Prediction Error of Estimators for High Dimensional Linear Regression

Adarsh Prasad*, Arun Sai Suggala

CMU

adarshp@andrew.cmu.edu, asuggala@andrew.cmu.edu

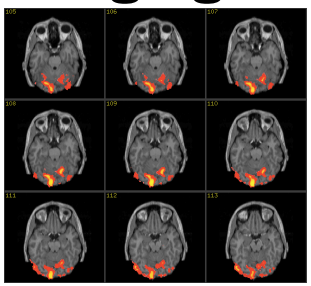
May 2, 2017

Overview

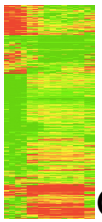
1. Background
   - Motivation
   - Setup.
2. Methods.
   - Subset Selection.
   - LASSO
   - IHT
3. Conclusion

Motivation[1]

- **Imaging**

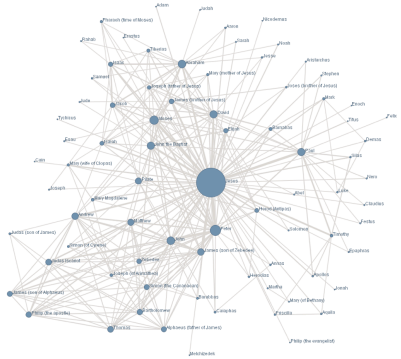


- **Gene (Microarray) Experiments)**



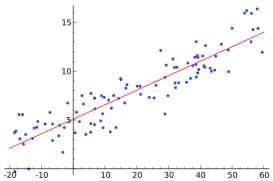
Costly Experiments

- **Social Networks**



Billions of Nodes

- **Linear Regression**

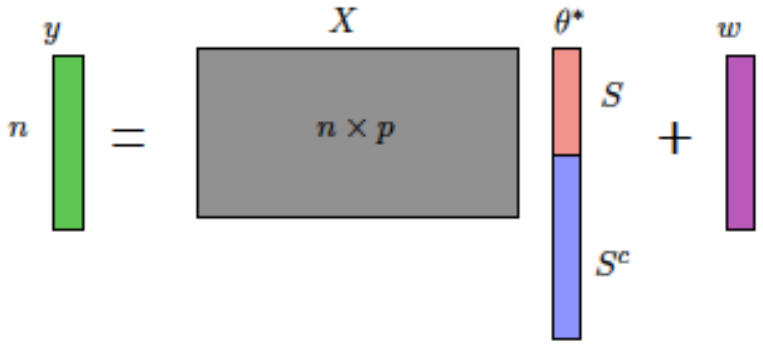


Source: Wikipedia

Modelling Investment risk, Spending, Demand given market conditions.

[1] Slide courtesy: Pradeep Ravikumar

Sparse Linear Regression[2]



$\|\theta^*\|_0 = |\{j \in \{1, \ldots, p\} : \theta^*_j \neq 0\}|$ is small

Estimate a sparse linear model:

$$\min_{\theta} \|y - X\theta\|_2^2$$
$$\text{s.t. } \|\theta\|_0 \leq k.$$

ℓ_0 constrained linear regression!

[2]Slide courtesy: Pradeep Ravikumar

Sparse Linear Regression: Evaluation Metric

- **In-sample Prediction error.**

$$\mathcal{E}(\theta) = \frac{1}{n} \|X(\theta - \theta^*)\|_2^2$$

Sparse Linear Regression: Evaluation Metric

- **In-sample Prediction error.**

$$\mathcal{E}(\theta) = \frac{1}{n} \|X(\theta - \theta^*)\|_2^2$$

- Estimator $\widehat{\theta}$ has **fast** rate (modulo log factors), if:

$$\mathcal{E}(\widehat{\theta}) = O\left(\frac{s}{n}\right)$$

- Estimator $\widehat{\theta}$ has **slow** rate (modulo log factors), if:

$$\mathcal{E}(\widehat{\theta}) = O\left(\frac{s}{\sqrt{n}}\right)$$

Section 2

Methods.

Subset Selection.

Formulation

$$\widehat{\theta}_{\text{Subset}} \in \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \frac{1}{2n} \|y - X\beta\|_2^2 \quad \text{s.t. } \|\beta\|_0 \leq s, \tag{1}$$

Subset Selection.

Formulation

$$\widehat{\theta}_{\text{Subset}} \in \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \frac{1}{2n} \|y - X\beta\|_2^2 \quad \text{s.t. } \|\beta\|_0 \leq s, \tag{1}$$

- Raskutti et al. [7] showed that with absolutely no assumptions on X, $\widehat{\theta}_{\text{Subset}}$ gets fast rates:

$$\mathcal{E}(\widehat{\theta}_{\text{Subset}}) \lesssim \frac{s \log(p/s)}{n}$$

- Also, the minimax rate[7], *i.e.* with constant probability:

$$\inf_{\widehat{\theta}} \sup_{\|\theta^*\|_0 \leq s} \mathcal{E}(\widehat{\theta}) \gtrsim \frac{s \log(p/s)}{n},$$

LASSO

Formulation

$$\widehat{\theta}_{\text{LASSO}} \in \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \frac{1}{2n} \|y - X\beta\|_2^2 + \lambda \|\beta\|_1 , \tag{2}$$

LASSO

Formulation

$$\widehat{\theta}_{\text{LASSO}} \in \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \frac{1}{2n} \|y - X\beta\|_2^2 + \lambda \|\beta\|_1 , \tag{2}$$

Slow Rates.

- Assuming only column normalization on X,

$$\|X_j\|_2 \leq \sqrt{n}, \forall j \in [p]$$

- $\mathcal{E}(\widehat{\theta}_{\text{LASSO}}) \lesssim \sqrt{\frac{\log p}{n}} \|\theta^*\|_1$
- Follows from *zero*th-order optimality and concentration of Gaussian maxima.

LASSO

Formulation

$$\widehat{\theta}_{\text{LASSO}} \in \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \frac{1}{2n} \|y - X\beta\|_2^2 + \lambda \|\beta\|_1 , \tag{2}$$

Slow Rates.

- Assuming only column normalization on X,

$$\|X_j\|_2 \leq \sqrt{n}, \forall j \in [p]$$

- $\mathcal{E}(\widehat{\theta}_{\text{LASSO}}) \lesssim \sqrt{\frac{\log p}{n}} \|\theta^*\|_1$
- Follows from *zero*th-order optimality and concentration of Gaussian maxima.

Fast Rates.

- Additionally, assume Restricted Eigenvalue (RE) or Restricted Strong Convexity (RSC) on X
- Control correlation between the columns of the design matrix.
- $\mathcal{E}(\widehat{\theta}_{\text{LASSO}}) \lesssim \frac{s \log p}{n}$

LASSO: Questions

- Can one relax the RE condition and still get the fast rates for LASSO?
- Are there any design matrices for which LASSO cannot achieve fast rates?

LASSO: Answers

- *Correlated columns actually help the prediction error.* [4].

Measure of Correlation

- For any subset $T \subset [p]$: V_T is column span of X_T.
- Let Π_T be the orthogonal projector onto V_T.
- Let ρ_T be the maximal distance between the normalized columns of X and the set V_T *i.e.*

$$\rho_T = n^{-\frac{1}{2}} \max_{j \in [p]} \left\| \left(\mathcal{I}_n - \Pi_T \right) x^j \right\|_2 ,$$

where $\mathcal{I}_n$ is the $n \times n$ identity matrix.

LASSO: Oracle Inequality.

Theorem (Oracle Inequality [4])

Let $T \subset [p]$ be the set of indices and let $\delta > 0$, $\gamma \geq 1$ be constants. Then, if the tuning parameter λ is not smaller than $\gamma\sigma\rho_T\sqrt{2\log(p/\delta)/n}$, then Lasso (2) satisfies

$$\mathcal{E}\left(\widehat{\theta}_{LASSO}\right) + \frac{2(\gamma-1)\lambda}{\gamma}\left\|\widehat{\theta}_{LASSO}\right\|_1 \leq \inf_{\beta\in\mathbb{R}^p}\left\{\mathcal{E}(\beta) + \frac{2(\gamma+1)\lambda}{\gamma}\|\beta\|_1\right\} + \frac{2\sigma^2\left(|T| + 2\log(1/\delta)\right)}{n}, \tag{3}$$

with probability at least $1 - 2\delta$.

Comments.

- Instantiate with $\beta = \theta^*$
- $T_n : \rho_{T_n} \precsim n^{-r}$ for a positive constant $r > 0$, *i.e.* All covariates are very close to this set.

LASSO: Fast Rates.

Corollary (Fast Rate)

For T_n such that $\rho_{T_n} \precsim n^{-r}$ for a positive constant $r > 0$. Then, if the tuning parameter satisfies $\lambda \geq c\sigma\sqrt{\log(p)/n^{2r+1}}$ for a sufficiently large constant $c > 0$, the Lasso (2) satisfies:

$$\mathcal{E}(\widehat{\theta}_{LASSO}) \precsim \max\left(\sqrt{\frac{\log(p)}{n^{2r+1}}}\left\|\theta^*\right\|_1, \frac{|T_n|}{n}\right),$$

with high probability.

Comments.

- If $\rho_{S^*} \precsim n^{-\frac{1}{2}}$, *i.e.* All covariates are within a constant Euclidean distance of the linear space spanned by the relevant covariate.
- Lasso achieves the fast rate s/n upto logarithmic factors, provided $\lambda = O(\frac{\sqrt{\log p}}{n})$

LASSO: Slow Rate.

Lemma (Slow Rate [4])

Let $n \geq 2$ be an integer and let m be the largest integer less than $\sqrt{2n}$, then let the design matrix X be defined as:

$$X \in \mathbb{R}^{n \times 2m} = \sqrt{\frac{n}{2}} \begin{bmatrix} \mathbf{1}_m^T & \mathbf{1}_m^T \\ \mathcal{I}_m & -\mathcal{I}_m \\ \mathbf{0}_{(n-m-1)\times m} & \mathbf{0}_{(n-m-1)\times m} \end{bmatrix}.$$

Let the true regression vector be $\theta^* \in \mathbb{R}^{2m}$ such that $\theta_1^* = \theta_{m+1}^* = 1$ and 0 otherwise. Also, let the noise term ϵ be i.i.d. Rademacher random variables. Then, for any $\lambda > 0$, the prediction error of $\widehat{\theta}_{LASSO}$ satisfies:

$$\mathbf{P}\left(\mathcal{E}(\widehat{\theta}_{LASSO}) \geq \frac{1}{2\sqrt{2n}}\right) \geq \frac{1}{2}$$

with high probability.

LASSO: Summary.

- Fast Rates: Under Orthogonality(RE) and Very high correlation.
- Slow Rates: Under constant correlation.

Iterative Hard Thresholding (IHT)

Subset Selection Formulation

$$\widehat{\theta}_{\text{Subset}} \in \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \frac{1}{2n} \|y - X\beta\|_2^2 \quad \text{s.t. } \|\beta\|_0 \leq s, \tag{4}$$

- IHT performs a projected gradient descent on the ℓ_0 constrained objective (4). It is an iterative algorithm. In iteration t of the algorithm, the current estimate θ of θ is updated as:

$$\theta^+ \leftarrow HT_s(\beta - \frac{\eta}{n} X^T(X\theta - y)),$$

where $HT_s(.)$ is the projection operator onto the space of s sparse vectors and η is the step size.

Iterative Hard Thresholding (IHT)

Subset Selection Formulation

$$\widehat{\theta}_{\text{Subset}} \in \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \frac{1}{2n} \|y - X\beta\|_2^2 \quad \text{s.t. } \|\beta\|_0 \leq s, \tag{4}$$

- IHT performs a projected gradient descent on the ℓ_0 constrained objective (4). It is an iterative algorithm. In iteration t of the algorithm, the current estimate θ of θ is updated as:

$$\theta^+ \leftarrow HT_s(\beta - \frac{\eta}{n} X^T(X\theta - y)),$$

  where $HT_s(.)$ is the projection operator onto the space of s sparse vectors and η is the step size.
- Blumensath and Davies [1] showed that this iterative process converges for appropriately chosen step size. We denote the point of convergence of IHT by $\widehat{\theta}_{\text{IHT}}$.

Iterative Hard Thresholding (IHT)

Theorem (IHT Fast Rates: [6])

Lets suppose the design matrix X has normalized columns and has RSS and RSC parameters given by $L_{2\tilde{s}} = L$ and $\alpha_{2\tilde{s}} = \alpha$ respectively. Let IHT be invoked with sparsity $\tilde{s} \geq 32 \left(\frac{L}{\alpha}\right)^2 s$ and step length $\eta = \frac{1}{2L}$, where s is the sparsity of the true vector θ^*. Then $\widehat{\theta}_{IHT}$, the point of convergence of IHT satisfies:

$$\mathcal{E}(\widehat{\theta}_{IHT}) \leq 4 \left(\frac{L}{\alpha}\right)^2 \frac{\sigma^2 (s + \tilde{s}) \log p}{n}$$

with probability at least $1 - 1/p^c$ for some constant $c > 0$.

Iterative Hard Thresholding (IHT):Proof Sketch

Proof Sketch: First Step.

The proof involves two main steps.

- In the first step we show that IHT converges to a local minimum of :

$$\underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \frac{1}{2n} \|y - X\beta\|_2^2 \quad \text{s.t. } \|\beta\|_0 \leq \tilde{s}, \tag{5}$$

which is also a fixed point of the hard thresholding operator $HT_{\tilde{s}}(.)$.

$$HT_{\tilde{s}}(\widehat{\theta}_{\mathsf{IHT}}) = \widehat{\theta}_{\mathsf{IHT}}.$$
$$\nabla_{\tilde{S}} f(\widehat{\theta}_{\mathsf{IHT}}) = 0,$$
$$\eta \|\nabla_{\tilde{S}^c} f(\widehat{\theta}_{\mathsf{IHT}})\|_\infty \leq \min_{i \in \tilde{S}} |(\widehat{\theta}_{\mathsf{IHT}})_i|,$$

where $\text{supp}(\widehat{\theta}_{\mathsf{IHT}}) = \tilde{S}$.

Iterative Hard Thresholding (IHT):Proof Sketch

Proof Sketch: Second Step.

- Using properties of the fixed point, one can show:

$$f(\widehat{\theta}_{\text{IHT}}) \leq f(\theta^*)$$

- Then using RSC,:

$$f(\theta^*) \leq f(\widehat{\theta}_{\text{IHT}}) + \left\langle \nabla f(\theta^*), \theta^* - \widehat{\theta}_{\text{IHT}} \right\rangle - \frac{\alpha}{2}\|\widehat{\theta}_{\text{IHT}} - \theta^*\|_2^2.$$

- Substituting $f(\widehat{\theta}_{\text{IHT}}) \leq f(\theta^*)$, and using cauchy-schwartz:

$$\|\widehat{\theta}_{\text{IHT}} - \theta^*\|_2 \leq \frac{2\sqrt{s+\tilde{s}}}{\alpha}\|\nabla f(\theta^*)\|_\infty$$

IHT: Remarks

Remark 1

Observe that IHT in the above theorem is run with a relaxed projection step. In each iteration, the projection is performed onto a $\tilde{s}$ sparse set which is larger than s, the sparsity of θ^*. There are results which analyze IHT when the projection is performed onto a s sparse set [2, 3]. However, they require the design matrix X to satisfy RIP conditions.

IHT: Remarks

Remark 1

Observe that IHT in the above theorem is run with a relaxed projection step. In each iteration, the projection is performed onto a $\tilde{s}$ sparse set which is larger than s, the sparsity of θ^*. There are results which analyze IHT when the projection is performed onto a s sparse set [2, 3]. However, they require the design matrix X to satisfy RIP conditions.

Remark 2

It is unclear if IHT can achieve slow rates similar to LASSO, just under column normalization condition.

Conclusion

Table: Summary of known results.

	Column Normed.	**(RE/RSC)**	**High Correlation**
LASSO	Slow Rate	Fast Rate	Fast Rate
IHT	?	Fast Rate	?
Greedy Methods	?	Fast Rate[3]	?

[3] Elenberg et al. [5] have shown how RSC implies a weak-form of submodularity for subset-selection problems, which in turn means constant-approximation performance for FSR, OMP for more general losses.

References

[1] Thomas Blumensath and Mike E Davies. Iterative thresholding for sparse approximations. *Journal of Fourier Analysis and Applications*, 14(5-6):629–654, 2008.

[2] Thomas Blumensath and Mike E Davies. Iterative hard thresholding for compressed sensing. *Applied and computational harmonic analysis*, 27(3):265–274, 2009.

[3] Coralia Cartis and Andrew Thompson. A new and improved quantitative recovery analysis for iterative hard thresholding algorithms in compressed sensing. *IEEE Transactions on Information Theory*, 61(4):2019–2042, 2015.

[4] Arnak S Dalalyan, Mohamed Hebiri, Johannes Lederer, et al. On the prediction performance of the lasso. *Bernoulli*, 23(1): 552–581, 2017.

[5] Ethan R Elenberg, Rajiv Khanna, Alexandros G Dimakis, and Sahand Negahban. Restricted strong convexity implies weak submodularity. *arXiv preprint arXiv:1612.00804*, 2016.

[6] Prateek Jain, Ambuj Tewari, and Purushottam Kar. On iterative hard thresholding methods for high-dimensional m-estimation. In *Advances in Neural Information Processing Systems*, pages 685–693, 2014.

[7] Garvesh Raskutti, Martin J Wainwright, and Bin Yu. Minimax rates of estimation for high-dimensional linear regression over lq-balls. *IEEE transactions on information theory*, 57(10):6976–6994, 2011.

A Gentle Introduction to Kernel PCA in a Landscape of Dimensionality Reduction Techniques

Nic Dalmasso

Carnegie Mellon University

Pittsburgh, May 1, 2017

Setup and Goal

We have $X_1, ..., X_n \in \mathbb{R}^n$ iid, with $X_i \sim P_X \ \forall i$.

Our goal is **dimensionality reduction**. We achieve that by finding a lower dimensional space V_d of dimension d to project the data onto with a projection Π such that the reconstruction error $R(V)$ is minimized:

$$V_d = \arg\min_{V \in \mathcal{V}_d} R(V) = \arg\min_{V \in \mathcal{V}_d} \mathbb{E} \left\| X - \Pi_V(X) \right\|_2^2 \tag{1}$$

Through the minimization of its empirical version:

$$\hat{V}_d = \arg\min_{V \in \mathcal{V}_d} R_n(V) = \arg\min_{V \in \mathcal{V}_d} \frac{1}{n} \sum_{i=1}^{n} \left(X_i - \Pi_V(X_i) \right)^2 \tag{2}$$

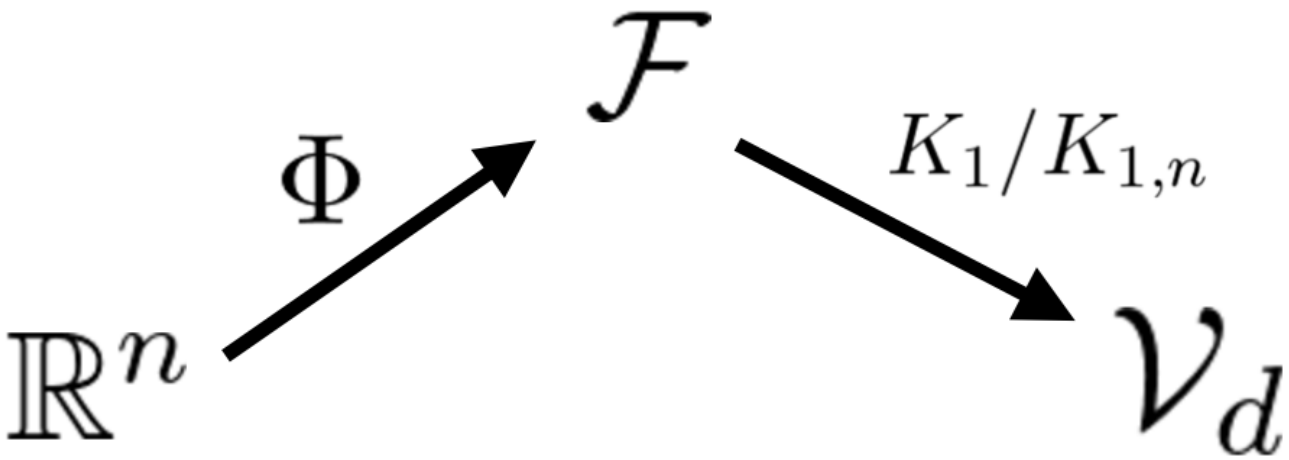
Principal Component Analysis - PCA

PCA achieves that by using the d eigenvectors associated to the largest d eigenvalues of the covariance matrix C (through its empirical version C_n).

Major drawback: it only captures linear structures of the data.

Kernel PCA

Kernel PCA projects the data into a higher dimensional space $\mathcal{F}$ first and then uses PCA on an object called *Kernel Integral Operator* K_1 - through its empirical version $K_{1,n}$.



Major pros: it captures non-linear structures in the data.
Major cons: $\mathcal{F}$ can even be ∞-dimensional. Not clear how to deal with Φ in general.

Theoretical Definitions for KPCA

1. We do not need to worry about Φ. We can work on $\mathcal{F}$ by using a Mercer kernel K, such that $K(x, y) = \langle \Phi(x), \Phi(y) \rangle$. In this case, $\mathcal{F}$ **is a RKHS**.
2. We define K_1 as the **kernel integral operator** on a function $f \in L^2$:

$$(K_1 f)(t) = \int f(x) k(x, t) d\mathbb{P}(x) \tag{3}$$

3. We define $K_2 = \mathbb{E}\left[K_1 \otimes K_1^*\right]$
4. The empirical version of both are easily obtainable:

$$(K_{1,n})_{i,j} = \frac{k(X_i, X_j)}{n} \quad , \quad (K_{2,n})_{i,j} = \frac{k^2(X_i, X_j)}{n} \tag{4}$$

$K_{1,n}$ is usually known as *Gram Matrix*.

Main Result on Reconstruction Error for KPCA

Under reasonable assumptions:

- **A1** $\forall x \in \mathbb{R}^n$ $k(x, \cdot)$ is measurable with respect to $\mathbb{P}$;
- **A2** $\exists M > 0$ such that $k(x, x) \leq M$ a.s. [$\mathbb{P}$];
- **A3** $\exists L > 0$ such that $\sup_{x,y \in \mathbb{R}^n}(k^2(x, x) + k^2(y, y) - 2k^2(x, y)) \leq L^2$.

Theorem (Global Upper Bound for Kernel PCA)

*Let **A1**,**A2**, **A3** hold. Given $\epsilon > 0$, the following holds with probability at least $1 - 3e^{-\epsilon}$:*

$$|R_n(\hat{V}_d) - R(V_d)| \lesssim \sqrt{\frac{d}{n}\mathrm{tr}(K_{2,n})} \tag{5}$$

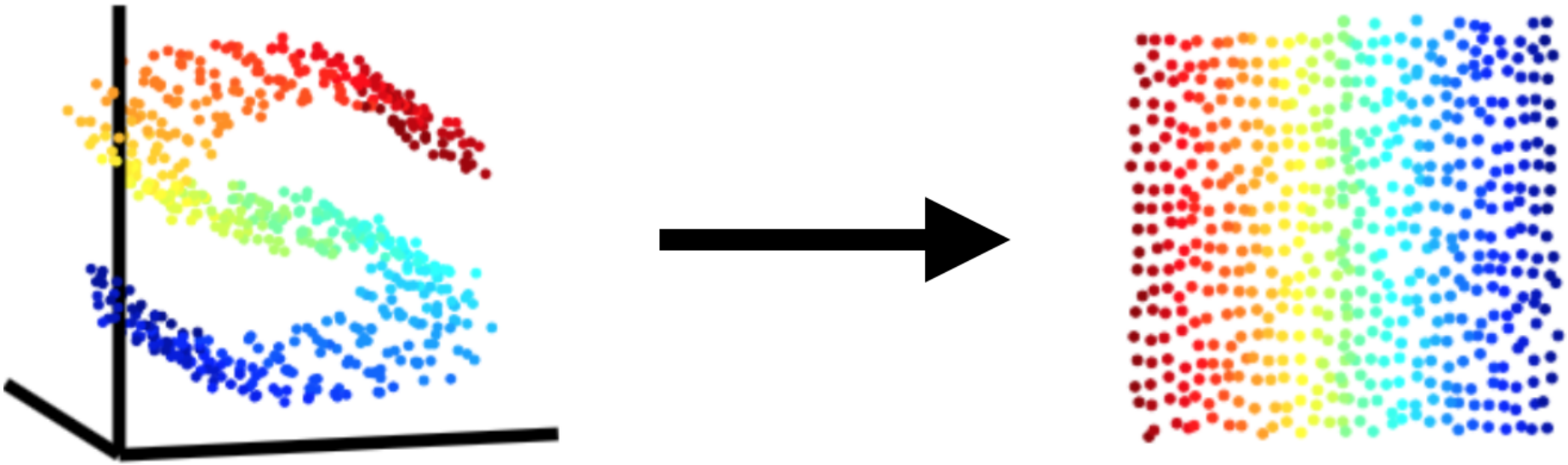
Can we Improve?

