problem of "warping"
- Straightforward to deal with missing values

Cons

- Heuristic approach (but can be motivated as "hard" version of a MLE to a mixture problem)
- Discontinuous approximation to the principal curve/manifold
  $\rightsquigarrow$ interpretation more difficult
  $\rightsquigarrow$ high variance

Pros and Cons

Pros

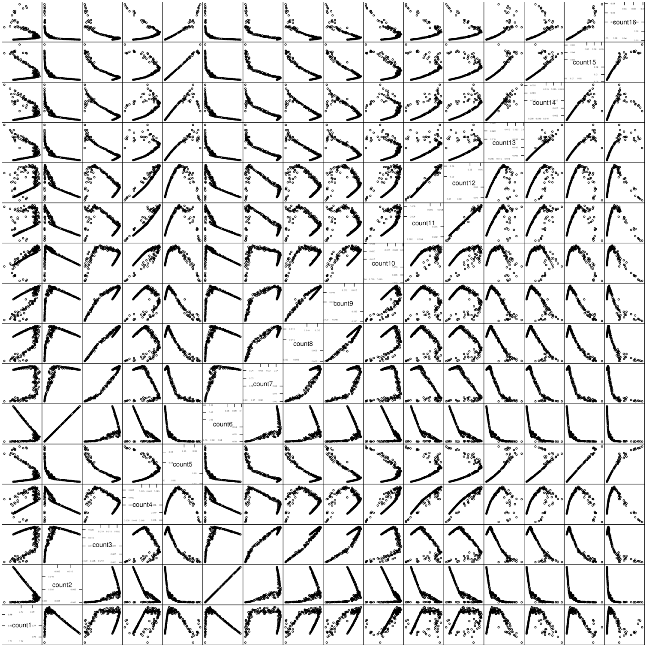
- Simple and fast algorithm
- Straightforward generalisation to r-dimensional manifolds in p-dimensional space
- Discontinuity allows avoiding the problem of “warping”
- Straightforward to deal with missing values

Cons

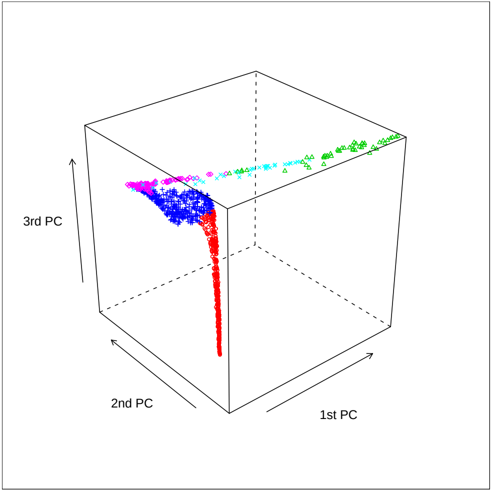
- Heuristic approach (but can be motivated as “hard” version of a MLE to a mixture problem)
- Discontinuous approximation to the principal curve/manifold
  $\rightsquigarrow$ interpretation more difficult
  $\rightsquigarrow$ high variance

Example: Photon counts

- Objective: Estimate physical properties of stars based on photometric data (photon counts for 16 frequency/colour bands)
- Model to be used to a catalogue of every object in the sky brighter than V=20 (GAIA satellite to be launched in 2011)
- We will focus on the prediction of the temperature (others a lot harder).
- Photon counts known to lie in a lower-dimensional manifold.
- We will use five four-dimensional hyperplane segments to approximate the manifold.


count16
count15
count14
count13
count12
count11
count10
count9
count8
count7
count6
count5
count4
count3
count2
count1

Results



Coverage R_C of

$$0.8325 = 1 - \frac{\text{Residual sum of squares of the k-segments model}}{\text{Residual sum of squares of the principal components}}$$

Local dimension reduction of the covariate space

- Situation: Supervised problem with large covariate space
- Use k-segments for local dimension reduction ("principal manifold as regulariser")
- Simple idea: Fit a regression / classification model in each segment
- "Soft thresholds", i.e. all data used is in each segment, however using weights:
- Weight of the *i*-th observation for the *k*-th segment:

$$w_{ik} = \exp(-\rho d_{ik}^2)$$

  (d_{ik} distance of the *i*-th observation from the *k*-th segment)
- Benefit of soft thresholds: Continuous prediction

Example: photon counts (ctd.)