Yes, if we put extra assumption of the eigenvalues of K_1.

1. Fixed d.
   With extra assumptions with the d^{th} gap on the eigenvalues of K_1 ($\lambda_{d,K_1} - \lambda_{d+1,K_1}$) the upper bound is same or tighter:
   - $n^{-\frac{1}{2}}$ when eigenvalues of K_1 decay polinomially;
   - n^{-1} when eigenvalues of K_1 decay exponentially.
2. Fixed n.
   It is much harder to improve that bound for an increasing d - possible only with strong assumptions.

ISOMAP

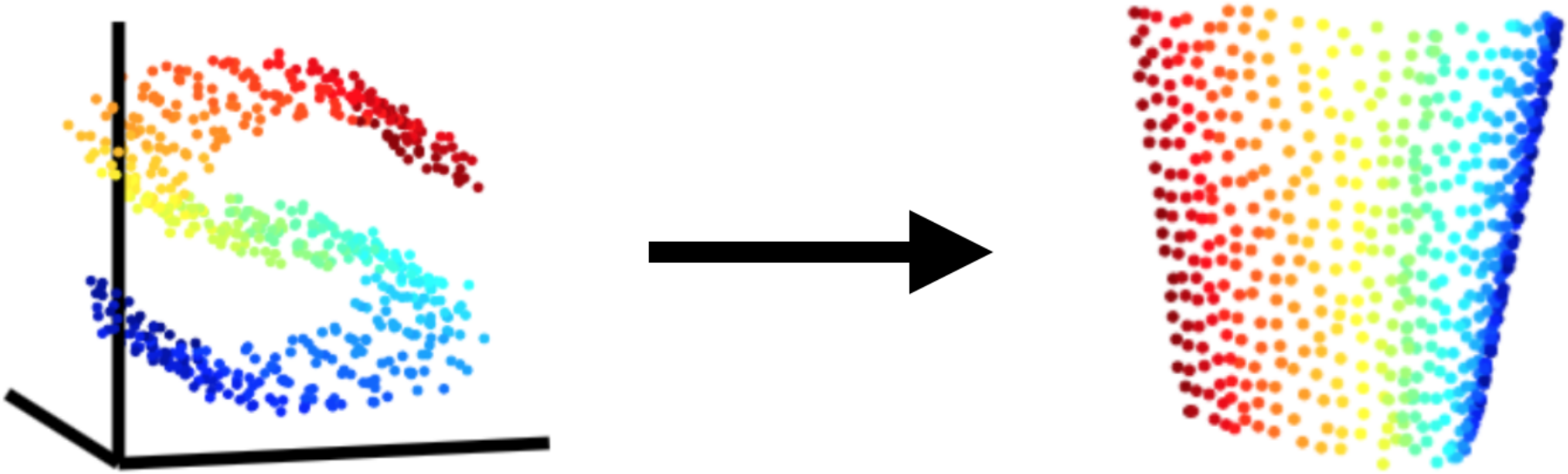
ISOMAP is a dimensionality reduction techniques which builds a graph-based distance between neighbour points in order for the projection to preserve *distances*.



Equivalent to KPCA using a Gram Matrix built from that graph-based distance.

LLE - Locally Linear Embedding

Locally linear embedding performs local linear regression and aims to project local areas preserving *angles*.



Equivalent to KPCA using a Gram Matrix built from the matrix of the coefficients of the local linear regressions.

THANK YOU
FOR YOUR ATTENTION

SDP Relaxation for K-Means Clustering

Chen Dan Yao Liu
Computer Science Department

10-702 Course Project

May 1, 2017

K-Means: Theory vs Practice

Theory:

K-Means is NP-Hard.

Lloyd’s algorithm has exponential worst-case time complexity.

There are algorithms with bounded approximation ratios.

Practice:

Everyone is using K-means.

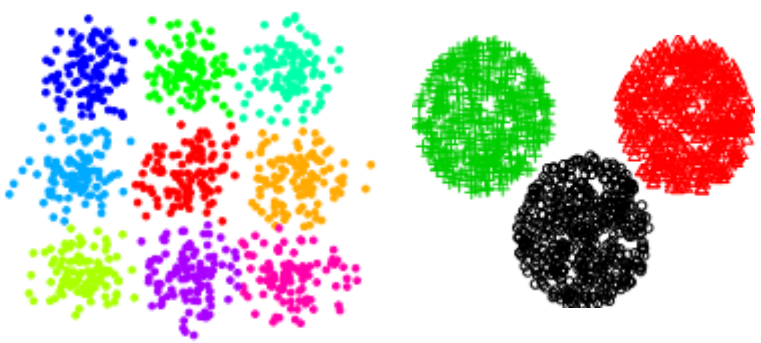
Lloyd’s algorithm converges very fast in practice.

Solutions with constant approximation ratio can be far away from optimal.

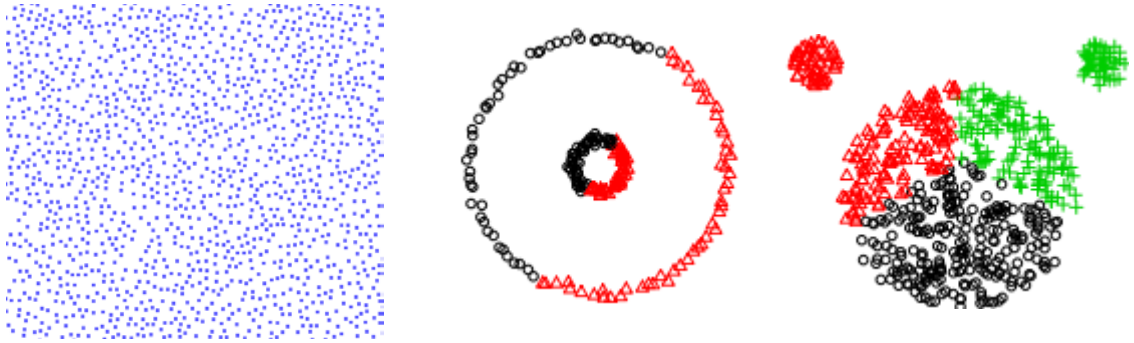
CDNM Thesis

("***C***lustering is ***D***ifficult when it does ***N***ot ***M***atter" ---- Shai Ben-David)

- **Worst-Case** complexity takes many **non-clusterable** instances into consideration.
  - Intuitively, K-means is suitable for data with nearly **ball-shaped** and **balanced** clusters.
  - Requires **non-worst case** performance measurement.



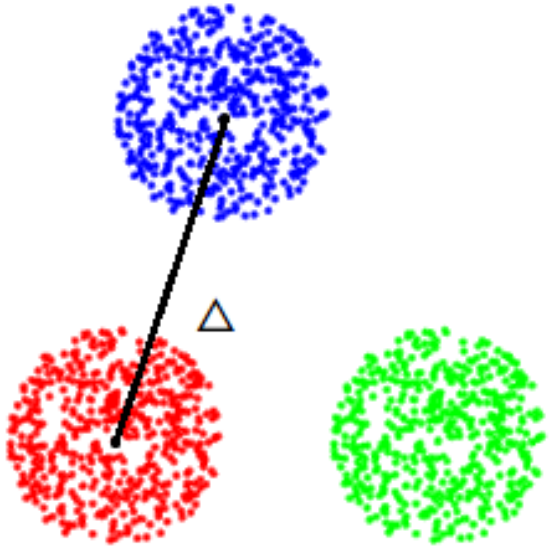
Clusterable Data

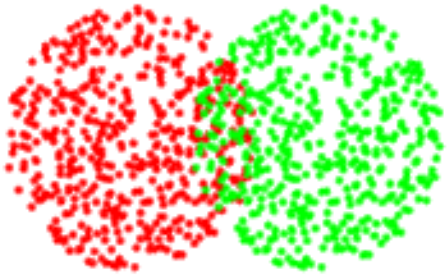


Non-Clusterable Data (for K-means)

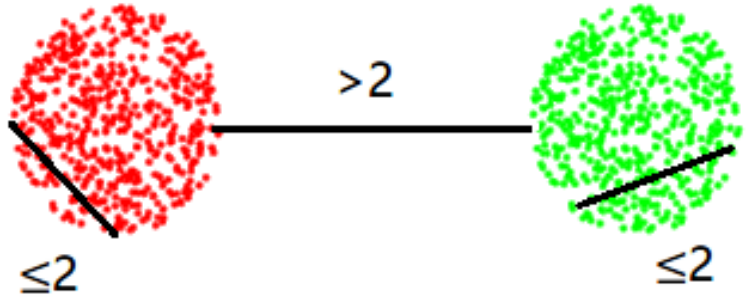
Stochastic Ball Model

- Suppose data is uniformly distributed on K unit balls in R^d.
- The centers $c_1, c_2, \ldots, c_k$ satisfy that $\|c_i - c_j\|_2 > \Delta$.
- For an K-means heuristic, can it recover these K unit balls?
- "Easy" for $\Delta > 4$, impossible for $\Delta < 2$.





$\Delta < 2$



$\Delta > 4$

Stochastic Ball Model

Theorem [ABCKVW'14] Even when Δ is very large, there exist an example which

- Lloyd's Algorithm
- K-means ++
- K-means # (K-means ++ with overseeding)

all fail with high probability.

Semi-Definite Programming relaxation gives recovery guarantee under Stochastic Ball Model!

Semi-Definite Programming (SDP)

$$LP: \quad \text{minimize} \quad c \cdot x$$
$$\text{s.t.} \quad a_i \cdot x = b_i, \quad i = 1, \ldots, m$$
$$x \in \Re^n_+.$$

$$SDP: \quad \text{minimize} \quad C \bullet X$$
$$\text{s.t.} \quad A_i \bullet X = b_i \quad , i = 1, \ldots, m,$$
$$X \succeq 0,$$

- SDP is LP in matrix form with additional semi-definite constraint.
- SDP can be solved in polynomial time.

SDP Relaxation for K-Means

The k-means objective can be formulated as:

$$\sum_{t=1}^{k} \sum_{i \in C_t} ||x_i - \frac{1}{|C_t|} \sum_{j \in C_t} x_j||^2 = \frac{1}{2}\mathrm{Tr}(DX)$$

$$X_{ij} = \begin{cases} 1/|C_t|, \text{if } i, j \text{ belong to } C_t \\ 0, \text{else} \end{cases}$$

Note that the constraint above is combinatorial.
Convex relaxations on these constraints leads to a SDP problem:

$$\begin{aligned} \text{minimize} \quad & \mathrm{Tr}(DX) \\ \text{subject to} \quad & X \geq 0, X \succeq 0, X1 = 1, \mathrm{Tr}(X) = k \end{aligned}$$

Recovery Guarantee for Stochastic Ball Model

Theorem 1 *For stochastic ball model, when* $\Delta > 2\sqrt{2}(1 + 1/\sqrt{d})$*, the SDP relaxation recovers the true clustering.*

Theorem 2 *For stochastic ball model, when* $\Delta > 2 + k^2/d$*, the SDP relaxation recovers the true clustering.*

Theorem 3 *For stochastic ball model, when* $\Delta < 4$*, the LP relaxation can fail to recover the true clustering of the points.*

Recovery Guarantee for Sub-Gaussian Mixtures

Theorem 4 *For high dimensional ($d >> \log n$) isotropic subgaussian mixtures, when $\|\mu_j - \mu_l\|^2 \geq d|\sigma_j^2 - \sigma_l^2| + \Omega(\sqrt{\frac{\log d}{d}})$ for all $1 \leq j < l \leq k$, we have the optimal solution of SDP $\hat{X}$ satisfying $\|\hat{X} - X_0\| = o_P(1)$, where X_0 is the underlying clustering matrix.*

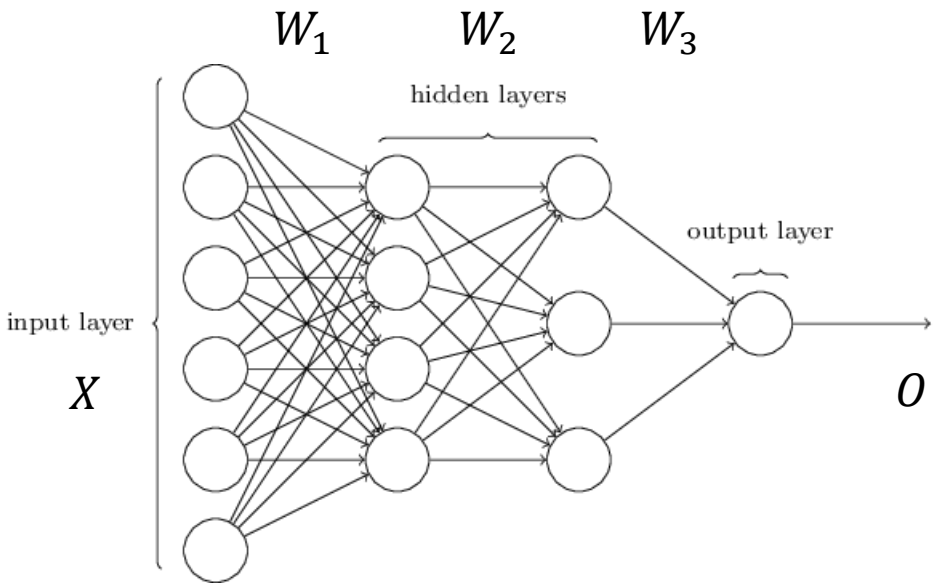
Theorem 5 *For subgaussian mixtures, when $\alpha = O(k)$ and $k = O(d)$, $\Delta^2 = \Omega(\varepsilon^{-1}k^2\alpha\sigma_{\max}^2)$ and $n = \Omega(d + \log(1/\delta))$, we have the optimal solution of SDP $\hat{X}$ satisfying $\|\hat{X} - X_0\|_F^2 = O(\varepsilon)$ w.p. $1 - \delta$.*

Exploring theories on training deep feed-forward neural networks

Presenter: Ermao Cai, Ruizhou Ding

Problem

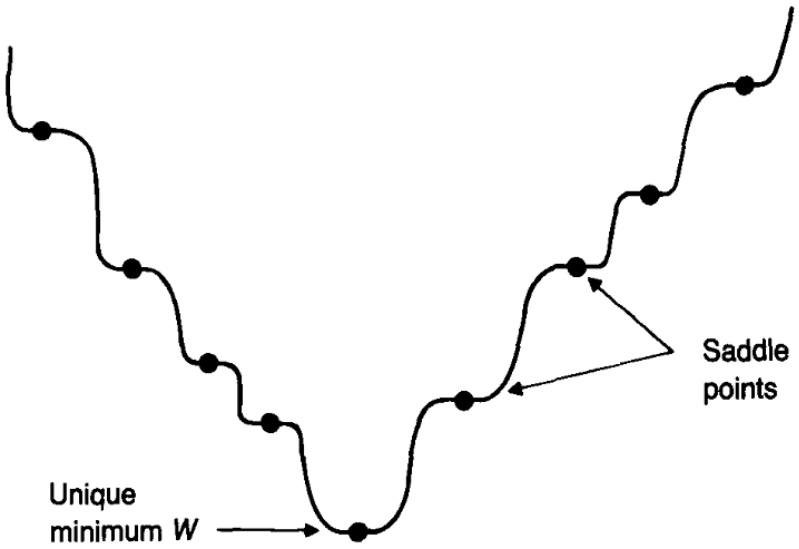
- Target: multilayer neural networks trained on data set $\{X^{(n)}, Y^{(n)}\}_{n=1}^{N}$
  - Output: $O = q\sigma\left(W_H^T \sigma\left(W_{H-1}^T \dots \sigma(W_1^T X)\right)\dots\right)$
    - q: normalization factor
    - σ: activation function
    - W_i: weights of the i-th layer
- Loss function:
  - $L(W) = \frac{1}{N}\sum_{n=1}^{N}\left(O^{(n)} - Y^{(n)}\right)^2$



- Goal: characterize the loss surface $L(W)$
  - Difficult: high-dimension, non-convex

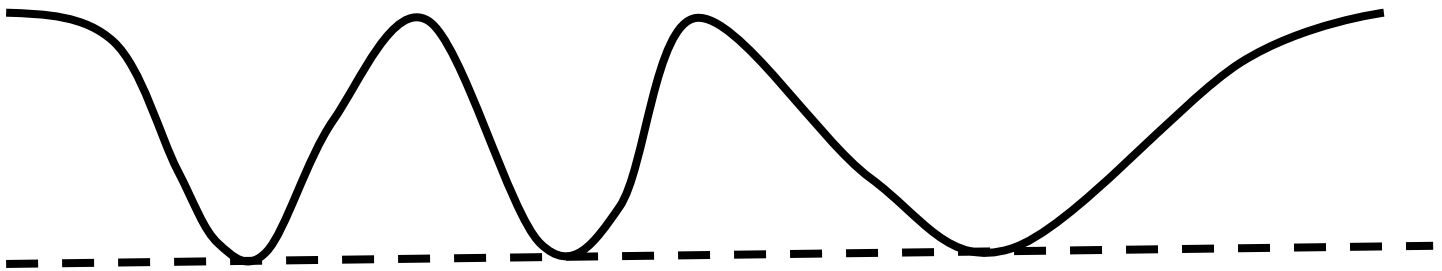
Many saddle points

- Statement 1: for $L(W)$, the ratio of the number of saddle points to local minima increases exponentially with the dimensionality N [Rasmussen and Williams, 2005]



No bad local minima

- Statement 2: under mild over-parameterization, the training error is zero at every differentiable local minimum, for almost every dataset and dropout-like noise realization.



A GUIDED QUEST THROUGH RANDOM FORESTS

BRIEF SPRINT

Robin Dunn

May 2, 2017

Carnegie Mellon University
Department of Statistics

MOTIVATION

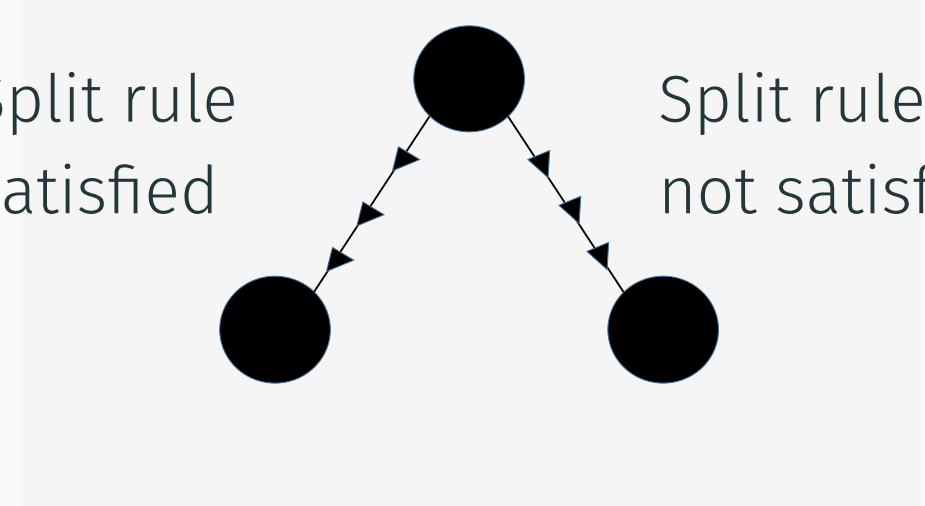
Problem	Random forests . . .
High-dimensional data	work with small subsets of features at a time.
Strong and weak predictors	find strong predictors.
Unsure of functional form	model as a partitioned subspace.
Need accurate results	have strong empirical success.
Want nice theoretical properties	To be continued . . .

HOW TO GROW A TREE

Idea: Aggregate many classification/regression trees.

Grow individual trees by CART method (Breiman [1984]):

- Bootstrapped training set
- Random subset of features for splitting
- Choose split that minimizes "impurity" of new nodes
- Grow until one unique observation per terminal node



CLASSIFICATION TREE THEORY

$h(X, \Theta)$: Tree classifier

$mg(X, Y) = 2P_\Theta(h(X, \Theta) = Y) - 1$

$\widehat{mg}(X, Y) = \frac{1}{K}\sum_{k=1}^{K}(2I(h(X, \Theta_k) = Y) - 1)$

s: $\mathbb{E}_{X,Y} mg(X, Y)$, strength of trees

$\bar{\rho}$: Average correlation between trees

Theorem (Breiman [2001])

$P_{X,Y}(\widehat{mg}(X, Y) < 0) \xrightarrow{a.s.} P_{X,Y}(mg(X, Y) < 0).$

Theorem (Breiman [2001])

Assume $s, \bar{\rho} > 0$. Then $P_{X,Y}(mg(X, Y) < 0) \leq \bar{\rho}(1 - s^2)/s^2$.

REGRESSION TREE THEORY

Biau (2012) analyzed a simplification of regression RFs.

k_n: Number of terminal nodes
$\mathcal{S}$: Set of strong predictors

Variance of individual tree is $O(k_n/n)$. (Devroye, Györfi, Lugosi [1996])

Theorem (Biau [2012]**)**

Variance of forest bounded above by $O\left(\frac{k_n}{n(\log k_n)^{|\mathcal{S}|/2M}}\right)$.

Theorem (Biau [2012]**)**

For optimal k_n, L_2 risk of estimator is $O\left(n^{\frac{-0.75}{|\mathcal{S}| \log 2 + 0.75}}\right)$.

REFERENCES

- Biau, Gérard. “Analysis of a Random Forests Model.” JMLR, 2012.
- Breiman, Leo. “Classification and Regression Trees.” Wadsworth Statistics/Probability Series, 1984.
- Breiman, Leo. “Random Forests.” Machine Learning, 2001.
- Devroye, Luc et al. “A Probabilistic Theory of Pattern Recognition.” Springer-Verlag, 1996.

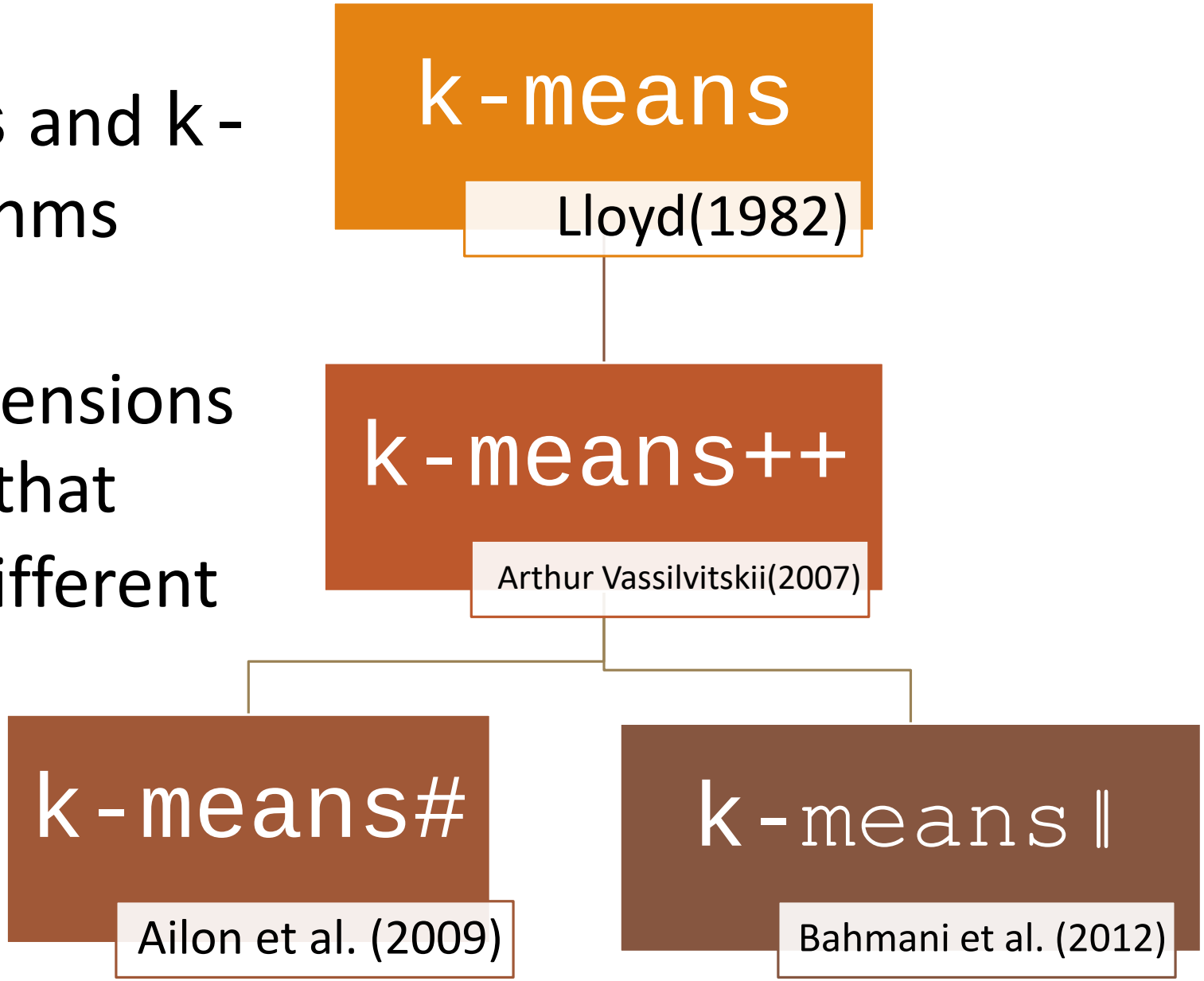
Thank you!