	L_1 loss (relative to constant model)			
	training error		validation error	
Linear model in each partition	458.65	(7.8%)	475.03	(7.4%)
Gaussian SVR in each partition	237.43	(4.1%)	254.16	(4.1%)
(Global Gaussian SVR)	402.08	(6.9%)	411.91	(6.4%)

Should we really use the local principal components?

We want to do supervised learning. Should we then use an entirely unsupervised method for (local) dimension reduction?

- Recall the objectives of dimension reduction.
  1. Project data onto lower-dimensional manifold / subspace …
  2. …under preservation of the relevant structure
- Rationale for using principal components:

  Variance = Information
- But do we have

  Variance = Information relevant to us ???
- Principal components and manifolds to a good job for objective (1). Objective (2) is not at all guaranteed.
- Example microarray data: Main source of variability usually some sort of contamination.

$\rightsquigarrow$ Maybe there are better projection directions.

Should we really use the local principal components?

We want to do supervised learning. Should we then use an entirely unsupervised method for (local) dimension reduction?

- Recall the objectives of dimension reduction.
  1. Project data onto lower-dimensional manifold / subspace …
  2. … under preservation of the relevant structure
- Rationale for using principal components:

  $$\text{Variance} = \text{Information}$$
- But do we have

  $$\text{Variance} = \text{Information relevant to us ???}$$
- Principal components and manifolds to a good job for objective (1). Objective (2) is not at all guaranteed.
- Example microarray data: Main source of variability usually some sort of contamination.

⇝ Maybe there are better projection directions.

Should we really use the local principal components?

We want to do supervised learning. Should we then use an entirely unsupervised method for (local) dimension reduction?

- Recall the objectives of dimension reduction.
  1. Project data onto lower-dimensional manifold / subspace …
  2. …under preservation of the relevant structure
- Rationale for using principal components:

$$\text{Variance} = \text{Information}$$

- But do we have

$$\text{Variance} = \text{Information relevant to us ???}$$

- Principal components and manifolds to a good job for objective (1). Objective (2) is not at all guaranteed.
- Example microarray data: Main source of variability usually some sort of contamination.

$\rightsquigarrow$ Maybe there are better projection directions.

Should we really use the local principal components?

We want to do supervised learning. Should we then use an entirely unsupervised method for (local) dimension reduction?

- Recall the objectives of dimension reduction.
  1. Project data onto lower-dimensional manifold / subspace …
  2. … under preservation of the relevant structure
- Rationale for using principal components:

  Variance = Information
- But do we have

  Variance = Information relevant to us ???
- Principal components and manifolds to a good job for objective (1). Objective (2) is not at all guaranteed.
- Example microarray data: Main source of variability usually some sort of contamination.

⇝ Maybe there are better projection directions.

Should we really use the local principal components?

We want to do supervised learning. Should we then use an entirely unsupervised method for (local) dimension reduction?

- Recall the objectives of dimension reduction.
  1. Project data onto lower-dimensional manifold / subspace …
  2. … under preservation of the relevant structure
- Rationale for using principal components:

  $$\text{Variance} = \text{Information}$$
- But do we have

  $$\text{Variance} = \text{Information relevant to us ???}$$
- Principal components and manifolds to a good job for objective (1). Objective (2) is not at all guaranteed.
- Example microarray data: Main source of variability usually some sort of contamination.

$\rightsquigarrow$ Maybe there are better projection directions.

Overview of possible projection direction (not exhaustive)

Comparison of different sequences of directions γ_j to extract $\mathbf{t}_j = \mathbf{X}\gamma_j$

Principal components

$$\gamma_j^{PC} = \underset{\|\gamma\|=1, \gamma' \mathbf{C} \gamma_\nu^{PC}}{\arg\max} \operatorname{Var}(\mathbf{X}\gamma).$$

Regularisation. Use of the manifold structure. No use of the response $\mathbf{y}$.

PLS

$$\gamma_j^{PLS} = \underset{\|\gamma\|=1, \operatorname{corr}^2(\mathbf{X}\gamma_\nu, \mathbf{X}\gamma)=0}{\arg\max} \operatorname{Cov}(\mathbf{y}, \mathbf{X}\gamma) = \underset{\|\gamma\|=1, \operatorname{corr}^2(\mathbf{X}\gamma_\nu, \mathbf{X}\gamma)=0}{\arg\max} \operatorname{corr}^2(\mathbf{y}, \mathbf{X}\gamma) \cdot \operatorname{var}(\mathbf{X}\gamma)$$

Regularisation. Use of the manifold structure. Use of the response $\mathbf{y}$.

Least-squares regression

$$\gamma_1^{LS} = \underset{\gamma \in \mathbb{R}^p, \|\gamma\|=1}{\arg\max} \operatorname{corr}^2(\mathbf{y}, \mathbf{X}\gamma).$$

No regularisation. No use of the manifold structure. Use of the response $\mathbf{y}$.

Overview of possible projection direction (not exhaustive)

Comparison of different sequences of directions γ_j to extract $\mathbf{t}_j = \mathbf{X}\boldsymbol{\gamma}_j$

Principal components

$$\boldsymbol{\gamma}_j^{PC} = \underset{\|\boldsymbol{\gamma}\|=1, \boldsymbol{\gamma}'\mathbf{C}\boldsymbol{\gamma}_\nu^{PC}}{\arg\max} \operatorname{Var}(\mathbf{X}\boldsymbol{\gamma}).$$

Regularisation. Use of the manifold structure. No use of the response $\mathbf{y}$.

PLS

$$\gamma_j^{PLS} = \underset{\|\gamma\|=1, \operatorname{corr}^2(\mathbf{X}\gamma_\nu, \mathbf{X}\gamma)=0}{\arg\max} \operatorname{Cov}(\mathbf{y}, \mathbf{X}\gamma) = \underset{\|\gamma\|=1, \operatorname{corr}^2(\mathbf{X}\gamma_\nu, \mathbf{X}\gamma)=0}{\arg\max} \operatorname{corr}^2(\mathbf{y}, \mathbf{X}\gamma) \cdot \operatorname{var}(\mathbf{X}\gamma)$$

Regularisation. Use of the manifold structure. Use of the response $\mathbf{y}$.

Least-squares regression

$$\gamma_1^{LS} = \underset{\gamma \in \mathbb{R}^p, \|\gamma\|=1}{\arg\max} \operatorname{corr}^2(\mathbf{y}, \mathbf{X}\gamma).$$

No regularisation. No use of the manifold structure. Use of the response $\mathbf{y}$.

Overview of possible projection direction (not exhaustive)

Comparison of different sequences of directions γ_j to extract $\mathbf{t}_j = \mathbf{X}\gamma_j$

Principal components

$$\gamma_j^{PC} = \operatorname*{arg\,max}_{\|\gamma\|=1, \gamma' \mathbf{C} \gamma_\nu^{PC}} \mathrm{Var}(\mathbf{X}\gamma).$$

Regularisation. Use of the manifold structure. No use of the response **y**.

PLS

$$\gamma_j^{PLS} = \operatorname*{arg\,max}_{\|\gamma\|=1, \mathrm{corr}^2(\mathbf{X}\gamma_\nu, \mathbf{X}\gamma)=0} \mathrm{Cov}(\mathbf{y}, \mathbf{X}\gamma) = \operatorname*{arg\,max}_{\|\gamma\|=1, \mathrm{corr}^2(\mathbf{X}\gamma_\nu, \mathbf{X}\gamma)=0} \mathrm{corr}^2(\mathbf{y}, \mathbf{X}\gamma) \cdot \mathrm{var}(\mathbf{X}\gamma)$$

Regularisation. Use of the manifold structure. Use of the response **y**.

Least-squares regression

$$\gamma_1^{LS} = \operatorname*{arg\,max}_{\gamma \in \mathbb{R}^p, \|\gamma\|=1} \mathrm{corr}^2(\mathbf{y}, \mathbf{X}\gamma).$$

No regularisation. No use of the manifold structure. Use of the response **y**.