Improving k-means++

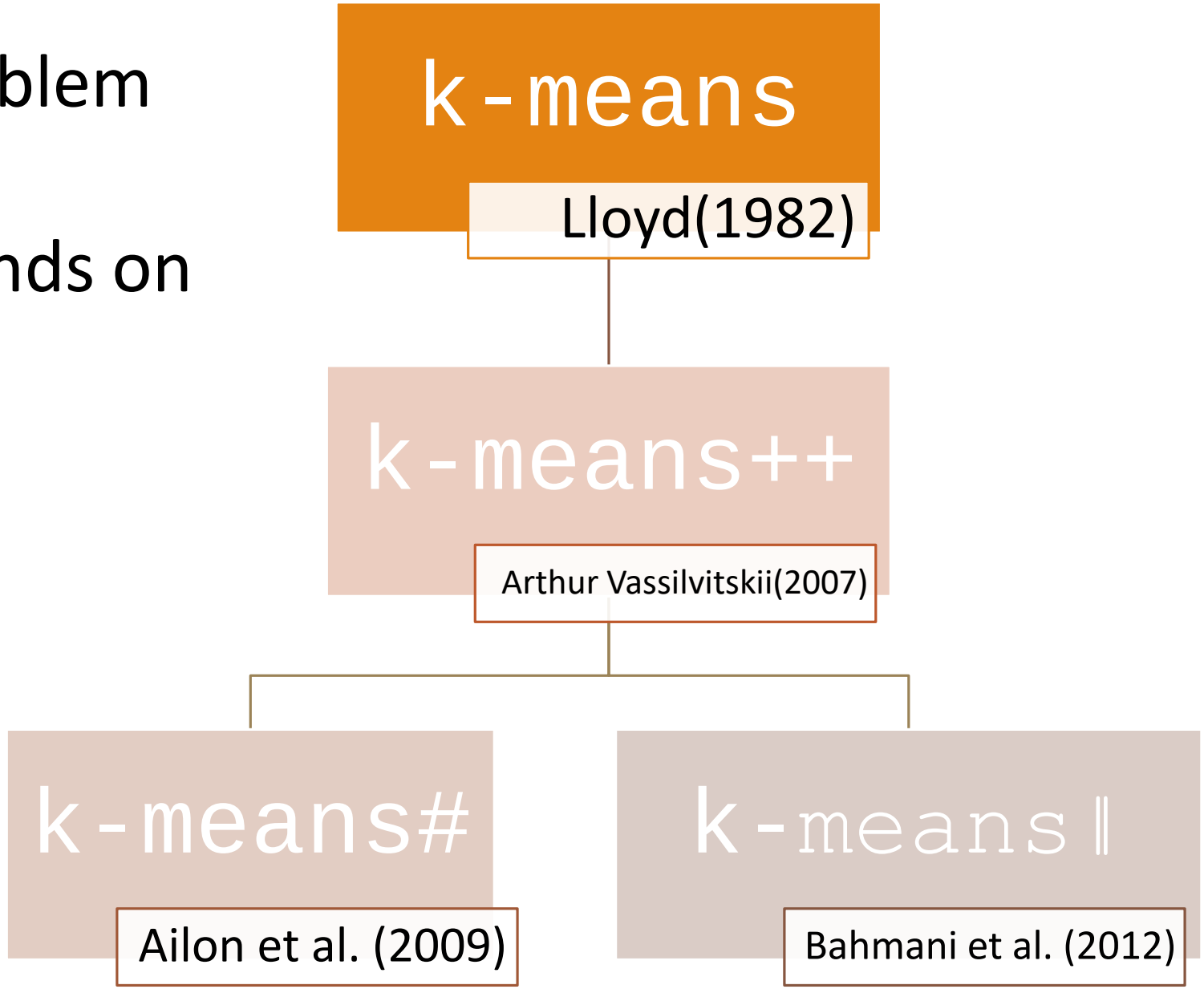
JIAYI LI

TIANYI YANG

- Review k-means and k-means++ algorithms
- Introduce two extensions of k-means++ that improve it from different aspects

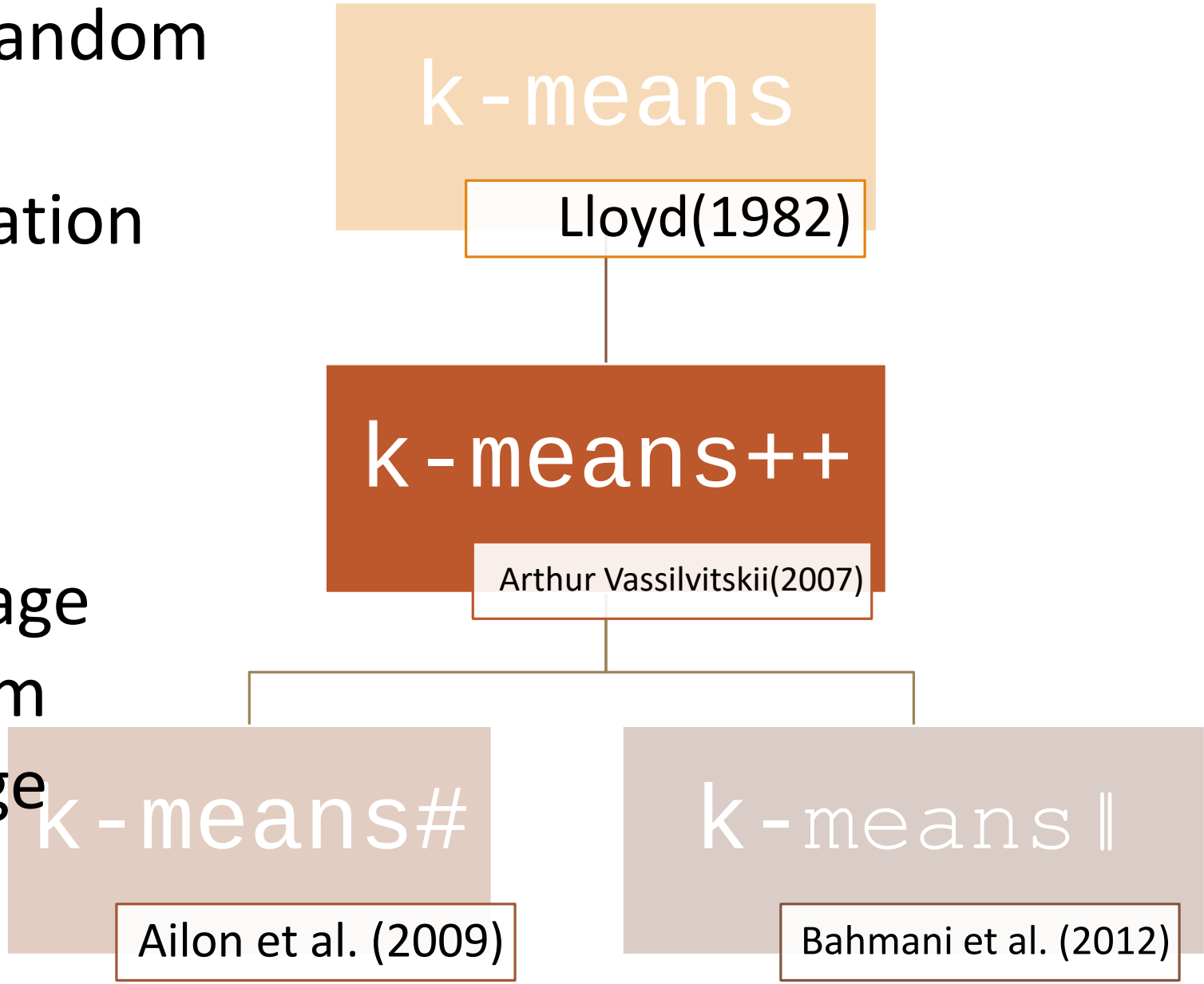


- Solves NP hard problem
- Simple
- Performance depends on initialization

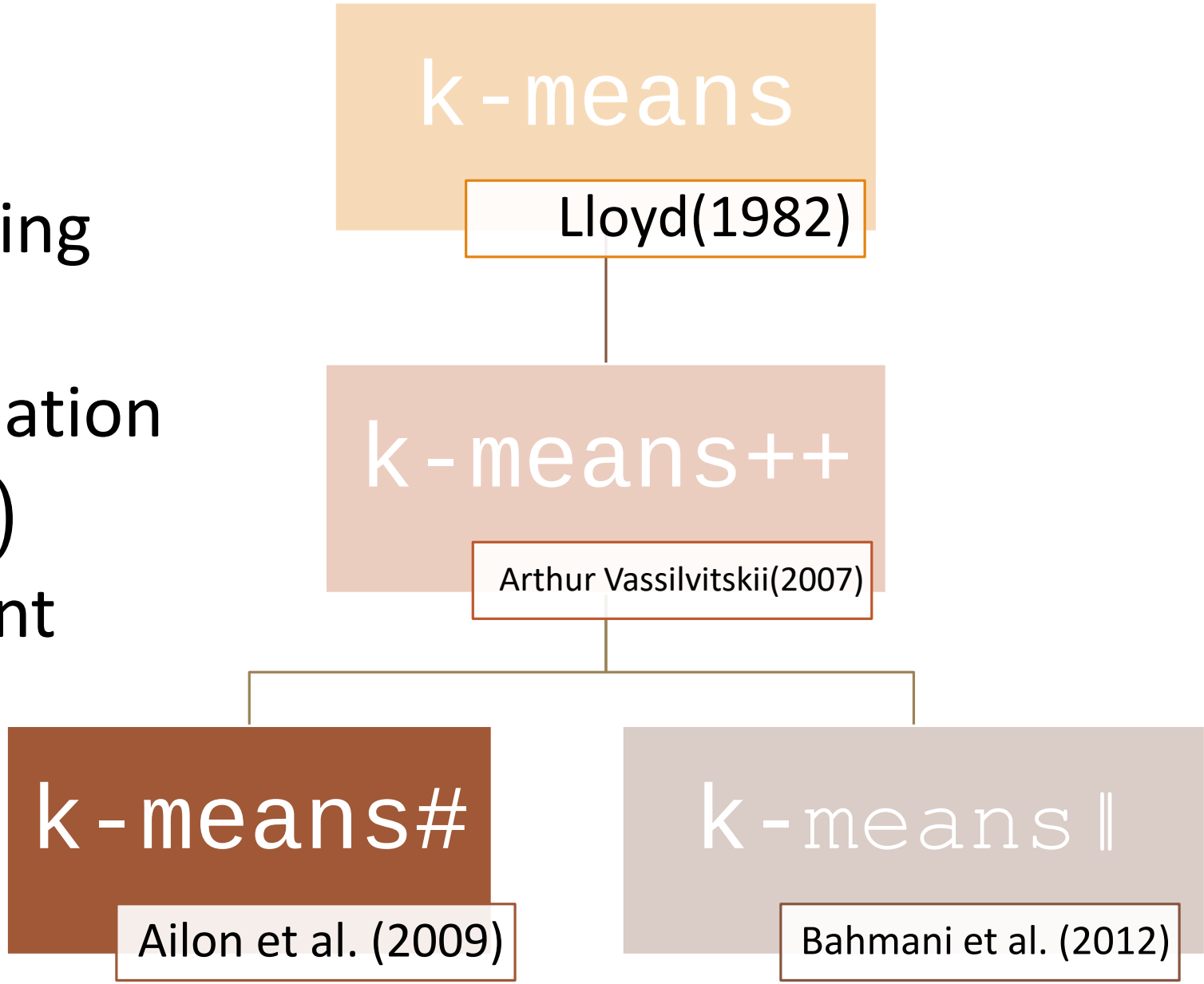


- Provides strategic random initialization
- $O(\log k)$ approximation

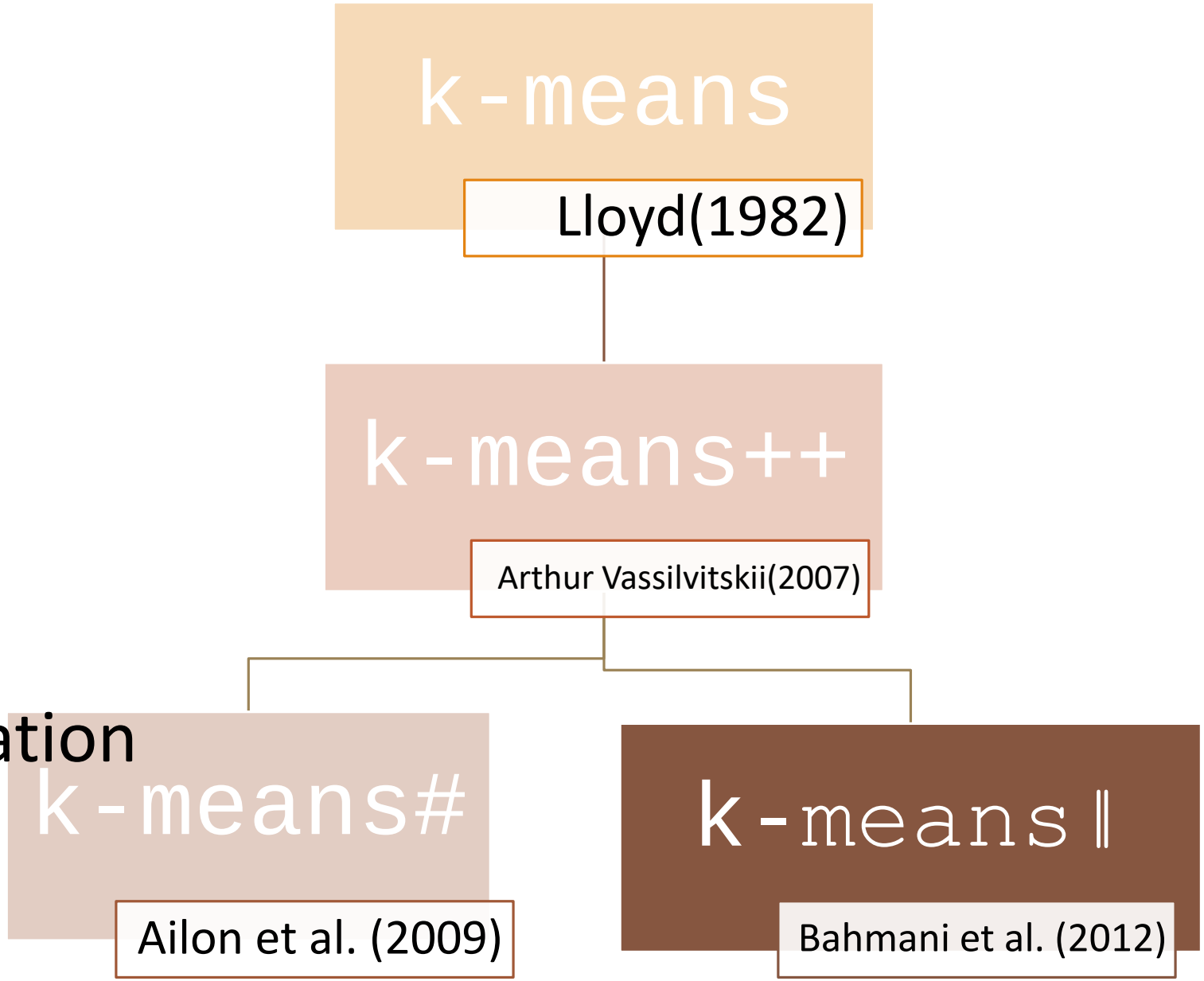
- Requires k passes through the data
  - Needs large storage
- Sequential algorithm
  - Slow when k large



- Single-pass streaming algorithm
- $O(\log k)$ approximation
- $O\left(\sqrt{nk}\log k \log n\right)$ storage requirement



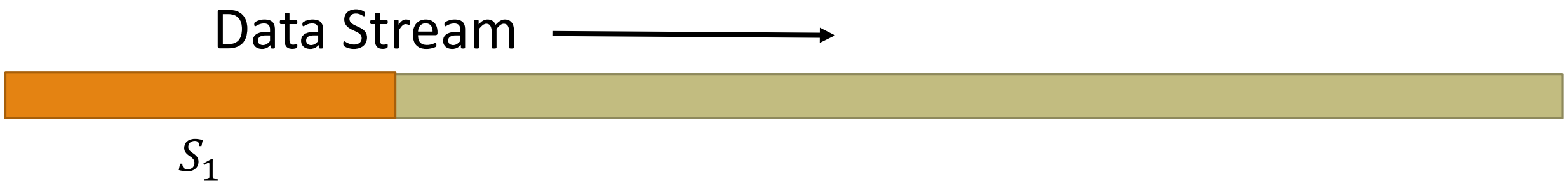
- Easy parallel implementation
- $O(\log n)$ runtime
- $O(\log k)$ approximation



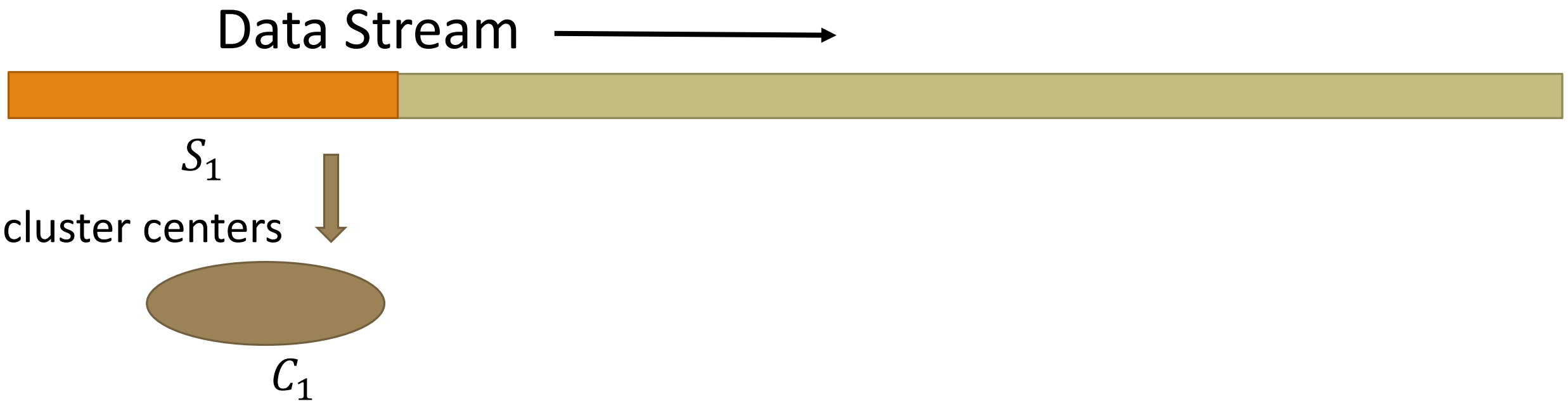
k-means# algorithm

Data Stream 

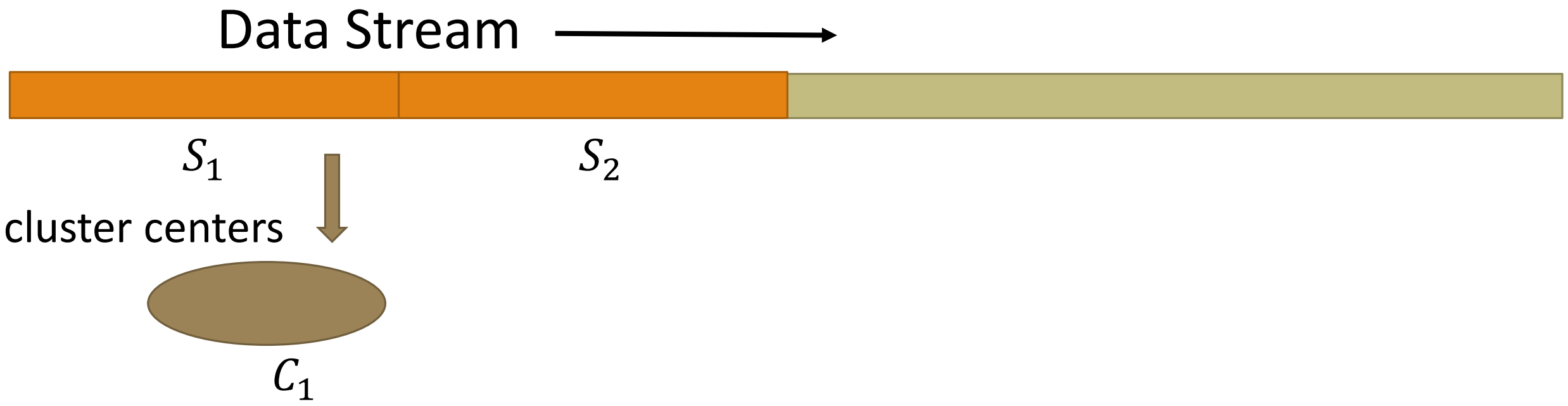
k-means# algorithm



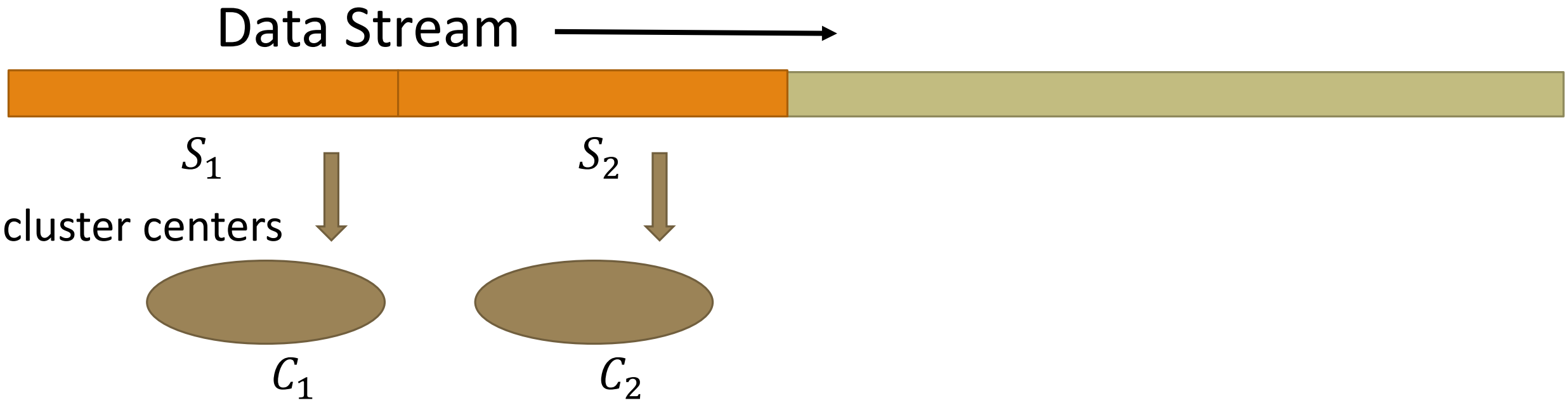
k-means# algorithm



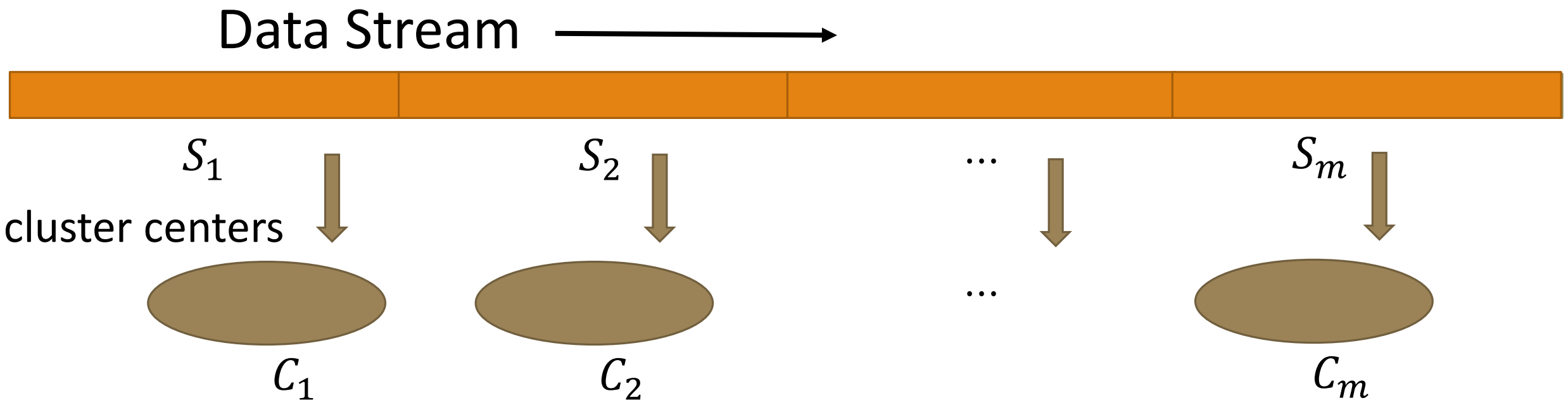
k-means# algorithm



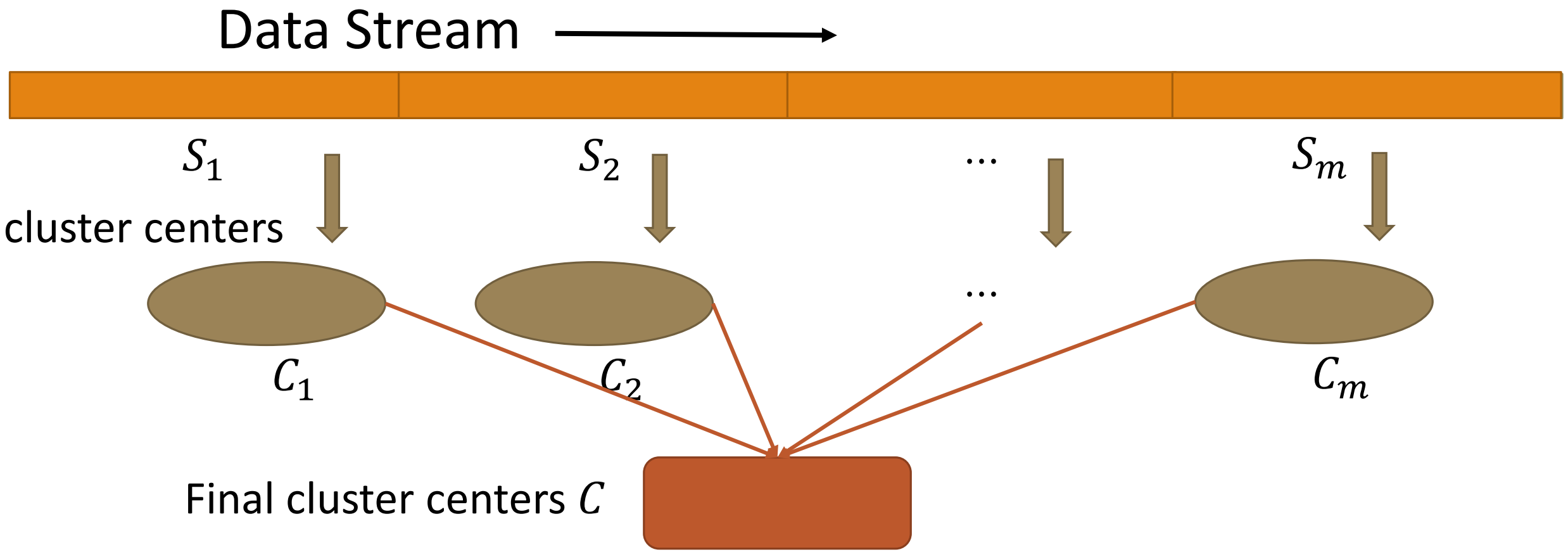
k-means# algorithm



k-means# algorithm

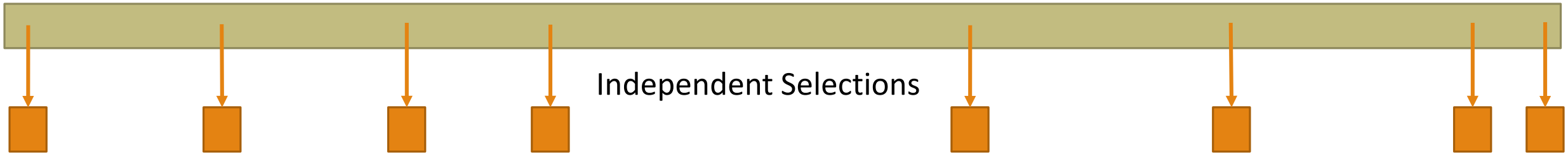


k-means# algorithm

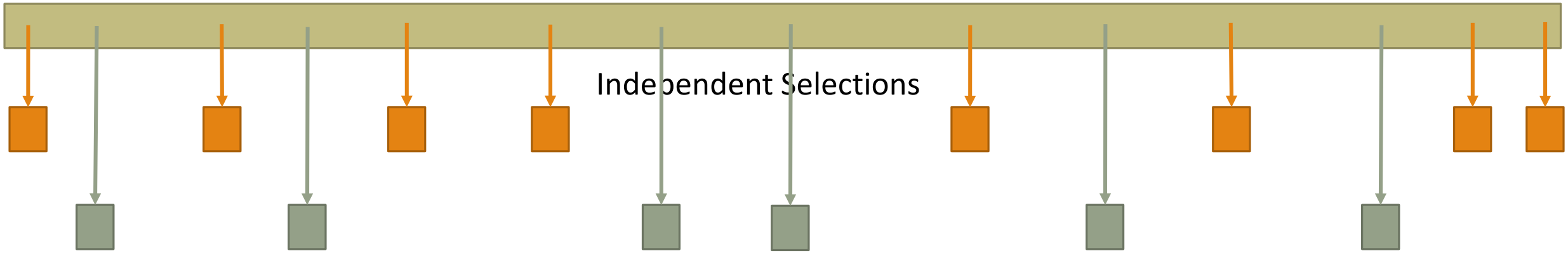


k-means‖ algorithm

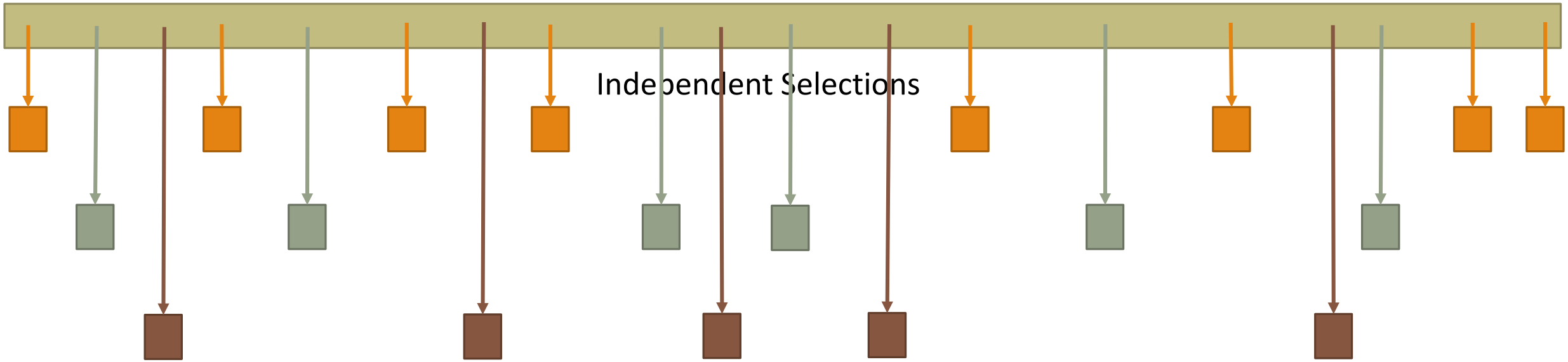
k-means‖ algorithm



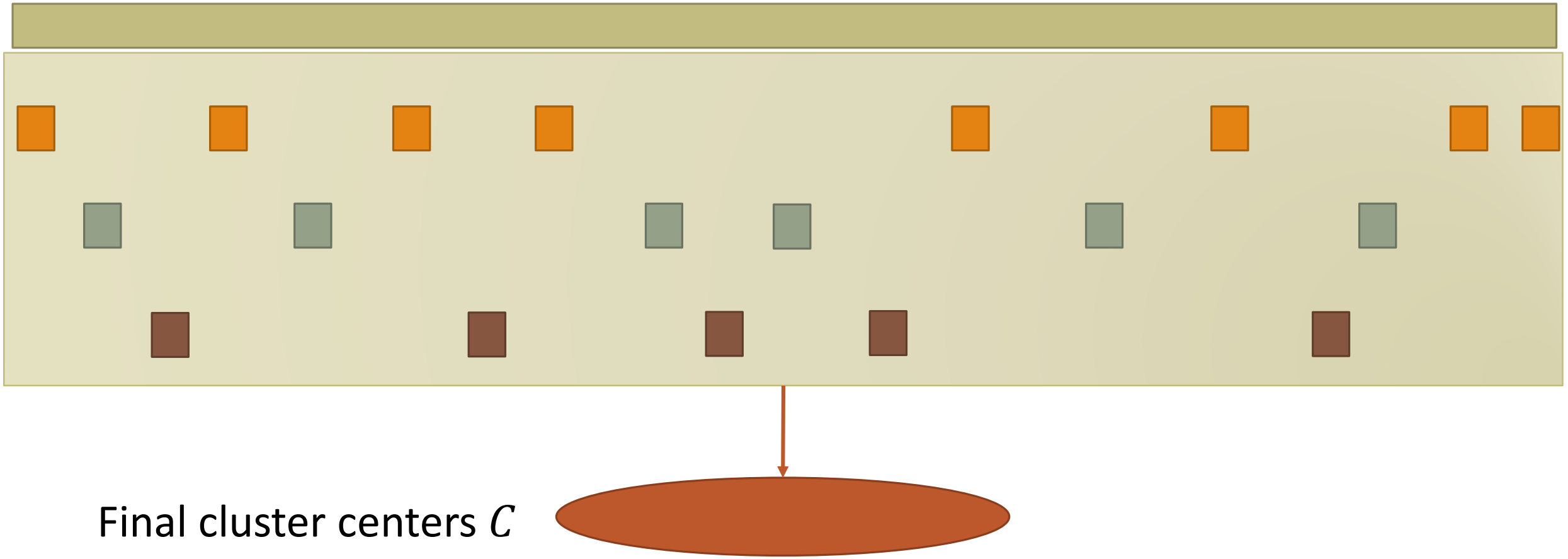
k-means‖ algorithm



k-means‖ algorithm



k-means‖ algorithm



Final cluster centers C

Thank you!

From Metropolis-Hastings and Beyond:

Parameter Inference in Undirected Graphical Models

Boxiang “Shawn” Lyu
Carnegie Mellon University

MCMC in Undirected Graphical Models

- Graphical models explain correlation between covariates: $\mathbb{P}(x_1, x_2, ..., x_n) = \frac{1}{Z} \prod_{c \in \mathcal{C}} \psi_c(\mathbf{x}_c)$
- MCMC generates samples according to the true distribution: $\frac{1}{n} \sum_{i=1}^{n} \phi(x_i) \to \mathbb{E}[\phi(x)]$
- Metropolis-Hastings is a typical MCMC algorithm
- Can be used in higher dimensions
- Don't have to estimate the partition function *Z*

Convergence and Rates of Convergence of Metropolis-Hastings Algorithm

- Proposal distribution *Q*, and accept new sample *y'* with probability $A(y'|y) = \min\left(1, \frac{P(y')Q(y|y')}{P(y)Q(y'|y)}\right)$
- With proper *Q*, converge to true distribution *P* quadratically in total variation distance
- "Burn in" time: roughly $(L/\epsilon)^2$ where L is distance between *Q* and *P*, ϵ the "distance" traveled by *Q* at each step.

- But… what about graph theory?
- Samples rejected: slow down rate of convergence
  - Gibbs sampling: initialize $x_1, x_2, \ldots, x_n$ Sample each x_i on the distribution $P(x_i | x_j : j \neq i)$
- Markov property: $P(x_i | x_j : j \neq i) = P(x_i | N(x_i))$
- Parallel Gibbs sampling: faster MCMC, utilize conditional independence between variables

Discussion

- MCMC is a type of algorithm suitable for parameter inference in graphical models.
    - Metropolis-Hastings algorithm is theoretically guaranteed to converge to true distribution under suitable conditions.
- Utilizing conditional independence relationship between variables allows for parallel MCMC algorithms
- Different perspectives: variational methods. View parameter learning as an optimization problem.

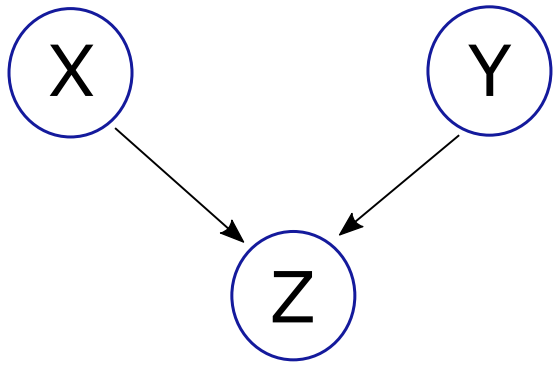
Bayesian Networks

Ciaran Evans
Department of Statistics, CMU

May 2, 2017

BACKGROUND

A Bayesian network represents conditional independence for a set of random variables with a DAG

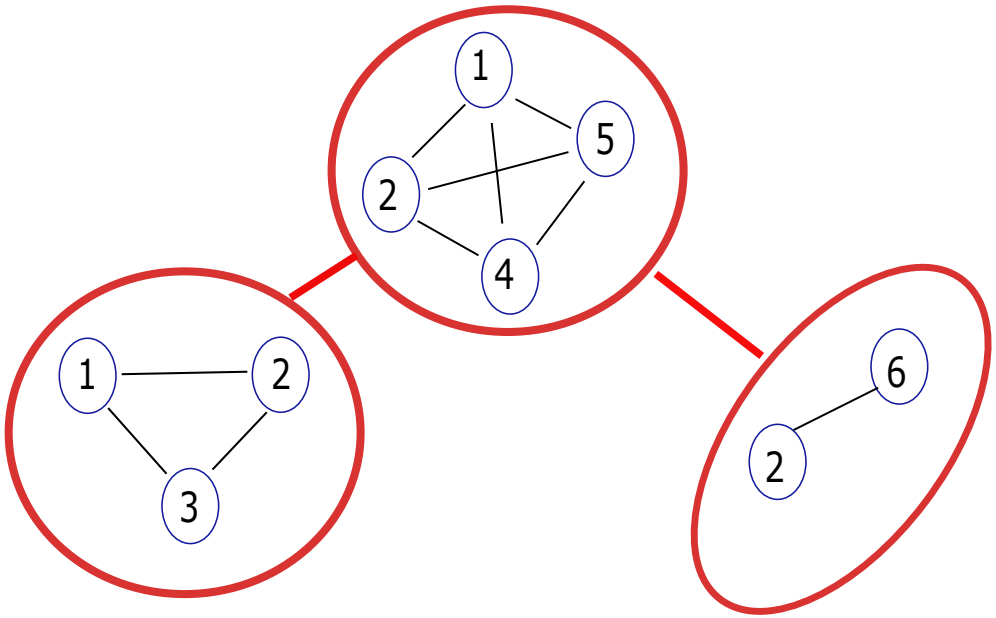


The joint density factors over the DAG:

$$p(x) = \prod_{v \in V} f(x_v | x_{pa(v)})$$

JUNCTION TREE OF CLIQUES

We transform the original graph into a junction tree of cliques



Definition: A tree is a **junction tree** if for any nodes C_1, C_2 their intersection $C_1 \cap C_2$ is contained in each node on the path between them.

Potentials on the Junction Tree

We represent the joint density in terms of potential defined on each node and edge of the junction tree:

$$p(x) = \frac{\prod_{C \in \mathcal{C}} \phi_C(x_C)}{\prod_{S \in \mathcal{S}} \phi_S(x_S)}$$

This representation is updated by **flows**, which modify the potentials

- $\phi{*}_{S_0} = \sum_{C_1 \setminus S_0} \phi_{C_1}$
- $\phi{*}_{C_2} = \phi_{C_2} \left(\frac{\phi{*}_{S_0}}{\phi_{S_0}} \right)$

Marginal Distributions

Important result:

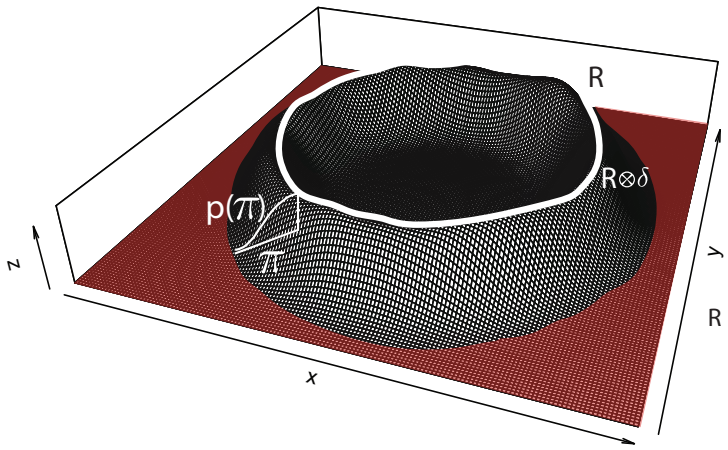
After passing appropriate flows, the final representation of the joint density gives the marginal distribution on each clique set of variables.

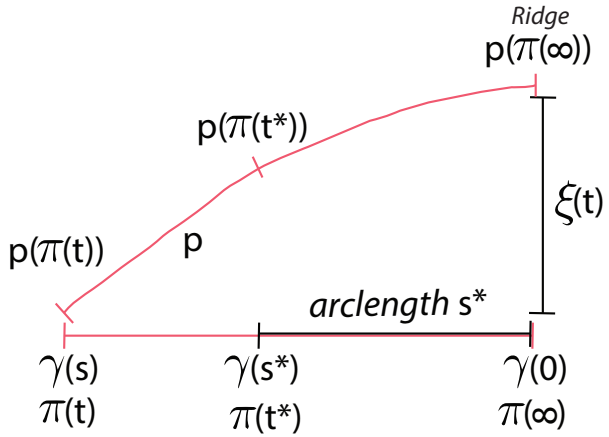
To incorporate prior information, update the initial potentials accordingly and pass flows again.

Thank you!

Density Ridge Estimate

Benjamin LeRoy





Main Theorem

Theorem 5 Let $\hat{R}^* = \hat{R} \cap (R \oplus \delta)$. Assuming we have nice structure for p, $\hat{p}$,with $h \asymp \sqrt{\psi_n}$,

$$\text{Haus}(R, \hat{R}^*) = O_p(\psi_n)$$

where R is true ridge, $\hat{R}$ is ridge for $\hat{p}_n$, ψ_n is related to maximum distance between the evaluation of the gradient, Hessian or Hessian derivative from the two Ridge's density.

Cures for curse of dimensionality in high-dimensional nonparametric regression

Kwangho Kim

Department of Statistics

Curse of dimensionality

- High-dimensional nonparametric regression with *n* samples and *p* regressors suffers from the curse of dimensionality:

  e.g. $R_n^2 \backsim n^{-\frac{4}{4+p}}$ (*Gyorfi et al. 2012*)

- Cures
  - Structural assumption: **Sparse additive model**
  - Dimensionality reduction: **high-dimensional feature screening**

1) High-dimensional feature screening

- Recently researchers proposed several nonparametric, model-free feature screening methods for high-dimensional data
  (e.g. *Fan et al. 2011; Zhu et al. 2011; Li et al. 2012*)

- *Comminges & Dalalyan (2012)* showed minimax rate of nonparametric support recovery cannot be smaller than $d \cdot log(\frac{p}{d})/n$
  where $d = card(true\ support)$

2) Sparse additive models

- Sparsity assumption

  S: f depends on at most d predictors

- Sparse additivity assumption

  SA: $f = \sum_{s=1}^{k} f_s$ where f_s depends on at most d predictors

- *Key Thm. Yang & Tokdar (2015)* provide tight minimax rates under SA

Sparsity vs. Sparse additivity

- Many widely used regression methods (Lasso, Dantzig selector, etc.) rely on S

- Under S, minimax remains smallest as long as

$$d = o(logn) = o(loglogp)$$

Extreme Sparsity!!

Sparsity vs. Sparse additivity

- Under SA, minimax remains smallest as long as

$$d = o\left((logp)^{\theta}\right), \qquad \theta \; not \; far \; from \; 1$$

- Ideally, when combined proper feature screening,

$$d = o(n)$$

➡ Can have much larger number of predictors!

Implementation

- I show with slight modification,

sparse additive model from *Raskutti et al. (2012)*

+

feature screening methods from *Li et al. (2012)*

can achieve previous result with $P \to 1$

Targeted Maximum Likelihood Estimation

What is it and why?

Semi-parametric Causal Inference

Assumptions are bad

Inference is good

Will running keep you alive longer?

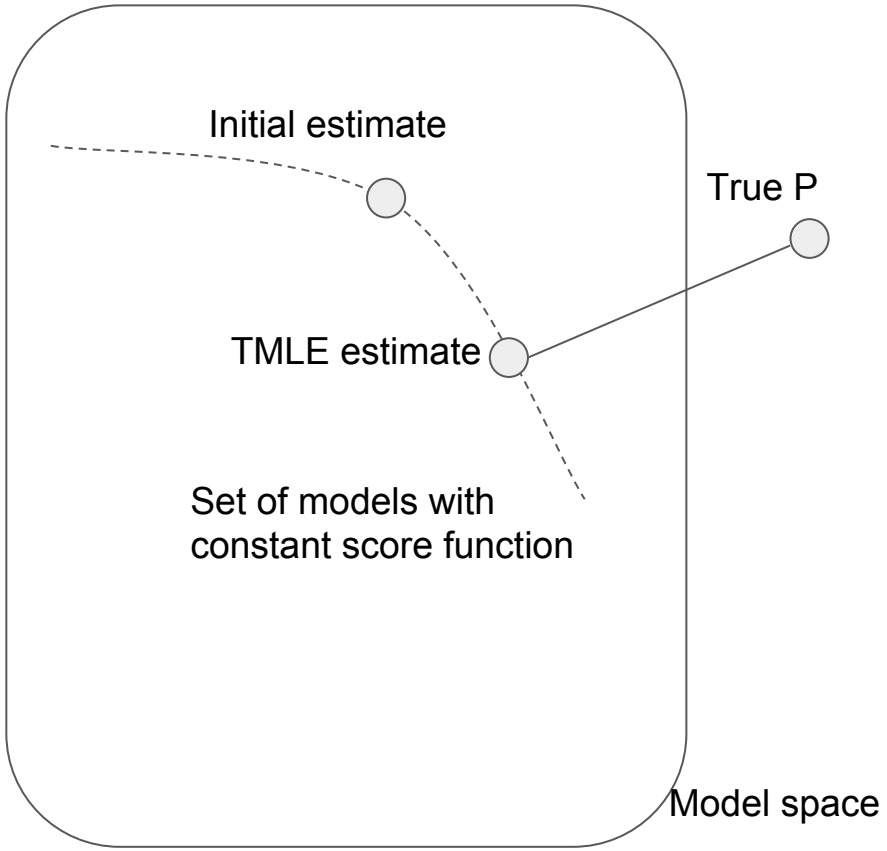
Will chocolate keep you alive longer?

How to target your MLE

- Pick a parameter
- Find its influence function
- Super learn it
- Move it around

$$p_n^k = p_n^{k-1}(\epsilon(P_n|p_n^{k-1}))$$

$$\psi_{TMLE} = \lim_{k \to \infty} \Psi(P_n^k)$$



Is it worth it?

- TMLE has a lot of good qualities: consistency, asymptotic linearity, asymptotic efficiency in a wide range of settings
- If you have a lot of nuisance functions, it may get better rates than a plug-in
- Robust to positivity assumption violations
- Respects the model's constraints
- Generally not very different
- Issues with coverage

Thank you!

Review: Minimax theory with computational constraints.

Minshi Peng and Shengming Luo

Department of Statistics
CMU

May 2, 2017

Classical minimax risk

Classical minimax risk is defined by:

$$R_n = \inf_{\hat{\theta} \in \Theta} \sup_{P \in \mathcal{P}} \mathbb{E}\Big[w(d(\hat{\theta}, \theta(P)))\Big], \tag{1}$$

However, there's **no constraints** on the choice of estimators, including those with prohibitive computational costs.