Overview of possible projection direction (not exhaustive)

Comparison of different sequences of directions γ_j to extract $\mathbf{t}_j = \mathbf{X}\gamma_j$

Principal components

$$\gamma_j^{PC} = \underset{\|\gamma\|=1, \gamma' \mathbf{C} \gamma_\nu^{PC}}{\arg\max} \operatorname{Var}(\mathbf{X}\gamma).$$

Regularisation. Use of the manifold structure. No use of the response **y**.

PLS

$$\gamma_j^{PLS} = \underset{\|\gamma\|=1, \operatorname{corr}^2(\mathbf{X}\gamma_\nu, \mathbf{X}\gamma)=0}{\arg\max} \operatorname{Cov}(\mathbf{y}, \mathbf{X}\gamma) = \underset{\|\gamma\|=1, \operatorname{corr}^2(\mathbf{X}\gamma_\nu, \mathbf{X}\gamma)=0}{\arg\max} \operatorname{corr}^2(\mathbf{y}, \mathbf{X}\gamma) \cdot \operatorname{var}(\mathbf{X}\gamma)$$

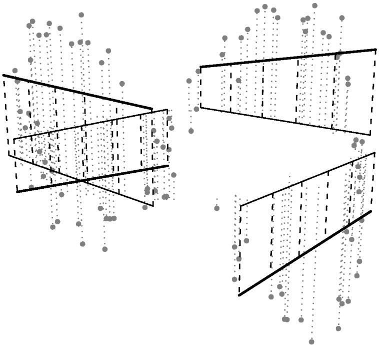
Regularisation. Use of the manifold structure. Use of the response **y**.

Least-squares regression

$$\gamma_1^{LS} = \underset{\gamma \in \mathbb{R}^p, \|\gamma\|=1}{\arg\max} \operatorname{corr}^2(\mathbf{y}, \mathbf{X}\gamma).$$

No regularisation. No use of the manifold structure. Use of the response **y**.

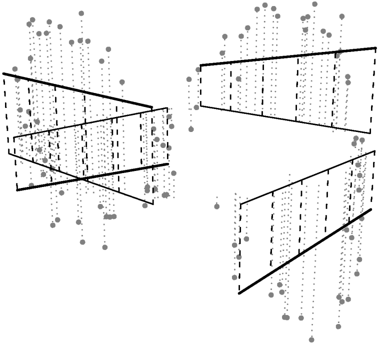
An illustrative example: sine wave on a circle in $\mathbb{R}^2$



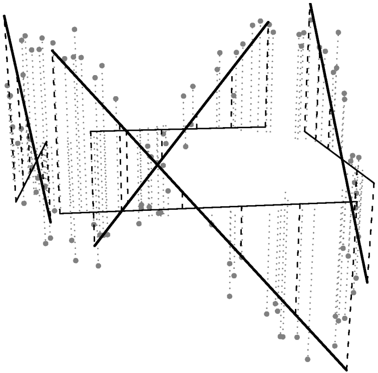
using principal components

using PLS directions

An illustrative example: sine wave on a circle in $\mathbb{R}^2$



using principal components



using PLS directions

Some more details on PLS

- PLS was first proposed in psychometrics
- Stone & Brooks (1990) showed the important property that the projections maximise covariance between $\mathbf{X}$ and $\mathbf{y}$ (holds for the original algorithm only if $\mathbf{y} \in \mathbb{R}$)
- Many different PLS methods with some different degree of equivalence

SIMPLS: Objective

$\mathbf{X} \in \mathbb{R}^{n \times p}$ and $\mathbf{Y} \in \mathbb{R}^{n \times q}$. Extract scores $\mathbf{t}_j := \mathbf{X}\mathbf{w}_j$ and $\mathbf{u}_j := \mathbf{Y}\mathbf{v}_j$ such that

1. Orthogonal $\mathbf{t}_j$: $\mathbf{t}_j'\mathbf{t}_k = 0$ for $j \neq k$.
2. Normalised weights: $\|\mathbf{w}_j\| = \|\mathbf{v}_j\| = 1$.
3. Maximal covariance: $\mathrm{cov}(\mathbf{t}_j, \mathbf{u}_j) = \mathbf{w}_j'\mathrm{cov}(\mathbf{X}, \mathbf{Y})\mathbf{v}_j \xrightarrow{!} \max.$

Some more details on PLS

- PLS was first proposed in psychometrics
- Stone & Brooks (1990) showed the important property that the projections maximise covariance between $\mathbf{X}$ and $\mathbf{y}$ (holds for the original algorithm only if $\mathbf{y} \in \mathbb{R}$)
- Many different PLS methods with some different degree of equivalence

SIMPLS: Objective

$\mathbf{X} \in \mathbb{R}^{n \times p}$ and $\mathbf{Y} \in \mathbb{R}^{n \times q}$. Extract scores $\mathbf{t}_j := \mathbf{X}\mathbf{w}_j$ and $\mathbf{u}_j := \mathbf{Y}\mathbf{v}_j$ such that

1. Orthogonal $\mathbf{t}_j$: $\mathbf{t}_j'\mathbf{t}_k = 0$ for $j \neq k$.
2. Normalised weights: $\|\mathbf{w}_j\| = \|\mathbf{v}_j\| = 1$.
3. Maximal covariance: $\operatorname{cov}(\mathbf{t}_j, \mathbf{u}_j) = \mathbf{w}_j'\operatorname{cov}(\mathbf{X}, \mathbf{Y})\mathbf{v}_j \xrightarrow{!} \max$.

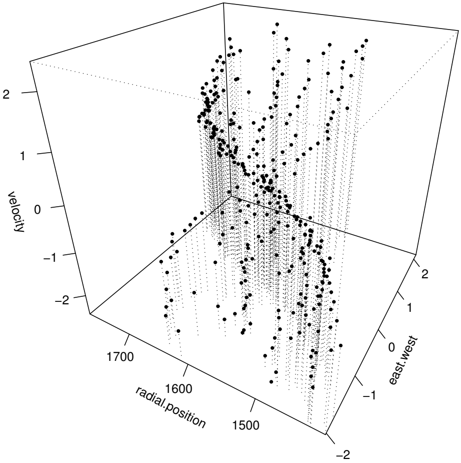
Some more details on PLS (ctd.)

The SIMPLS algorithm

```
1. Set $\mathbf{S}_0 := \mathbf{S} = \mathbf{X}'\mathbf{Y}$.
2. For $j = 1 \ldots, r$:
    i. Compute $\mathbf{w}_j$ (first left singular value) and $\mathbf{v}_j$ (first right singular value) from an SVD on $\mathbf{S}_{j-1}$.
    ii. Compute the scores $\mathbf{t}_j := \mathbf{X}\mathbf{w}_j$.
    iii. Compute the loadings $\mathbf{p}_j := \frac{\mathbf{X}'\mathbf{t}_j}{\mathbf{t}_j'\mathbf{t}_j}$.
    iv. Set $\mathbf{S}_j := \mathbf{S} - \mathbf{P}_j(\mathbf{P}_j'\mathbf{P}_j)^{-1}\mathbf{P}_j'\mathbf{S}$.
3. Set $\mathbf{B}_r := (\mathbf{W}_r\mathbf{W}_r')\mathbf{X}'\mathbf{Y}$.
```

Another example from astronomy

Objective: Predict radial velocity of a galaxy given its east/west position and its radial position



Another example from astronomy (ctd.)

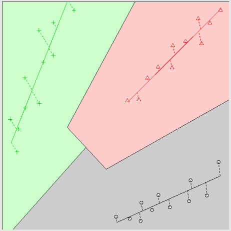
	Training set		Test set	
	L_2 error	(sd)	L_2 error	(sd)
Using principal components	1642.00	(578.4)	1758.44	(615.5)
Using PLS directions	511.42	(79.3)	577.05	(101.9)
MARS	2965.64	(334.7)	3738.76	(494.7)
GAM	3027.14	(321.6)	3554.26	(385.5)
PPR	2207.73	(622.0)	3317.94	(820.7)

(Data set split into a training set of 162 observations and a test set of 161 observations.)