An example: Sparse linear regression

Classical minimax results:

Theorem

$$\inf_{\hat{\theta}} \sup_{\theta \in B_0(k)} \mathbb{E}[\frac{1}{n}\|X(\hat{\theta} - \theta^*)\|^2] \geq \frac{\sigma^2 k \log(d)}{n}.$$

The matching upper bound could be derived by the following ℓ_0 based estimator: ([NP-hard])

$$\hat{\theta}_{\ell_0} := \arg \min_{\theta \in B_0(k)} \|y - X\theta\|_2^2. \tag{2}$$

Computational efficient minimax rate

However, Lasso estimator $\hat{\theta}_{\ell_1}$ ([Poly-time]) gives an upper bound:

$$\sup_{\theta \in B_0(k)} \mathbb{E}[\frac{1}{n}\|X(\hat{\theta} - \theta*)\|^2] \leq \frac{1}{\gamma^2(X)} \frac{\sigma^2 k \log(d)}{n}, \quad (3)$$

where $\gamma(X) \leq 1$ is the RE constant.

Computational efficient minimax lower bound:

Theorem

$$\inf_{\theta_{poly}} \sup_{\theta \in B_0(k)} \mathbb{E}[\frac{1}{n}\|X(\hat{\theta} - \theta^*)\|^2] \geq \frac{C}{\gamma^2(X)} \frac{\sigma^2 k^{1-\delta} \log(d)}{n}$$

Key technique

There's no general framework for deriving such a computational efficient minimax rate. What people usually do at present:

- Relate original problem to a problem known to be NP-hard. (e.g. planted clique, 3-set cover problem).
- Use contradiction: if an efficient method existed for the original problem, it would lead to an efficient solution to the NP-hard problem.

Analysis of Spectral Clustering

Guokun Lai & Jingzhou Liu

May 2, 2017

Spectral Clustering Algorithm

(1) Construct a similarity graph from the original similarities between data points, and denote this weighted adjacency matrix as W.

(2) Compute the unnormalized Laplacian $L = D - W$ (normalized Laplacian $L = D^{-1}(D - W)$).

(3) Compute the first k eigenvectors $u_1, ..., u_k$ of L, composing them into $U \in \mathbb{R}^{n \times k}$.

(4) Then for every vertex i we have a k-dimension vector $u_i \in \mathbb{R}^k$, i.e. the i-th row of matrix U.

(5) Run k-means algorithm on $u_1, .., u_n$ to get the clustering for the vertices in the graph.

Relationship to the Graph Cut

The Spectral Clustering Algorithm with unnormalized Laplacian minimizes Ratio Cut approximately.

$$\underset{A_1,\cdots,A_k}{\text{minimize}} \quad \sum_{i=1}^{k} \frac{cut(A_i, \overline{A}_i)}{|A_i|} \tag{1}$$

The Spectral Clustering Algorithm with normalized Laplacian minimizes Normalized Cut approximately.

$$\underset{A_1,\cdots,A_k}{\text{minimize}} \quad \sum_{i=1}^{k} \frac{cut(A_i, \overline{A}_i)}{vol(A_i)} \tag{2}$$

Where $vol(A_i) = \sum_{j \in A_i} d_j$, and d_j is the degree of the jth node.

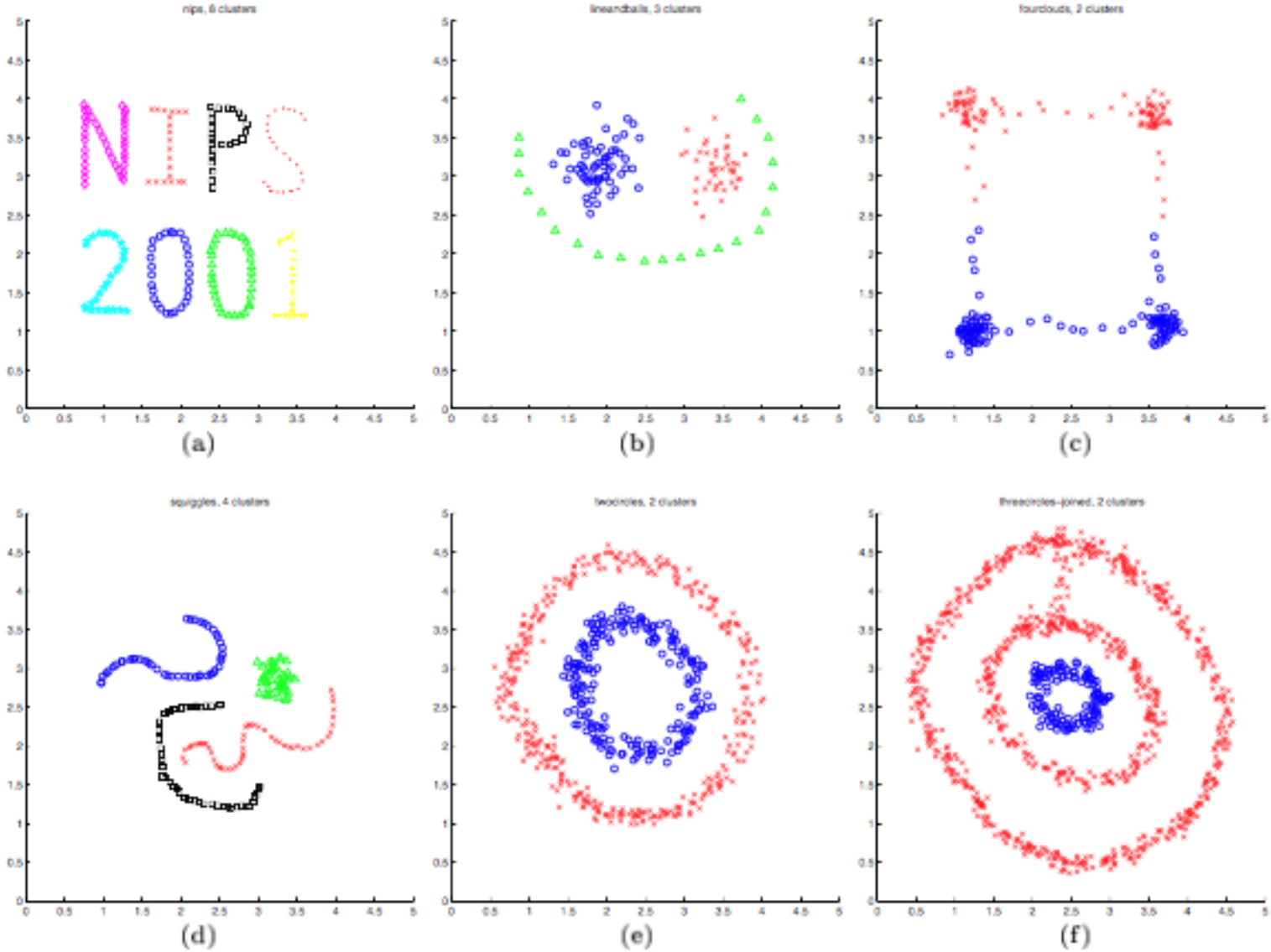
Relationship to the Random Walk

Define the Markov transition matrix as $M = D^{-1}W$. It has eigenvalue λ_i and eigenvector v_i. The random walk process converges to the unique equilibrium distribution π_s. Then we have

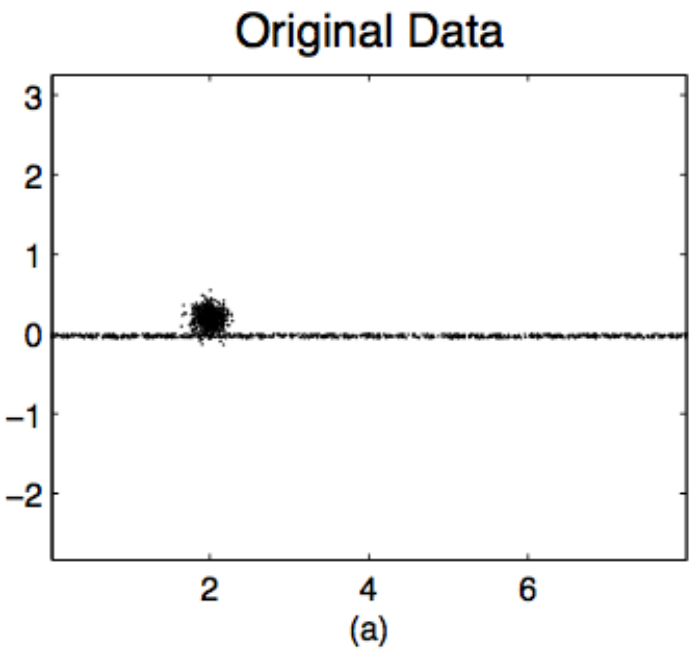
$$\sum_j \lambda_j^{2t}(v_j(x) - v_j(y))^2 = ||p(z,t|x) - p(z,t|y)||^2_{L_2(1/\pi_s)} \qquad (3)$$

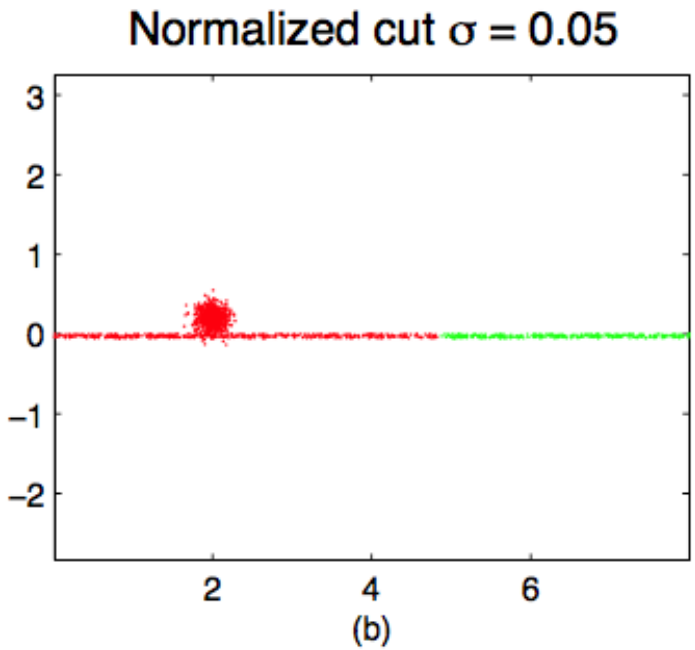
The spectral method want to capture the major pattern of the random walk on whole graph.

Success Cases

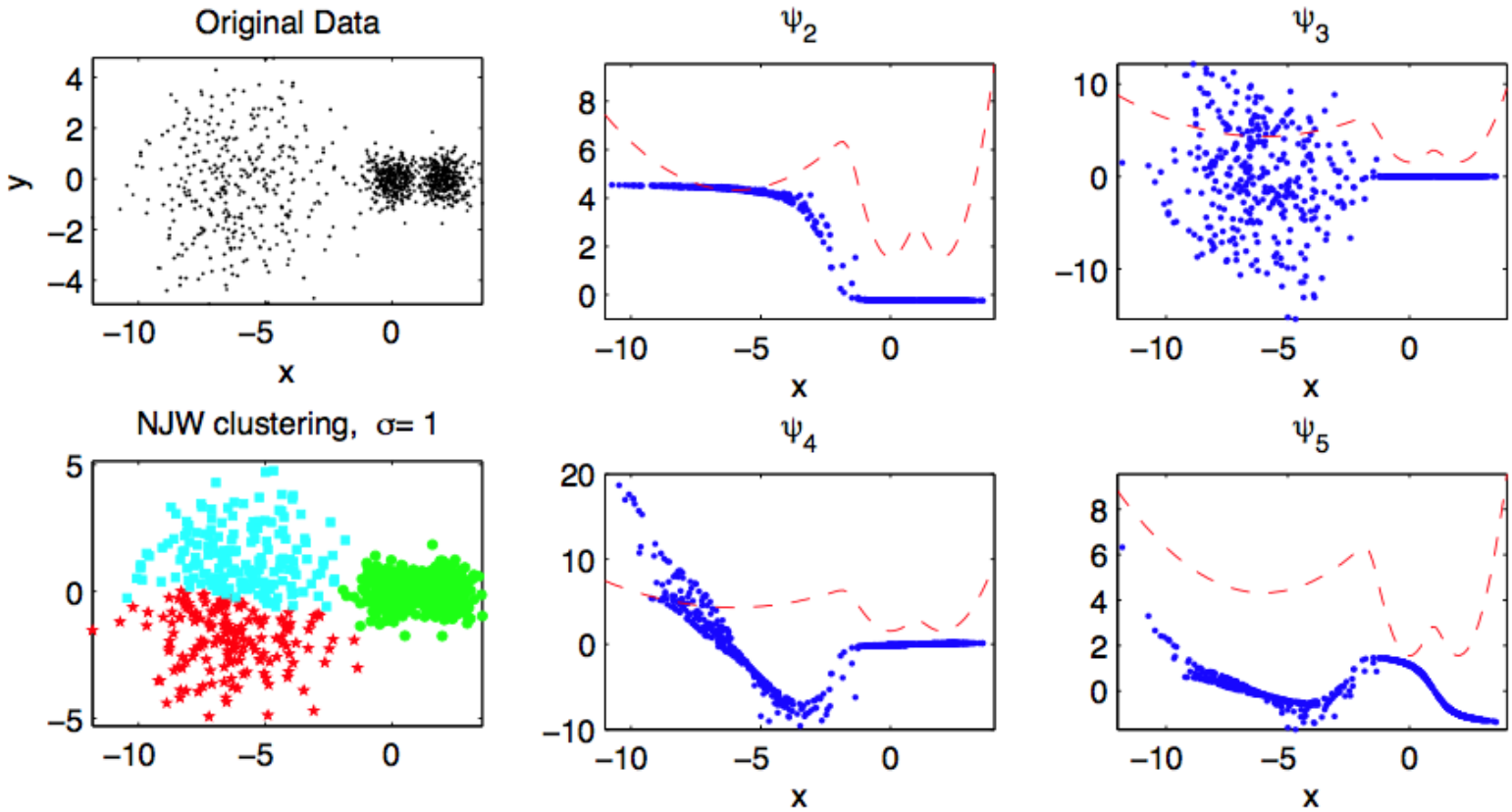


Failure Cases - Graph Cut





Failure Cases - Random Walk



PROJECT PRESENTATION

KERNEL MEAN EMBEDDINGS AND ITS APPLICATIONS

NAJI SHAJARISALES

MOTIVATION

1. Independence of Random Variables:

Is one text is related to the other in another language?

provincial and municipal governments in relation to the issue of child care, but the reality is that there have been no cuts to child care funding from the federal government to the provinces. In fact, we have increased federal investments for

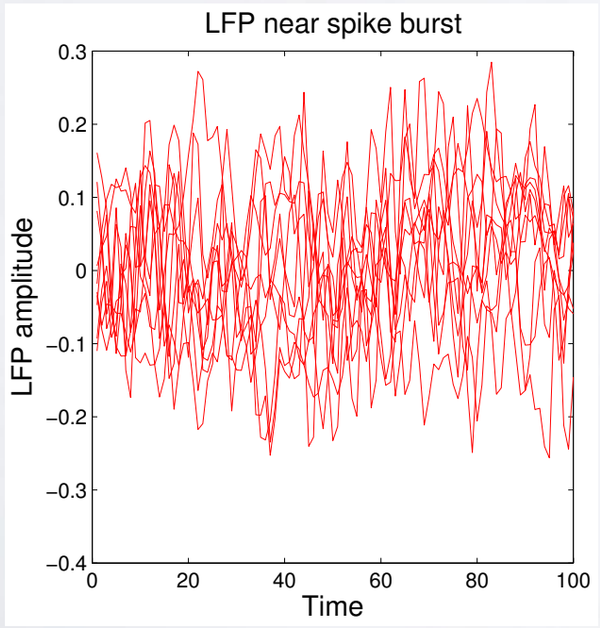
$\overset{?}{\Leftrightarrow}$

paux subissent de fortes pressions en ce qui concerne les services de garde, mais le gouvernement n'a pas réduit le financement qu'il verse aux provinces pour les services de garde. Au contraire, nous avons augmenté le finance-

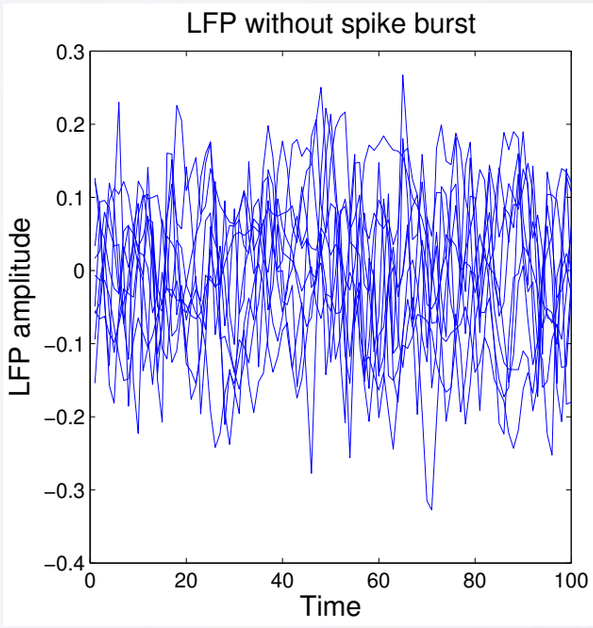
2. Difference Between Distributions:

Are LFPs near spike burst similar to LFPs with spike burst?

P



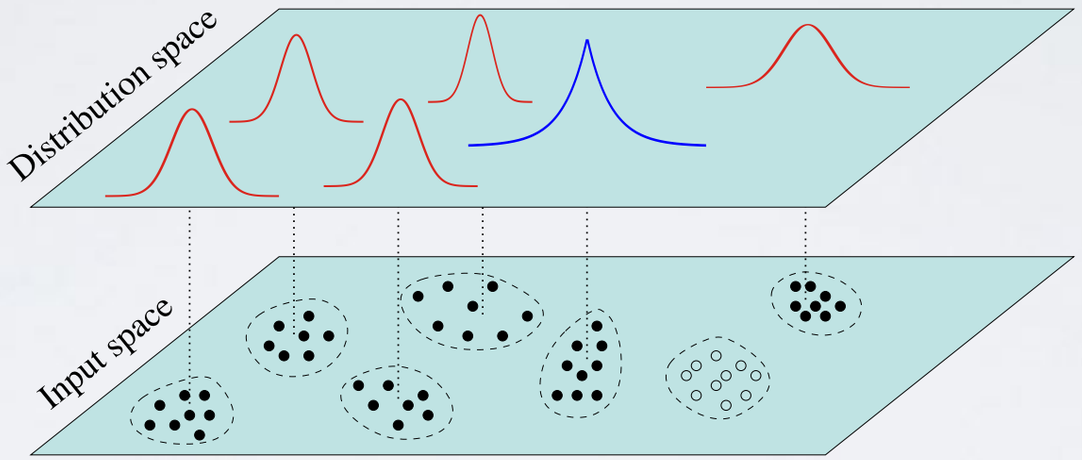
Q



MOTIVATION

3. Distributional Learning:

Given samples from distributions, find the anomalous distribution



General Objectives in Finding Methods:

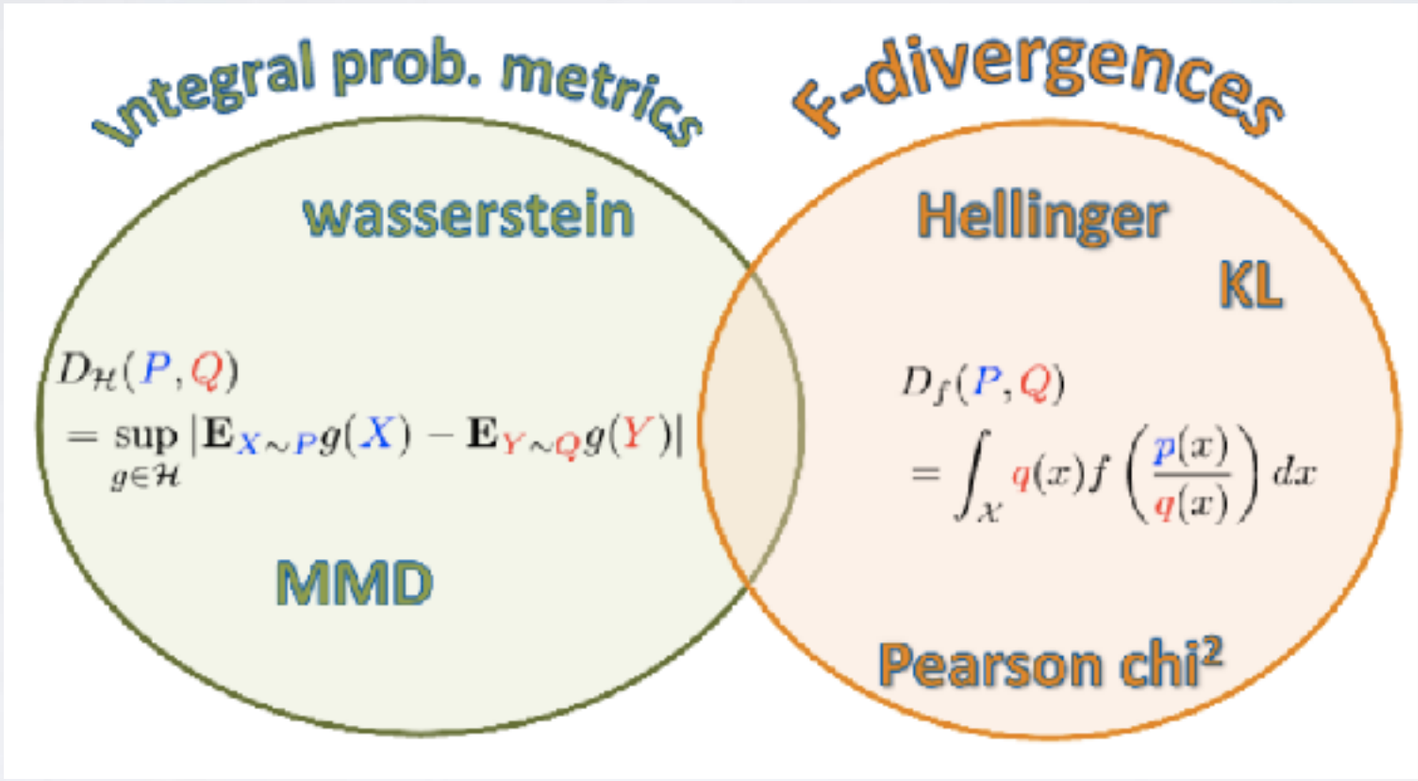
- Domain adaptive methods
- Computationally fast methods
- Consistent methods
- High convergence speed

TWO-SAMPLE TEST?

Example 1:

Looking for a distance d between P and Q s.t.

$$P \neq Q \iff d(P,Q) \neq 0 \quad (*)$$



- Slow convergence rate dependent on P and Q
- Sophisticated bias correction and partitioning

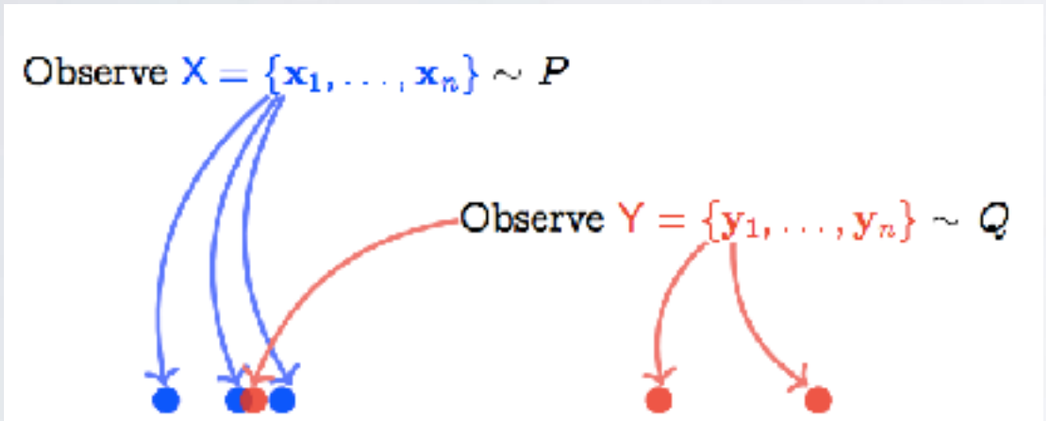
KERNEL MEAN EMBEDDING

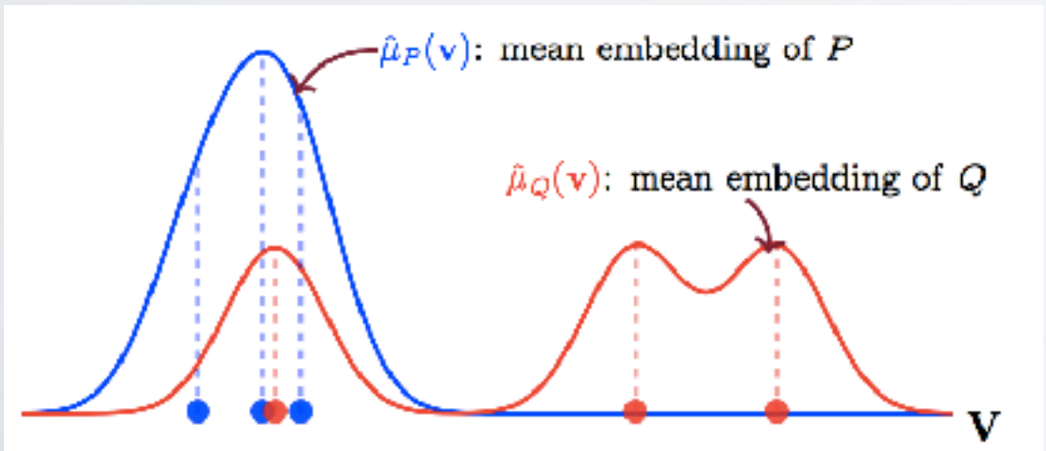
The **Kernel Mean Embedding (KME)** of a probability distribution P over $\mathcal{Z}$ associated with a measurable, bounded, and positive-definite kernel k is

$$\mu_k(P) := \int_Z k(z, \cdot) dP(z) \in \mathcal{H}_k.$$

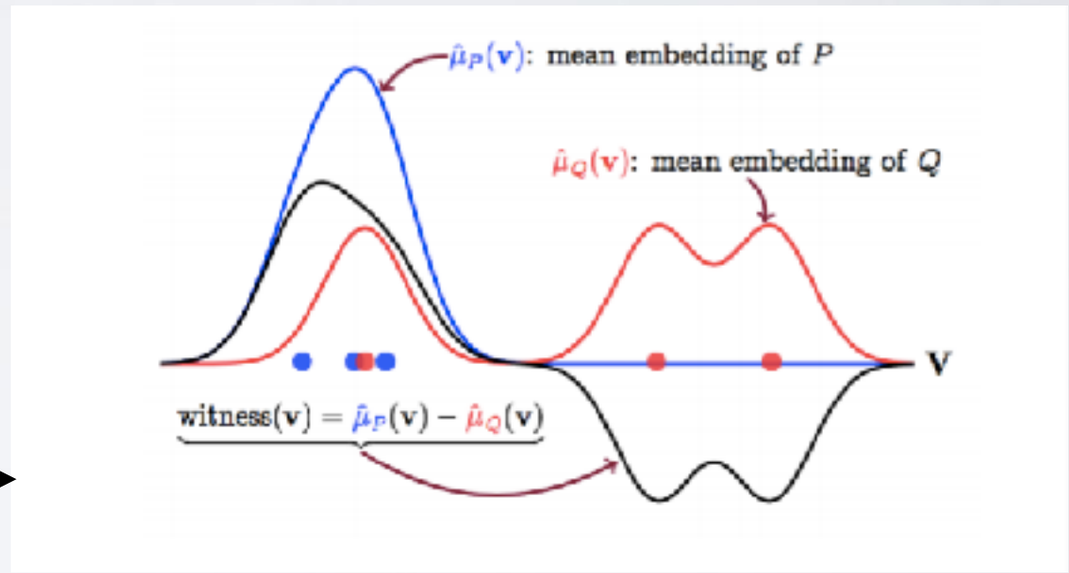
Example 1: Maximum Mean Discrepancy (MMD)

$$d(P,Q) = \int\int K(x,y)dP(x)dP(y) + \int\int K(x,y)dQ(x)dQ(y) - 2\int\int K(x,y)dP(x)dQ(y)$$





$$\hat{\mu}_P(\mathbf{v}) := \frac{1}{m}\sum_{i=1}^{m} k(x_i, v)$$



Estimator: $T = \frac{2}{n(n-1)}\sum_{i<j} K(X_i, X_j) +$

$$\frac{2}{m(m-1)}\sum_{i<j} K(Y_i, Y_j) - \frac{2}{mn}\sum_{i,j} K(X_i, Y_j)$$

$$T - d(P,Q) = O_P(\frac{1}{\sqrt{N}})$$

KERNEL MEAN EMBEDDING

- Domain adaptive methods ✅
- Computationally fast methods ✅
- Consistent methods ✅
- High convergence speed ✅

NO FREE LUNCH

Minimax Risk: $R_\varepsilon^{(m)}(f_0;\mathcal{H}) = \inf_\phi R_\varepsilon^{(m)}(\phi;f_0;\mathcal{H})$

Theorem 1. *For the one-sample problem under known Hölder regularity, there is a constant $c>0$ depending only on (s,d,L) such that*

$$R_\varepsilon^{(m)}(\mathcal{H}_s^d(L)) \geq 1/2, \quad \text{if } \varepsilon \leq c\, m^{-2s/(4s+d)}. \tag{16}$$

Theorem 4. *For the two-sample problem under known Hölder regularity, there is a constant $c>0$ depending only on (s,d,L) such that*

$$R_\varepsilon^{(m,n)}(\mathcal{H}_s^d(L)) \geq 1/2, \quad \text{if } \varepsilon \leq c(m\wedge n)^{-2s/(4s+d)}. \tag{29}$$

DISTRIBUTIONAL LEARNING CAUSAL DISCOVERY

Consider the two generative models:

causal model	**anticausal model**
$x \sim P$	$y \sim P$
$\epsilon \sim Q$	$\epsilon \sim Q$
$f \sim \mathcal{F}$	$f \sim \mathcal{F}$
$y \leftarrow f(x, \epsilon)$	$x \leftarrow f(y, \epsilon)$
+1	-1

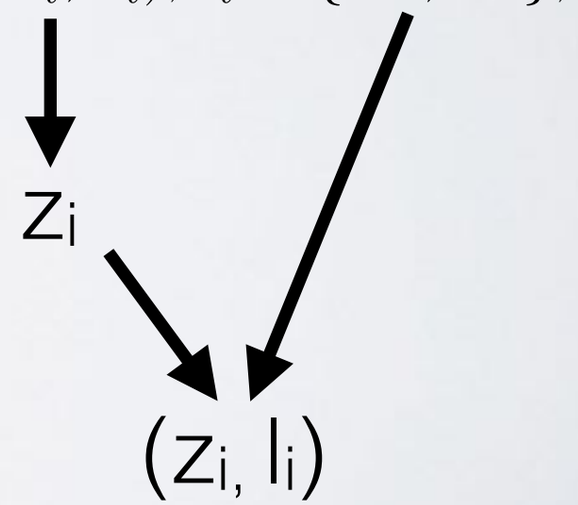
Algorithm:

Input

i) labeled causal samples $\{(S_i, l_i)\}_{i=1}^{n}$; $S_i = \{(x_{ij}, y_{ij})\}_{j=1}^{n_i} \sim P^{n_i}(X_i, Y_i)$, $l_i \in \{-1, +1\}$,

ii) measurable and bounded kernel function k, and

iii) number of random features m.



Testing

i) featurize test sample S_0 as $\mu_{k,m}(S_0)$ as in training, and

ii) return $\hat{f}_n(\mu_{k,m}(S_0))$.

DISTRIBUTIONAL LEARNING CAUSAL DISCOVERY

Training?

1. Minimize $R_\varphi(f) = \mathop{\mathbb{E}}_{(z,l)\sim\mathbb{P}}\big[\varphi\big(-f(z)l\big)\big],$

DISTRIBUTIONAL LEARNING
CAUSAL DISCOVERY

Training?

1. Minimize $R_\varphi(f) = \mathop{\mathbb{E}}_{(z,l)\sim\mathbb{P}}\big[\varphi\big(-f(z)l\big)\big],$

2. Minimize $\hat{R}_\varphi(f) = \frac{1}{n}\sum_{i=1}^{n}\varphi\big(-f(z_i)l_i\big).$

DISTRIBUTIONAL LEARNING CAUSAL DISCOVERY

Training?

1. Minimize $R_\varphi(f) = \mathop{\mathbb{E}}_{(z,l)\sim\mathbb{P}}\big[\varphi\big(-f(z)l\big)\big],$ ✖

2. Minimize $\hat{R}_\varphi(f) = \frac{1}{n}\sum_{i=1}^{n}\varphi\big(-f(z_i)l_i\big).$ ✖

3. Minimize $\tilde{R}_\varphi(f) = \frac{1}{n}\sum_{i=1}^{n}\varphi(-l, f(\mu_k(P_{S_i})))$

DISTRIBUTIONAL LEARNING CAUSAL DISCOVERY

Training?

1. Minimize $R_\varphi(f) = \mathop{\mathbb{E}}_{(z,l)\sim\mathbb{P}}\big[\varphi\big(-f(z)l\big)\big],$ ✗

2. Minimize $\hat{R}_\varphi(f) = \frac{1}{n}\sum_{i=1}^{n}\varphi\big(-f(z_i)l_i\big).$ ✗

3. Minimize $\tilde{R}_\varphi(f) = \frac{1}{n}\sum_{i=1}^{n}\varphi(-l, f(\mu_k(P_{S_i})))$ ✓

DISTRIBUTIONAL LEARNING CAUSAL DISCOVERY

Assumptions:

i) $\exists$ *Mother distribution* $\mathcal{M}$ on $\{\text{cause-effect measures } P \text{ on } \mathcal{Z}\} \times \{-1, 1\}$,

ii) $\{(P_i, l_i)\}_{i=1}^n \sim \mathcal{M}^n$; with l_i indicating $X_i \to Y_i$ or $X_i \leftarrow Y_i$ for P_i,

iii) training data of the form $S_i = \{(X_{i,j}, Y_{i,j})\}_{j=1}^{n_i} \sim P_i^{n_i}$,

iv) measurable and bounded kernel k with $\sup_{z \in \mathcal{Z}} k(z, z) \leq 1$,

v) class $\mathcal{F}_k$ of functionals mapping $\mathcal{H}_k$ to $\mathbb{R}$ with Lipschitz constants uniformly bounded by $L_{\mathcal{F}}$,

vi) minimization of surrogate risk $R_\varphi(f) := \mathbb{E}_{(P,l)\sim\mathcal{M}} \left[\varphi\big(-f\big(\mu_k(P)\big)l\big)\right]$ in $\mathcal{F}_k$,

vii) $\varphi\colon \mathbb{R} \to \mathbb{R}^+$ is L_φ-Lipschitz s.t. $\varphi(z) \geq \mathbb{1}_{z>0}$ and $\varphi\big(z\big) \leq B$ for all z.

Theorem:

With probability not less than $1 - \delta$ *over all sources of randomness*

$$R_\varphi(\tilde{f}_n) - R^*_{\varphi,\mathcal{F}_k} \leq 4L_\varphi R_n(\mathcal{F}_k) + 2B\sqrt{\frac{\log(2/\delta)}{2n}} + \frac{4L_\varphi L_{\mathcal{F}}}{n} \sum_{i=1}^{n} \left(\sqrt{\frac{\mathbb{E}_{z\sim P_i}[k(z,z)]}{n_i}} + \sqrt{\frac{\log(2n/\delta)}{2n_i}} \right).$$

CONCLUSION

- Domain adaptive methods ✅
- Computationally fast methods ✅
- Consistent methods ✅
- High convergence speed ✅
- Useful for distributional learning ✅
- Curse of dimensionality and not minimax efficient ❌

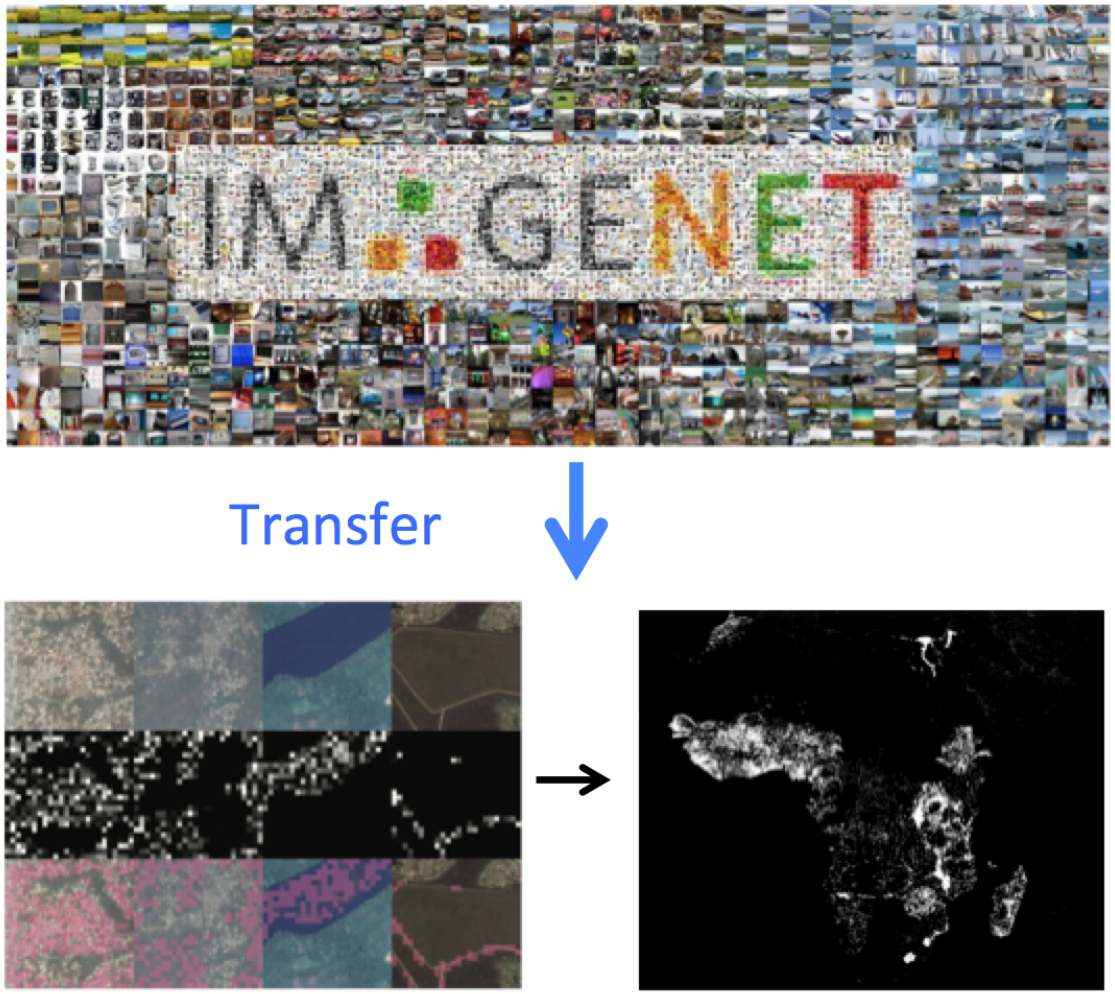
Transfer Learning for Sustainability, International Development and Public Policy

Lynn Kaack

May 2, 2017

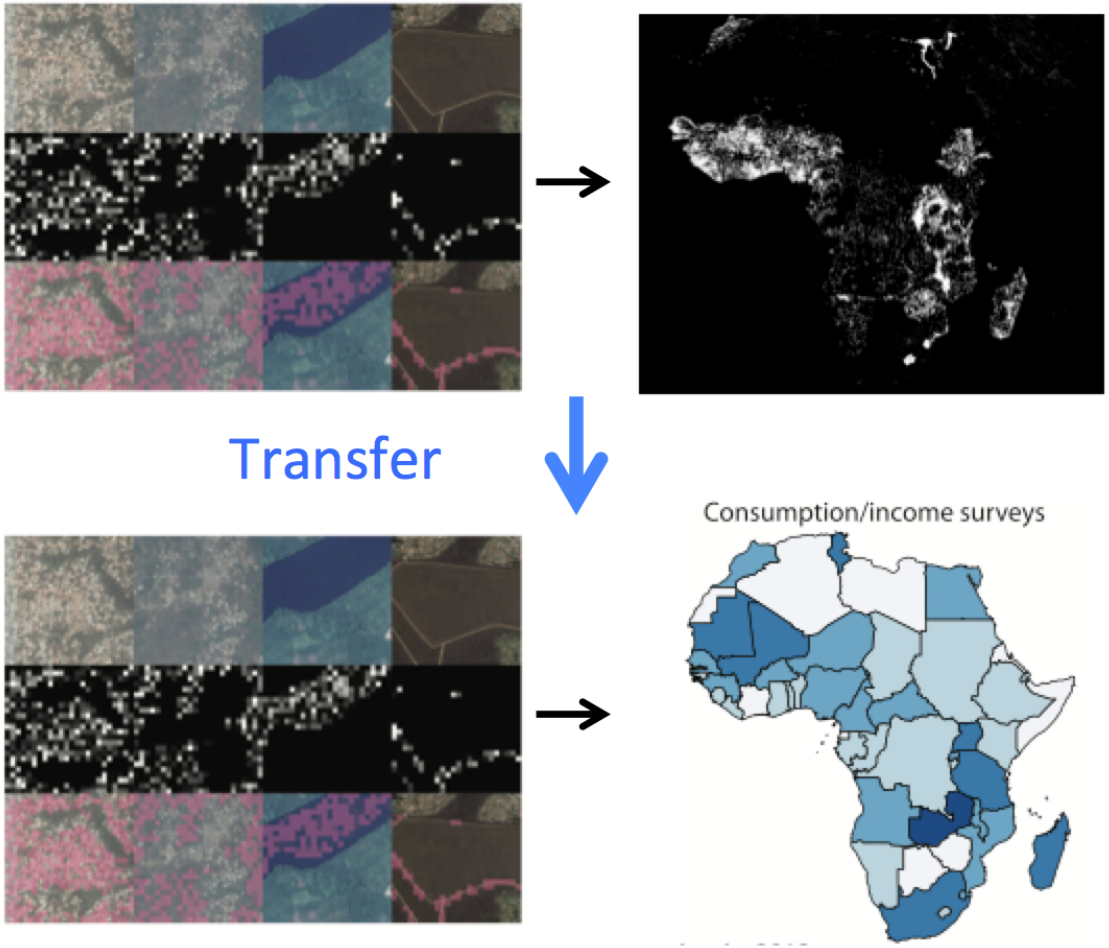
Example: Transfer learning for sustainability

Predicting poverty: Xie et al. (2015), Jean et al. (2016)



Example: Transfer learning for sustainability

Xie et al. 2015, Jean et al. 2016



Example of empirical validation

Approach

Compare to models without transfer learning

Results from Xie et al. 2015

	Survey	ImgNet	Lights	ImgNet +Lights	Transfer
Accuracy	0.754	0.686	0.526	0.683	**0.716**
F1 Score	0.552	0.398	0.448	0.400	**0.489**
Precision	0.450	0.340	0.298	0.338	**0.394**
Recall	0.722	0.492	0.914	0.506	0.658
AUC	0.776	0.690	0.719	0.700	**0.761**

Table 1: Cross validation test performance for predicting aggregate-level poverty measures. Survey is trained on survey data collected in the field. All other models are based on satellite imagery. Our transfer learning approach outperforms all non-survey classifiers significantly in every measure except recall, and approaches the survey model.

Transfer learning setting

- Source domain $\mathcal{D}_S$, learning task $\mathcal{T}_S \rightarrow$ target domain $\mathcal{D}_T$, task $\mathcal{T}_T$
- $\mathcal{D}_S \neq \mathcal{D}_T$, or $\mathcal{T}_S \neq \mathcal{T}_T$

Domain adaptation: A case of transductive transfer learning: $\mathcal{T}_S = \mathcal{T}_T$, equal feature spaces $\mathcal{X}_S = \mathcal{X}_T$, but different marginal distributions $P(\mathcal{X}_S) \neq P(\mathcal{X}_T)$

Target error bound for domain adaptation

Ben-David et al. (2010)
With probability at least $1 - \delta$

- $\mathcal{H} \triangle \mathcal{H}$-divergence,
  $\mathcal{U}_S, \mathcal{U}_T$ are unlabeled samples

$$\epsilon_T(h) \leq \epsilon_S(h) + \frac{1}{2}\hat{d}_{\mathcal{H}\triangle\mathcal{H}}(\mathcal{U}_S, \mathcal{U}_T) + \lambda + 4\sqrt{\frac{d\log(2m') + \log(2/\delta)}{m'}}$$

Target error bound for domain adaptation

Ben-David et al. (2010)
With probability at least $1 - \delta$

- $\mathcal{H} \triangle \mathcal{H}$-divergence,
  $\mathcal{U}_S, \mathcal{U}_T$ are unlabeled samples

$$\epsilon_T(h) \leq \epsilon_S(h) + \frac{1}{2}\hat{d}_{\mathcal{H}\triangle\mathcal{H}}(\mathcal{U}_S, \mathcal{U}_T) + \lambda + 4\sqrt{\frac{d\log(2m') + \log(2/\delta)}{m'}}$$

- error of ideal hypothesis $\lambda = \epsilon_S(h*) + \epsilon_T(h*)$

Target error bound for domain adaptation

Ben-David et al. (2010)
With probability at least $1 - \delta$

- $\mathcal{H} \triangle \mathcal{H}$-divergence,
  $\mathcal{U}_S, \mathcal{U}_T$ are unlabeled samples

$$\epsilon_T(h) \leq \epsilon_S(h) + \frac{1}{2}\hat{d}_{\mathcal{H}\triangle\mathcal{H}}(\mathcal{U}_S, \mathcal{U}_T) + \lambda + 4\sqrt{\frac{d\log(2m') + \log(2/\delta)}{m'}}$$

- error of ideal hypothesis $\lambda = \epsilon_S(h*) + \epsilon_T(h*)$
- $\mathcal{H}$ hypothesis space of VC dimension d; m' size of $\mathcal{U}_S, \mathcal{U}_T$ each

Sparse mixed logit model for discrete choice data

Cristobal De La Maza

Department of Engineering and Public Policy
Carnegie Mellon University

10-702 - CCML 2017

Introduction

Suppose these 3 vehicles below were the only vehicles available for purchase, which would you choose?

Each option will look like this:

Attribute*	Option 1	Option 2	Option 3
Vehicle Type	Hybrid 300 mile range on 1 tank	Plug-In Hybrid & 300 mile range on 1 tank (first 40 miles electric)	Hybrid 300 mile range on 1 tank
Brand	German	American	American
Purchase Price	$32,000	$15,000	$50,000
Fast Charging Capability	--	Available	--
Operating Cost (Equivalent Gasoline Fuel Efficiency)	6 cents per mile (60 MPG equivalent)	19 cents per mile (20 MPG equivalent)	12 cents per mile (30 MPG equivalent)
0 to 60 mph Acceleration Time**	7 seconds (Medium-Fast)	8.5 seconds (Medium-Slow)	8.5 seconds (Medium-Slow)
	○	○	○

*To view an attribute description, click on:
**The average acceleration for cars in the U.S. is 0 to 60 mph in 7.4 seconds

Figure: Choice experiment example from Halveston et al (2015) -

Random utility theory

- $P(choice = i) = P_{ni} = \frac{e^{\beta^t(z_n, x_{in})}}{\sum_{j \in A} e^{\beta^t(z_n, x_{jn})}}$
- Individual n will prefer alternative i with $U_i = V_i + \epsilon_i$ if $U_i \geq U_j \ \forall j$ [9]
- Error ϵ_i typically with a Type-I extreme value distribution [8].
- Deterministic utility $V_{in} = V_{in}(z_n, x_{in})$ depends on characteristics of the individual z_n and attributes of each alternative x_{in}.

Mixed logit model

- Mixed logit models consider β_i as random variables (usually $\beta_i \sim N(\mu, \Sigma)$) [6] [10].
- With many random parameters, variance-covariance matrix Σ will be dense, obscuring interpretation.
- Model estimated via stochastic programming with simulated maximum likelihood estimator (SLL) [14]:

$$\underset{\beta}{\text{minimize}} - SLL(\beta) = -\sum_{n=1}^{N} \sum_{jinA} I_{ni} log(\int P_{ni}/\beta \cdot f(\beta) d\beta)$$

Where $E_\beta(P_{ni}) = \int P_{ni}/\beta \cdot f(\beta) d\beta \approx \frac{1}{R} \sum_{r=1}^{R} \frac{e^{\beta_r^t x_{in}}}{\sum_{j \in A} e^{\beta_r^t x_{jn}}}$

R is the number of draws from $f(\beta)$.