Comparison with CARTs

PLS projection trees

- Partitioning implied by projections onto line segments
- Use of structure in the covariates



⇝ low variance, high bias (ideal weak learner)

CARTs

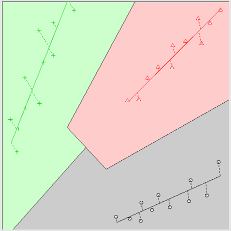
- Partitioning implied by cuts parallel to the axes
- No use of structure in the covariates

⇝ high variance, low bias (needs to be heavily shrunken to be a weak learner)

Comparison with CARTs

PLS projection trees

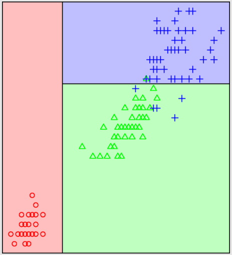
- Partitioning implied by projections onto line segments
- Use of structure in the covariates



⇝ low variance, high bias (ideal weak learner)

CARTs

- Partitioning implied by cuts parallel to the axes
- No use of structure in the covariates



⇝ high variance, low bias (needs to be heavily shrunken to be a weak learner)

Boosting: Idea

- Aggregates *weak learners* to form a powerful “ensemble” (reducing the bias).
- Weak learner: Method with low variance but high bias (“primitive method”)
- Essentially an additive model where the model is fitted several times using changing weights.
- Can be seen as some sort of coordinate descent in a function space.
- Empirically known to be rather resistant against overfitting (can be interpreted as some sort of large margin method)
- Usually shrunken stumps (usually multiplied by a factor $< 10^{-3}$).
- Are PLS projection trees better weak learners?

L_2 boost algorithm

```
1. Fix a maximal number of iterations $h_{max}$.
2. Set $\hat{F}^{(0)} \equiv 0$.
3. Iterate for $h = 1, \ldots, h_{max}$:
    i. Compute the current residual $\varepsilon_i^{(h)} := y_i - \hat{F}^{(h-1)}(\mathbf{x}_i)$.
    ii. Compute estimator $\hat{f}^{(h)}(\mathbf{x}_i)$ using the current weights $\boldsymbol{\epsilon}^{(h)}$ as regressand.
    iii. Set $\hat{F}^{(h)}(\mathbf{x}) := \hat{F}^{(h-1)}(\mathbf{x}) + \hat{f}^{(h)}(\mathbf{x})$.
```

Simulated Example

50 observations with 10 covariates $x_{i,1}, \ldots ; x_{i,10} \sim U(0,1)$ and $(\varepsilon_i \sim N(0, 0.2^2))$

$$y_i = \sum_{j=1}^{5} x_{i,2j-1} \cdot x_{i,2j} \cdot \sin(x_{ij}) + \varepsilon_i$$

	Training set		Test set	
	L_2 error	(sd)	L_2 error	(sd)
Boosted PLS trees	0.87393	(0.195)	1.33154	(0.185)
Boosted stumps	0.63020	(0.147)	1.69423	(0.189)
MARS	0.78294	(0.303)	1.99869	(1.037)
GAM	0.76840	(0.204)	1.66810	(0.441)

Simulated Example

50 observations with 10 covariates $x_{i,1}, \ldots ; x_{i,10} \sim U(0,1)$ and $(\varepsilon_i \sim N(0, 0.2^2))$

$$y_i = \sum_{j=1}^{5} x_{i,2j-1} \cdot x_{i,2j} \cdot \sin(x_{ij}) + \varepsilon_i$$

	Training set		Test set	
	L_2 error	(sd)	L_2 error	(sd)
Boosted PLS trees	0.87393	(0.195)	1.33154	(0.185)
Boosted stumps	0.63020	(0.147)	1.69423	(0.189)
MARS	0.78294	(0.303)	1.99869	(1.037)
GAM	0.76840	(0.204)	1.66810	(0.441)

Summary

- Presented a simple method to approximate principal manifolds by hyperplane segments.
- Proposed alternative directions to the principal components for supervised settings (namely the PLS direction)
- Leads to a “projection tree” algorithm
- Hopefully serves as an inspiration for methods combining regularisation using principal manifolds and supervised learning.

Thank you.