Sparse group mixed logit model

- Some literature on mixed models with sparsity penalties [7] [3] [12] [4] [5] [11].
- A sparse group mixed logit model will correspond to [13]:

$$\underset{\beta}{\text{minimize}} - SLL(\theta) + P_\lambda(\theta) =$$
$$-SLL(\theta) + \alpha\lambda \sum_{i=1}^{p} \gamma_j |\theta_j| + (1-\alpha)\lambda \tfrac{1}{2} \sum_{l=1}^{m} \sqrt{p_l}\gamma_{(l)} \left\|\theta_{(l)}\right\|$$

- Here m is the number of groups of variables in the data.
- Weights γ for diagonal terms of variance-covariance matrix Σ, σ_{ii} must be forced to 0.
- One group for off-diagonal terms of variance-covariance matrix Σ, σ_{ij} with $i \neq j$.

Simulation bias and model bias

Two main sources of bias:

- Lasso regression will bias coefficients towards zero. Fact from $\text{prox}_{\lambda t}(\theta_j) = S_{\lambda t}(\theta_j) = \text{sign}(\theta_j) \cdot \max\{|\theta_j| - \lambda t, 0\}$.
- Additionally, simulation bias is [1] [2]:

  If, $\sqrt{R}(LL(\theta) - SLL(\theta)) \xrightarrow{d} N(0, \frac{1}{N \cdot R} \sum_{r=1}^{R} \sum_{n=1}^{N} \frac{\sigma_i^2(\theta)}{RP_i(\theta)^2})$

  $SLL(\theta) - LL(\theta) \leq \frac{\alpha_\delta}{N \cdot R} \sqrt{\sum_{r=1}^{R} \sum_{n=1}^{N} \frac{\sigma_i^2(\theta)}{RP_i(\theta)^2}} = O(\frac{1}{\sqrt{R}})$

- No papers found on risk rates for sparse mixed models.

References I


Fabian Bastin and Cinzia Cirillo. "Reducing simulation bias in mixed logit model estimation". In: *Journal of Choice Modelling* 3.2 (2010), pp. 71–88.


Fabian Bastin, Cinzia Cirillo, and Philippe L Toint. "Convergence theory for nonconvex stochastic programming with an application to mixed logit". In: *Mathematical Programming* 108.2 (2006), pp. 207–234.


Howard D Bondell, Arun Krishna, and Sujit K Ghosh. "Joint Variable Selection for Fixed and Random Effects in Linear Mixed-Effects Models". In: *Biometrics* 66.4 (2010), pp. 1069–1077.


Zhen Chen and David B Dunson. "Random effects selection in linear mixed models". In: *Biometrics* 59.4 (2003), pp. 762–769.

References II


Andreas Groll and Gerhard Tutz. "Variable selection for generalized linear mixed models by L 1-penalized estimation". In: *Statistics and computing* 24.2 (2014), pp. 137–154.


David A Hensher and William H Greene. "The mixed logit model: the state of practice". In: *Transportation* 30.2 (2003), pp. 133–176.


Joseph G Ibrahim et al. "Fixed and random effects selection in mixed effects models". In: *Biometrics* 67.2 (2011), pp. 495–503.


Jacob Marschak et al. *Binary choice constraints on random utility indicators*. Tech. rep. Cowles Foundation for Research in Economics, Yale University, 1959.


Daniel McFadden et al. "Conditional logit analysis of qualitative choice behavior". In: (1973).

References III

- Daniel McFadden and Kenneth Train. "Mixed MNL models for discrete response". In: *Journal of applied Econometrics* (2000), pp. 447–470.
- Jürg Schelldorfer, Lukas Meier, and Peter Bühlmann. "Glmmlasso: an algorithm for high-dimensional generalized linear mixed models using â1-penalization". In: *Journal of Computational and Graphical Statistics* 23.2 (2014), pp. 460–477.
- Jürg Schelldorfer et al. "Estimation for High-Dimensional Linear Mixed-Effects Models Using â1-Penalization". In: *Scandinavian Journal of Statistics* 38.2 (2011), pp. 197–214.
- Noah Simon et al. "A sparse-group lasso". In: *Journal of Computational and Graphical Statistics* 22.2 (2013), pp. 231–245.

References IV

Kenneth E Train. *Discrete choice methods with simulation*. Cambridge university press, 2009.

Deep Variational Inference

Qizhe Xie

May 2, 2017

Evidence Lower Bound (ELBO)

The evidence lower bound (ELBO) is defined as

$$\mathcal{L} = \mathbb{E}_q[\log p(\mathbf{z}, \mathbf{x})] - \mathbb{E}_q[\log q_\phi(\mathbf{z}|\mathbf{x})] \tag{1}$$

Theorem

Maximizing ELBO is equivalent to minimizing the KL distance between the variational distribution $q_\phi(\mathbf{z}|\mathbf{x})$ and the true posterior distribution $p_\theta(\mathbf{z}|\mathbf{x})$.

Proof.

$$\begin{aligned}
\mathcal{L} &= \mathbb{E}_q[\log p_\theta(\mathbf{z}, \mathbf{x})] - \mathbb{E}_q[\log q_\phi(\mathbf{z}|\mathbf{x})] \\
&= \mathbb{E}_q[\log p_\theta(\mathbf{x})] + \mathbb{E}_q[\log p_\theta(\mathbf{z}|\mathbf{x})] - \mathbb{E}_q[\log q_\phi(\mathbf{z}|\mathbf{x})] \qquad (2) \\
&= \log p_\theta(\mathbf{x}) - KL(q_\phi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z}|\mathbf{x}))
\end{aligned}$$

□

Efficient Optimization: Auto-encoding Variational Bayes

- Naive Monte Carlo estimation has high variance.
- Reparametrization trick: reparameterize the random variable $\hat{\mathbf{z}} \sim q_\phi(\mathbf{z}|\mathbf{x})$ with a differentiable transformation $\hat{\mathbf{z}} = g_\phi(\epsilon, \mathbf{x})$ using a noise variable ϵ.
- Efficient optimization by SGD.

$$\mathbb{E}_q[\log p_\theta(\mathbf{x}|\mathbf{z})] = \frac{1}{L}\sum_{l=1}^{L}(\log p_\theta(\mathbf{x}|\mathbf{z}^{(l)})) \tag{3}$$

$$\text{where} \quad \mathbf{z}^{(l)} = g_\phi(\mathbf{x}, \boldsymbol{\epsilon}^{(l)}) \quad \text{and} \quad \boldsymbol{\epsilon}^{(l)} \sim p(\boldsymbol{\epsilon}) \tag{4}$$

Rich Posterior: Normalizing Flows, Mixture of Distributions

- Normalizing flows: Transforming a random variable $\mathbf{z}_0$ with distribution q_0 through a chain of K transformations:

$$\mathbf{z}_K = f_K \circ \cdots \circ f_2 \circ f_1(\mathbf{z}_0) \tag{5}$$

Theorem

Suppose we adopt a family of transformations of the form $f_k(\mathbf{z}) = \mathbf{z} + \mathbf{u}_k h(\mathbf{w}_k^T \mathbf{z} + b_k)$. *The flow-based ELBO is*

$$\mathbb{E}_{q_0(z_0)}\left[\ln q_0(\mathbf{z}_0)\right] - \mathbb{E}_{q_0(z_0)}\left[\sum_{k=1}^{K} \ln\left|1 + \mathbf{u}_k^\top \psi_k(\mathbf{z}_{k-1})\right|\right] - \mathbb{E}_{q_0(z_0)}\left[\ln p(\mathbf{x}, \mathbf{z}_K)\right] \tag{6}$$

- Mixture of Distributions: Approximating posterior using a mixture of distributions with bootstrapping.

Analysis of auto-regressive VAEs through Bits-Back Coding

VAE can be seen as encoding data in a two-part code $p(\mathbf{z})$ and $p(\mathbf{x}|\mathbf{z})$.

Lemma

The average code length encoded by VI is
$C_{BitsBack}(\mathbf{x}) = \mathbb{E}_{\mathbf{x}\sim\mathrm{data},\mathbf{z}\sim q(\mathbf{z}|\mathbf{x})}[\log q(\mathbf{z}|\mathbf{x}) - \log p(\mathbf{z}) - logp(\mathbf{x}|\mathbf{z})]$.

Theorem

The two-part code from VAE suffers at least a length of $KL(q(\mathbf{z}|\mathbf{x})||p(\mathbf{z}|\mathbf{x}))]$.

Corollary

Asymptotically, any distribution $p(\mathbf{x})$ *can be modeled perfectly without using* $\mathbf{z}$ *in auto-regressive models.*

Sensitivity of Variational Bayes to Prior

Suppose the prior $\mathbf{z}$ is dependent on ϵ modeled by $p(\mathbf{z}|\epsilon)$.

Theorem

The robustness of $\mathbb{E}_{p(\mathbf{z}|\epsilon,\mathbf{x})}[g(\mathbf{z})]$ *with respect to perturbation can be characterized by*

$$\frac{\partial \mathbb{E}_{p(\mathbf{z}|\epsilon,\mathbf{x})}[g(\mathbf{z})]}{\partial \epsilon}|_{\epsilon} \leq \max(\frac{1}{\epsilon}, \frac{1}{1-\epsilon}) \mathbb{E}_{p(\mathbf{z}|\epsilon,\mathbf{x})}[|g(\mathbf{z}) - \mathbb{E}_{p(\mathbf{z}|\epsilon,\mathbf{x})}[g(\mathbf{z})]|] \tag{7}$$

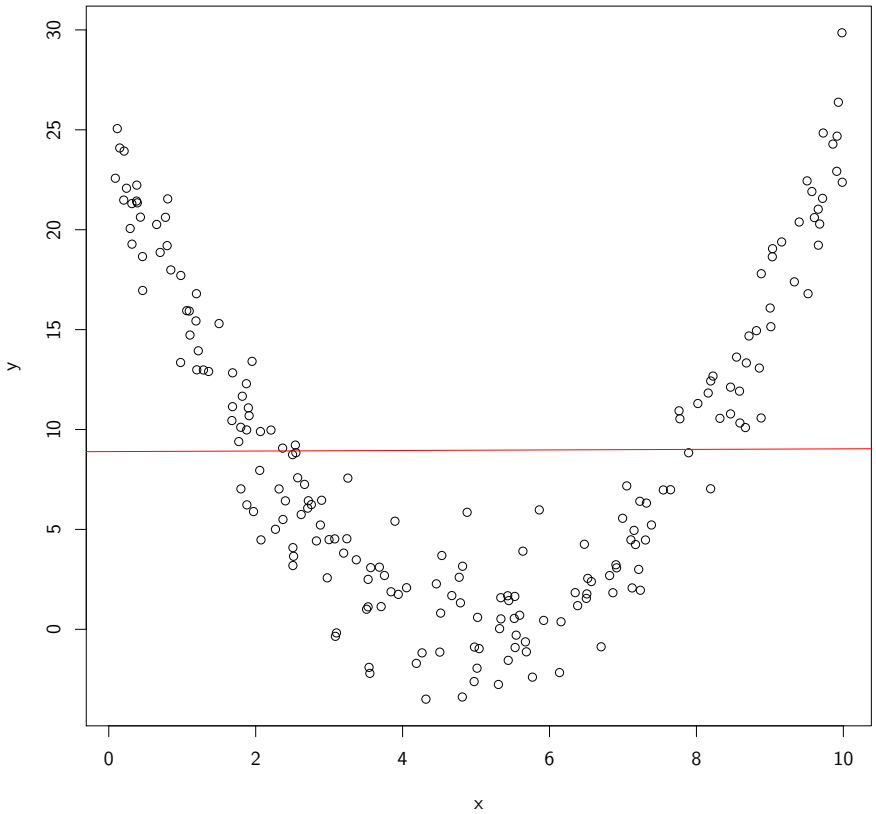
Mercer Kernel Methods for Testing Independence

Octavio Mesner

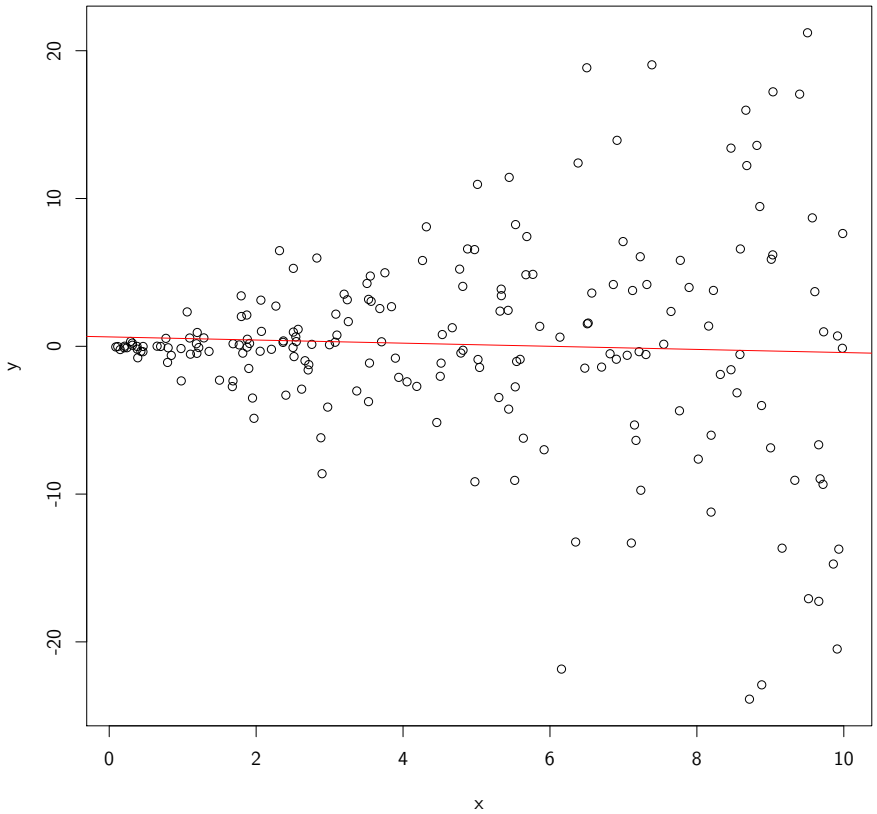
Statistical Machine Learning
Spring 2017

Correlation and Independence

$\rho = 0.01$, p-value=0.942



$\rho = -0.05$, p-value=0.517



Characteristic Kernels

- $\mathcal{P}$, set of probability measures on a space $\mathcal{X}$ with RKHS $\mathscr{H}$
- $\mu : \mathcal{P} \to \mathscr{H}$, $\mathbb{P} \mapsto \int_{\mathcal{X}} k(x, \cdot) d\mathbb{P}(x)$
- k is *characteristic* if μ is injective
- Intuitively, $\mathscr{H}$ is rich enough to represent higher order moments of $\mathbb{P}$
- Recall: X, Y independent iff $\phi_{X,Y} = \phi_X \phi_Y$

Cross Covariance Operator

- $(X, Y, Z) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$; RKHSs $\mathcal{H}_\mathcal{X}, \mathcal{H}_\mathcal{Y}, \mathcal{H}_\mathcal{Z}$; kernels k_X, k_Y, k_Z
- The *cross-covariance operator*, $\Sigma_{YX} : \mathcal{H}_\mathcal{X} \to \mathcal{H}_\mathcal{Y}$ defined by

$$\langle g, \Sigma_{YX} f \rangle_{\mathcal{H}_\mathcal{Y}} = \mathbb{E}[f(X)g(Y)] - \mathbb{E}[f(X)]\mathbb{E}[g(Y)]$$

  for all $f \in \mathcal{H}_\mathcal{X}, g \in \mathcal{H}_\mathcal{Y}$
- The *conditional cross-covariance operator* is defined as

$$\Sigma_{YX|Z} = \Sigma_{YX} - \Sigma_{YZ}\Sigma_{ZZ}^{-1}\Sigma_{ZX}$$

Independence

Theorem

1. *If $k_X k_Y$ is characteristic then*

$$\Sigma_{YX} = 0 \Leftrightarrow X \perp\!\!\!\perp Y$$

2. *Let $\ddot{X} = (X, Z)$ and $k_{\ddot{X}} = k_X k_Z$*
   If $k_{\ddot{X}} k_Y$ if characteristic then

$$\Sigma_{YX|Z} = 0 \Leftrightarrow X \perp\!\!\!\perp Y|Z$$

Proof Outline:
Since $\mathbb{E}[f(X)g(Y)] - \mathbb{E}[f(X)]\mathbb{E}[g(Y)] = 0$ for $f \in \mathscr{H}_{\mathcal{X}}, g \in \mathscr{H}_{\mathcal{Y}}$
We can show that $\phi_{X,Y} = \phi_X \phi_Y$ if we can make f, g look characteristic

Hilbert-Schmidt Operator Norm

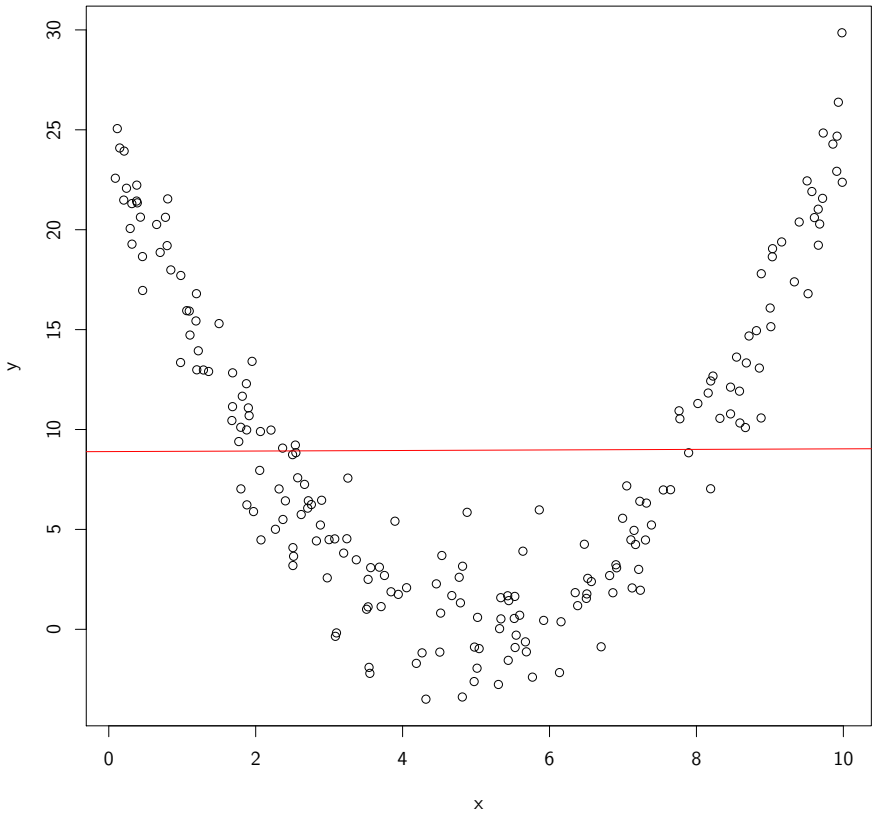
Let $(x_i)_{i\in I}, (y_j)_{j\in J}$ be orthonormal bases for $\mathscr{H}_{\mathcal{X}}$ and $\mathscr{H}_{\mathcal{Y}}$, respectively

$$\|\Sigma_{YX}\|_{HS}^2 = \sum_{i\in I}\sum_{j\in J} \langle \Sigma_{YX} x_i, y_i \rangle_{\mathscr{H}_{\mathcal{Y}}}^2$$

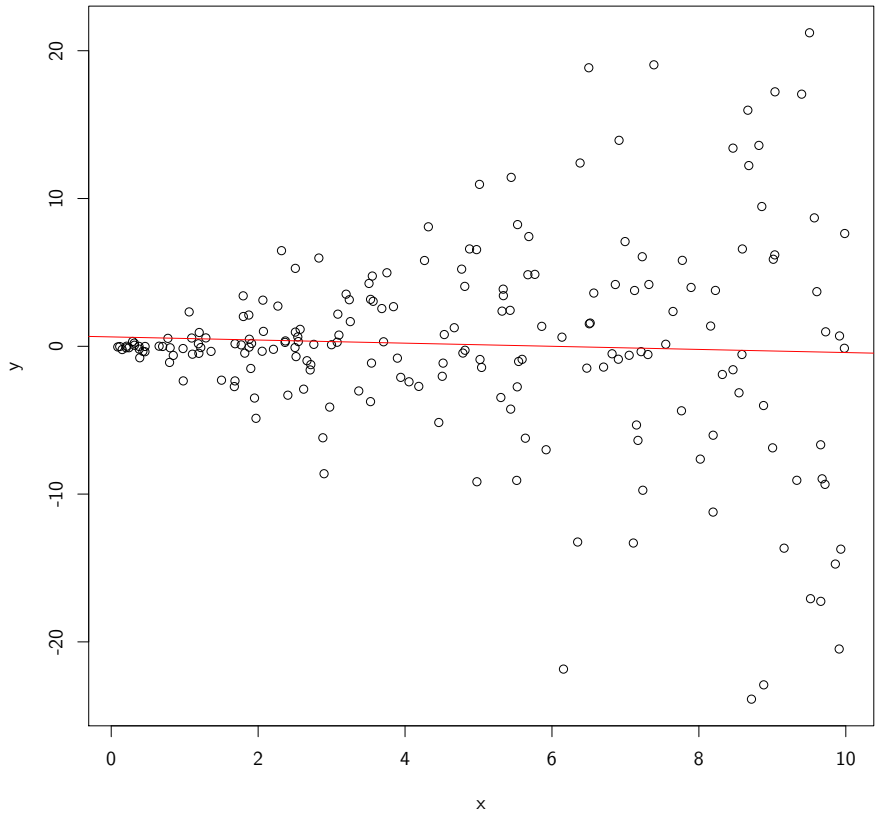
- $\|\Sigma_{YX}\|_{HS}^2 = 0 \Leftrightarrow \Sigma_{YX} = 0$
- Test Statistic is based on an estimate of this value

Mercer Kernel Tests for Independence

p-value $= 1.46 \times 10^{-18}$



p-value $= 1.178 \times 10^{-18}$



References I

- Francis R Bach and Michael I Jordan, *Kernel independent component analysis*, Journal of machine learning research **3** (2002), no. Jul, 1–48.
- Kenji Fukumizu, Arthur Gretton, Xiaohai Sun, and Bernhard Schölkopf, *Kernel measures of conditional dependence*, NIPS, vol. 20, 2007, pp. 489–496.
- Arthur Gretton, Kenji Fukumizu, Choon Hui Teo, Le Song, Bernhard Schölkopf, Alexander J Smola, et al., *A kernel statistical test of independence*, NIPS, vol. 20, 2007, pp. 585–592.

10702 - Statistical Machine Learning

Fusing Side-Information for One-Shot Learning

Yao-Hung Hubert Tsai and Mu-Chu Lee

Machine Learning Department, Carnegie Mellon University

yaohungt@cs.cmu.edu, *muchul@andrew.cmu.edu*

May 2, 2017

Hilbert-Schmidt Independence Criterion (HSIC)

- Side Information:
  human annotated attributes, unsupervised word vectors, and object tree hierarchical structures
- Notations: learned output visual embeddings $g_\theta(\mathbf{X})$ and side information $\mathbf{R}$
- HSIC [1] acts as a non-parametric independence test between two random variables, $g_\theta(\mathbf{X})$ and $\mathbf{R}$, by computing the Hilbert-Schmidt norm of the covariance operator over the corresponding domains $\mathcal{G} \times \mathcal{R}$.

Hilbert-Schmidt Independence Criterion (HSIC) (cont'd)

- Let k_g and k_r be the kernels on $\mathcal{G}, \mathcal{R}$ with associated Reproducing Kernel Hilbert Spaces (RKHSs), a slightly biased empirical estimation of HSIC can be written as

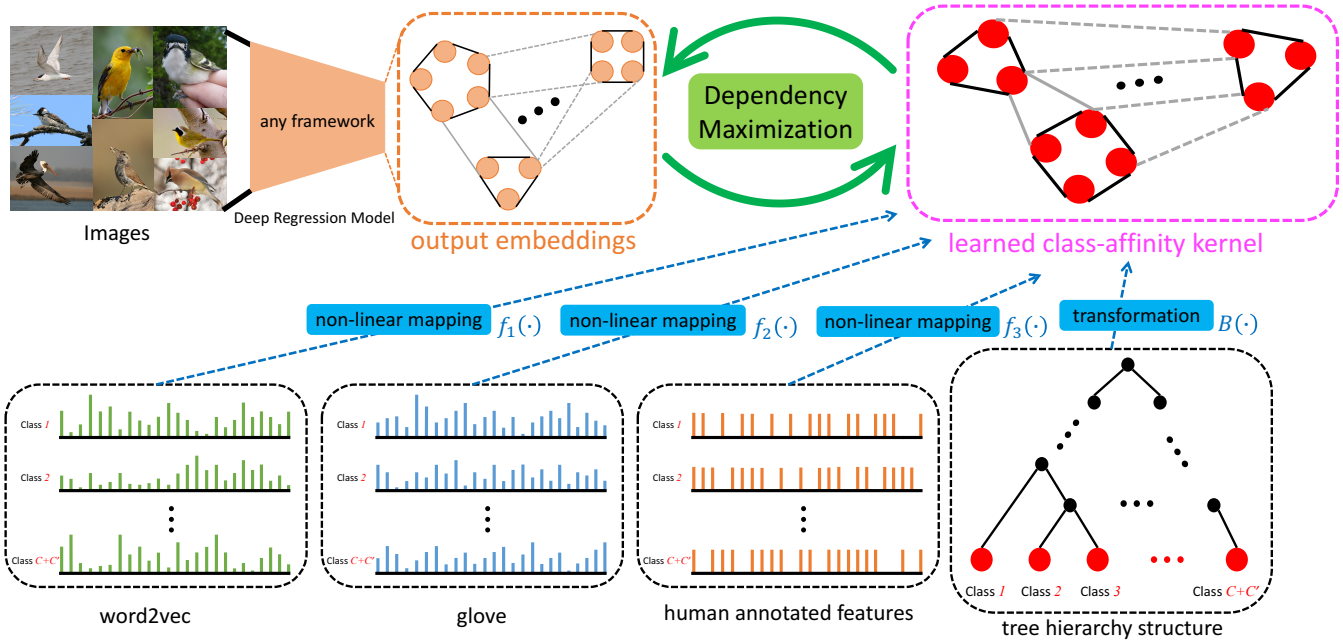
$$\mathrm{HSIC}(\mathbf{G}, \mathbf{R}) = \frac{1}{(N-1)^2}\mathrm{tr}(\mathbf{H}\mathbf{K}_{\mathbf{G}}\mathbf{H}\mathbf{K}_{\mathbf{R}}),$$

where $\mathbf{K}_{\mathbf{G}ij} = k_g(x_i, x_j)$, $\mathbf{K}_{\mathbf{R}ij} = k_r(y_i, y_j)$, and $\mathbf{H}_{ij} = \mathbb{1}_{\{i=j\}} - \frac{1}{(N-1)^2}$.

Yao-Hung Hubert Tsai and Mu-Chu Lee

Proposed Framework

- Dependency Maximization on the visual embeddings and side information under statistical guarantee.
- We train on images and side information across 'lots-of-examples' and 'few-examples' categories.



References

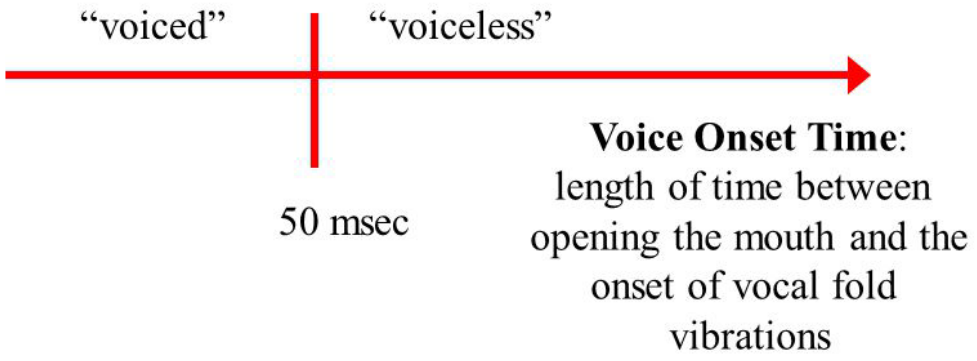
A. Gretton, O. Bousquet, A. Smola, and B. Schölkopf, "Measuring statistical dependence with hilbert-schmidt norms," in *International conference on algorithmic learning theory*, 2005.

Risk Analysis for Structured Prediction Algorithms

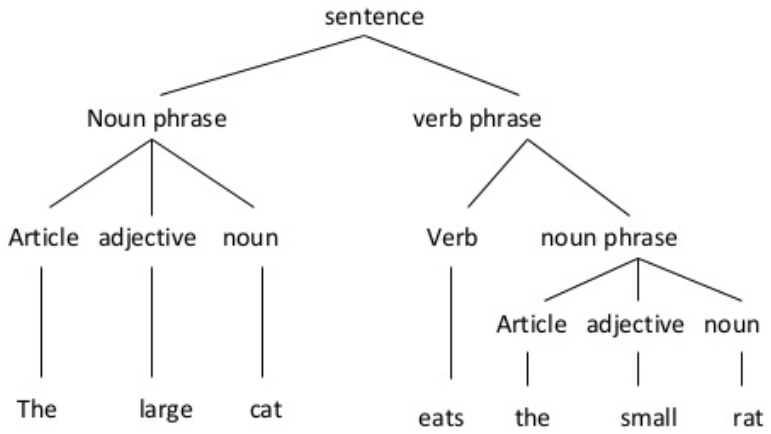
Wenbo Zhao (wzhao1)

May 2, 2017

Many supervised tasks involve predicting structured outputs.



Voice onset time: sequential



Syntactic parsing: tree

Image segmentation: graph

Structured prediction formulation

$$\text{Data } \{(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_N, \mathbf{y}_N)\} \in \mathcal{X} \times \mathcal{Y}$$
$$\text{Feature map } \phi : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^d$$
$$\text{Prediction } \mathbf{y}^*_{\mathbf{w}}(\mathbf{x}) = \text{argmax}_{\mathbf{y} \in \mathcal{Y}} \langle \mathbf{w}, \phi(\mathbf{x}, \mathbf{y}) \rangle$$
$$\text{Loss } L(\mathbf{y}, \mathbf{y}^*_{\mathbf{w}}(\mathbf{x}))$$
$$\text{Risk } R^L_p(\mathbf{w}) = \mathbb{E}_{(\mathbf{x},\mathbf{y}) \sim p}[L(\mathbf{y}, \mathbf{y}^*_{\mathbf{w}}(\mathbf{x}))]$$
$$\text{Minimize risk } \mathbf{w}^* = \text{argmin}_{\mathbf{w}} \, \mathbb{E}_{(\mathbf{x},\mathbf{y}) \sim p}[L(\mathbf{y}, \mathbf{y}^*_{\mathbf{w}}(\mathbf{x}))]$$

Review of three structured prediction algorithms: (1) structured SVM, (2) structured perceptron, (3) Markov random fields.

Structured SVM $\min_{\mathbf{w},\xi} \frac{1}{2}\|\mathbf{w}\|^2 + \frac{C}{N}\sum_{i=1}^{N}\xi_i$

$$\text{s.t. } \langle \mathbf{w}, \delta\phi_i(\mathbf{y})\rangle \geq 1 - \xi_i,\ \xi_i \geq 0,\ \forall \mathbf{y} \in \mathcal{Y}\backslash \mathbf{y}_i, \forall i$$

→ Structured hinge loss

Structured perceptron $\mathbf{w}^{t+1} = \mathbf{w}^t + \eta^t(\phi(\mathbf{x}_t, \mathbf{y}^*_{\mathbf{w}^t}(\mathbf{x}_t)) - \phi(\mathbf{x}_t, \tilde{\mathbf{y}}^*_t))$

Markov random fields $\text{Potential function } \psi(\mathbf{x}, y_i, y_j) = \exp\{\sum_{k=1}^{N} w_k f_k(\mathbf{x}, y_i, y_j)\}$

$\text{Maximizing log-likelihoods } \mathbf{w}^* = \text{argmax}_{\mathbf{w}} \sum_{e_{ij}\in\mathcal{E}} \langle \mathbf{w}, \mathbf{f}(\mathbf{x}, y_i, y_j)\rangle$

→ Structured log loss

Key results

1. Risk bound on structured SVM

Solve dual problem with cutting plane method, lower-bound on improvement $\delta\Theta \geq \min\{\frac{C\epsilon}{2N}, \frac{\epsilon^2}{8\Delta_i^2 R_i^2}\}$, $\Delta_i = \max_{\mathbf{y}} L(\mathbf{y}_i, \mathbf{y})$, $R_i = \max_{\mathbf{y}} \|\delta\phi_i(\mathbf{y})\|$.

2. Convergence of structured perceptron

The expected update direction approaches the negative direction of $\nabla_{\mathbf{w}}\mathbb{E}[L(\mathbf{y}, \mathbf{y}^*_{\mathbf{w}}(\mathbf{x}))]$ in the limit as the update weight ϵ^t goes to zero

$$\nabla_{\mathbf{w}}\mathbb{E}[L(\mathbf{y}, \mathbf{y}^*_{\mathbf{w}}(\mathbf{x}))] = \lim_{\epsilon \mapsto 0} \frac{1}{\epsilon}\mathbb{E}[\phi(\mathbf{x}, \tilde{\mathbf{y}}^*) - \phi(\mathbf{x}, \mathbf{y}^*_{\mathbf{w}}(\mathbf{x}))]$$

3. Risk bound on Markov random field

Generalization bound in terms of γ-margin per-label loss $L^\gamma(\mathbf{w}, \mathbf{x})$: For any $\delta > 0$, there exists a constant K such that for any $\gamma > 0$ and $m > 1$ samples, the per-label loss is bounded by

$$\mathbb{E}[L(\mathbf{w}, \mathbf{x})] \leq \mathbb{E}[L^\gamma(\mathbf{w}, \mathbf{x})] + \sqrt{\frac{K}{m}\left[\frac{R^2_{edge}\|\mathbf{w}\|_2^2 q^2}{\gamma^2}[\ln m + \ln l + \ln q + \ln k] + \ln\tfrac{1}{\delta}\right]}$$

with probability at least $1-\delta$, maximum edge degree in the graph $q = \max_i |\{(i,j) \in \mathcal{E}\}|$, number of classes in a label k, number of labels l.

Key results

4. Convergence of structured SVM with subgradient method

Linear convergence of constant stepsize sequence: Stepsize sequence $\{\alpha_t\}$, $\alpha_t = \alpha \leq \frac{1}{\lambda}$, for a particular region of radius R around the minimum, $\forall w, g \in \partial c(w)$, $\|g\| \leq C$, the algorithm converges at a linear rate to a region of the minimum $w^* = \text{argmin}_{w \in \mathcal{W}} c(w)$ bounded by $\|w_{min} - w^*\| \leq \frac{C}{\lambda}$.

5. Proof of convergence of conditional random fields inference algorithms

6. Proof of bounds for approximate Markov random fields inference algorithms

7. Comparison of probit, orbit and ramp losses, proof of their consistency, convergence rate and error bounds

Fast Feature Hashing in Linear/Non-linear Models

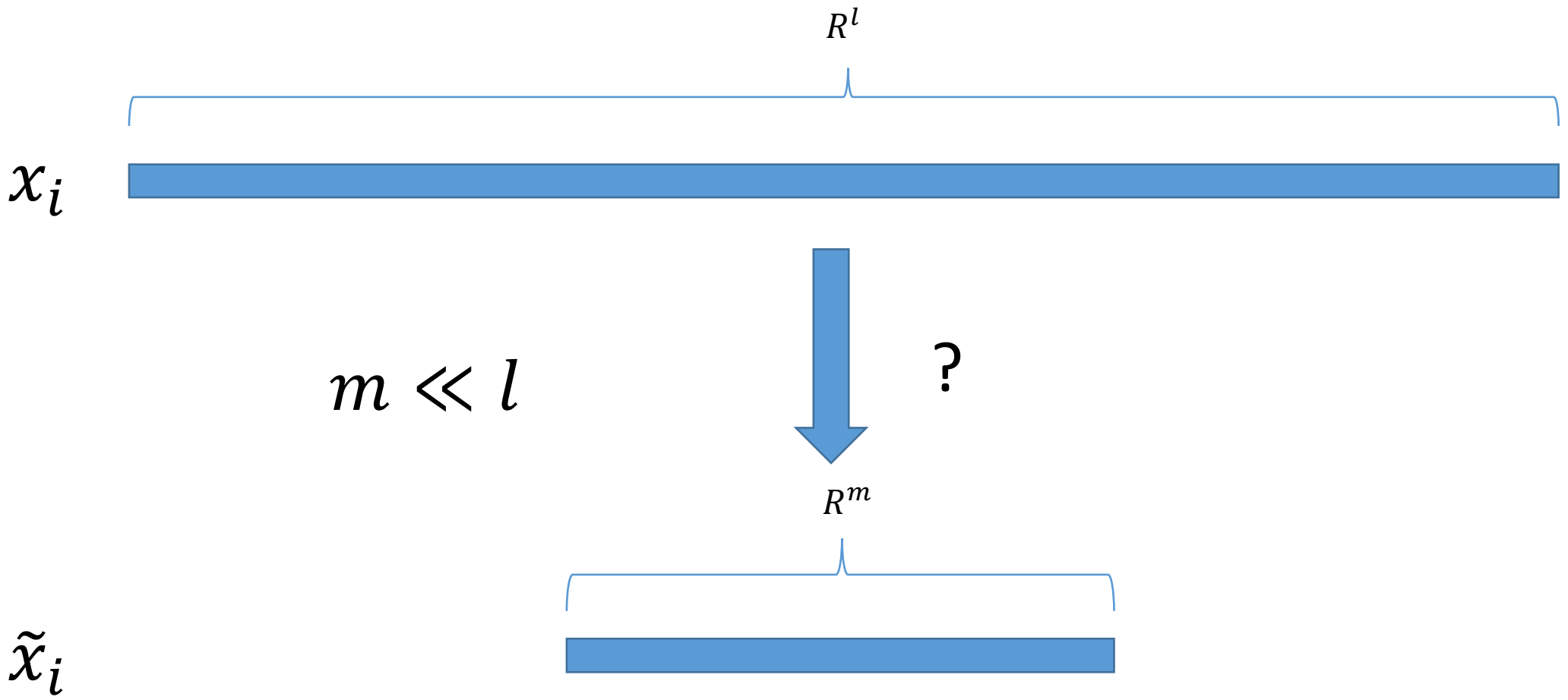
Yong Zhuang (yongzhua)

Notations

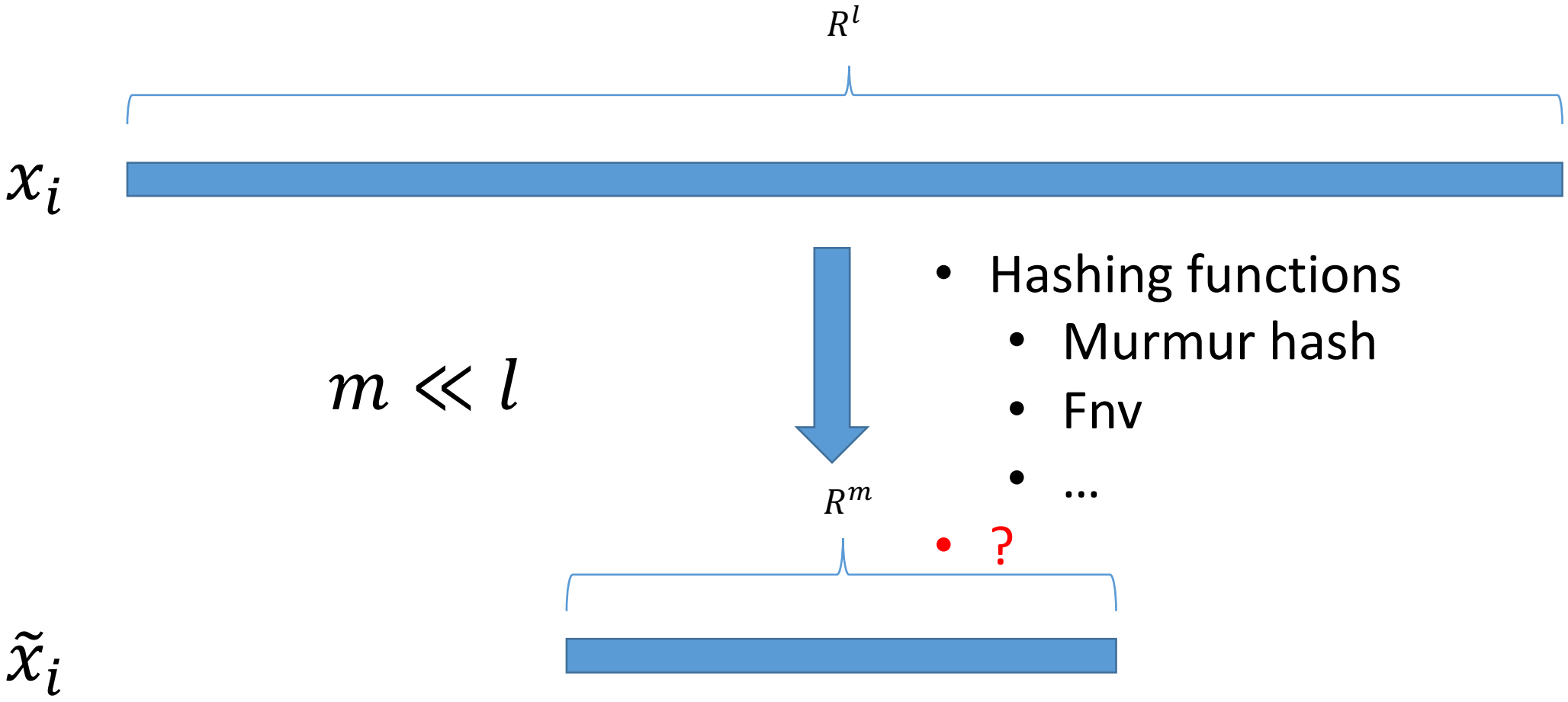
- Assume that we have a training data $x_1, x_2, \ldots, x_n \in R^l$ where l is the dimension of features and n is the number of instances.
- l can be millions, billions, or even more.

- If the feature dimension is too large, then we may not be able to train our model efficiently or even store the model into the memory.

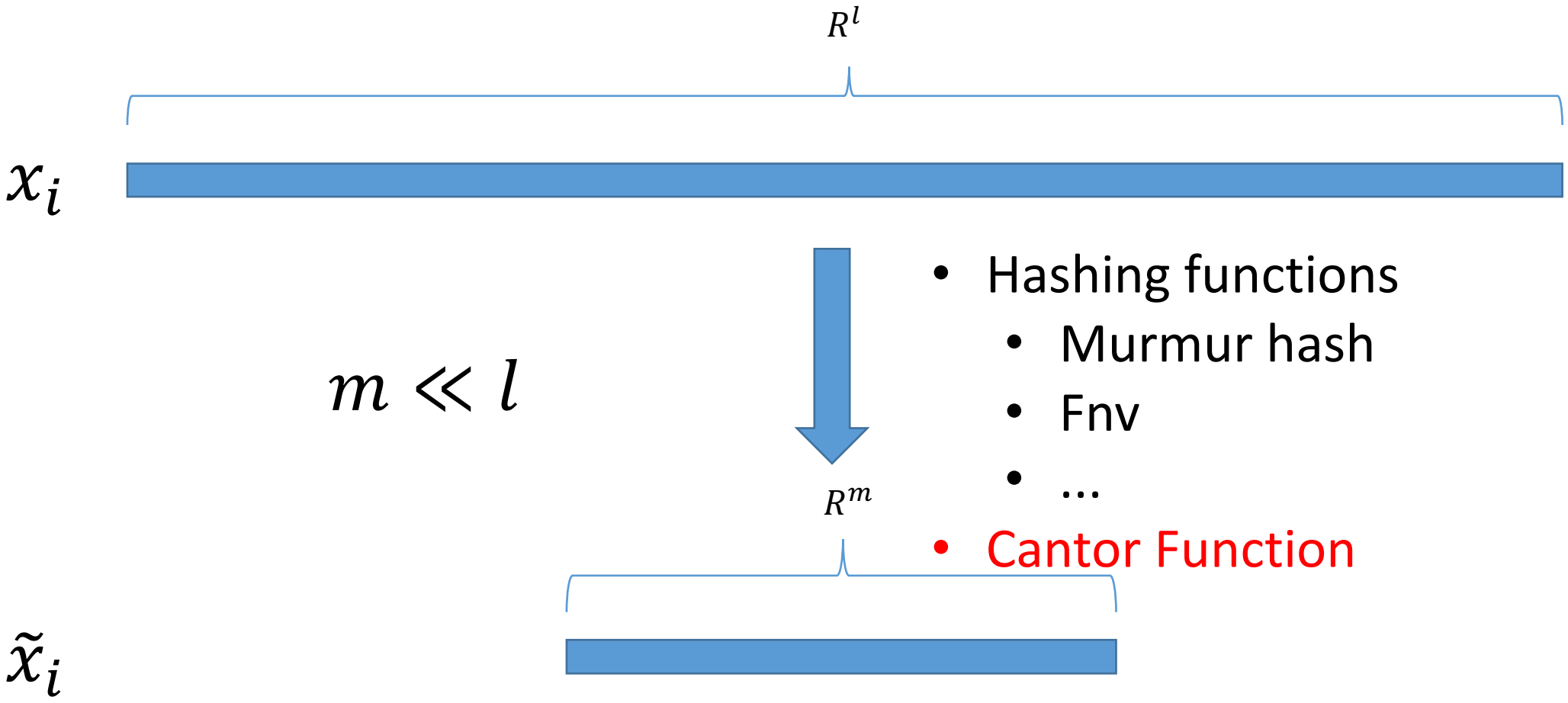
Problem



Possible Approaches



What I have done?



- Hashing functions
  - Murmur hash
  - Fnv
  - …
- Cantor Function

Experiments

		model size	time	#dot	#axpy	#epochs	#acc	#logloss
criteo	direct	none	none	none	none	none	none	none
criteo	dict	none	none	none	none	none	none	none
criteo	cantor	17777212	38336.6	1114399972	602461996	28	79.0824	0.44844
criteo	murmur	17777212	55423.9	1114399972	602461654	28	79.0783	0.44843
criteo	fnv	17777212	56458.5	1114399972	602423787	28	79.0818	0.44844
criteo	disk	none	none	none	none	none	none	none
avazu-app	direct	none	none	none	none	none	none	none
avazu-app	dict	5201110	9293.9	303412464	164562773	24	87.1241	0.33112
avazu-app	cantor	17777205	1826.6	316054650	165425480	25	87.1212	0.33107
avazu-app	murmur	17777205	2453.1	303412464	164599487	24	87.1182	0.33152
avazu-app	fnv	17777205	2698.6	278128092	154413155	22	87.1201	0.33124
avazu-app	disk	4247842	1171.2	455118696	250027488	36	87.0854	0.33142
avazu-site	direct	none	none	none	none	none	none	none
avazu-site	dict	7143563	16205.9	659899604	338412347	28	80.6077	0.43672
avazu-site	cantor	17777177	2410.7	565628232	312355462	24	80.6114	0.43663
avazu-site	murmur	17777177	3914.0	565628232	312333409	24	80.6043	0.43674
avazu-site	fnv	17777177	4144.0	612763918	335252201	26	80.6087	0.43683
avazu-site	disk	6194921	3302.0	966281563	584921794	41	80.5648	0.43787

Statistical Properties of Cantor Function

- $\phi: \mathbb{N} \times \mathbb{N} \longrightarrow \mathbb{N}$
- $\phi(x_1, x_2) = \frac{1}{2}(x_1 + x_2)(x_1 + x_2 + 1) + x_2$ where $x_1, x_2 \in \mathbb{N}$.

- It is an biased estimator.
- However, both of bias and variance decrease to 0 as m decreases.

Thanks

Machine Learning Methods for Causal Inference

Evan Sherwin | Rahul Ladhania

CCML S17

May 2, 2017

Prediction v. causal prediction

- Traditional prediction asks, what is $P(Y|X = x)$?

Prediction v. causal prediction

- Traditional prediction asks, what is $P(Y|X = x)$?
- Causal prediction asks, what is $P(Y|do(X = x))$?
  - What happens to Y when X is manipulated

Causality and machine learning

- Machine learning literature tends to focus on prediction tasks

Causality and machine learning

- Machine learning literature tends to focus on prediction tasks
- Growing literature incorporates various forms of machine learning into causal inference

Causality and machine learning

- Machine learning literature tends to focus on prediction tasks
- Growing literature incorporates various forms of machine learning into causal inference
    - Graphical models for causal discovery and inference

Causality and machine learning

- Machine learning literature tends to focus on prediction tasks
- Growing literature incorporates various forms of machine learning into causal inference
  - Graphical models for causal discovery and inference
  - ML-based propensity scores to compare similar treatment and control groups

Causality and machine learning

- Machine learning literature tends to focus on prediction tasks
- Growing literature incorporates various forms of machine learning into causal inference
  - Graphical models for causal discovery and inference
  - ML-based propensity scores to compare similar treatment and control groups
  - ML for model specification under selection on observables
  - Learning unobserved features from other features

Causality and machine learning

- Machine learning literature tends to focus on prediction tasks
- Growing literature incorporates various forms of machine learning into causal inference
  - Graphical models for causal discovery and inference
  - ML-based propensity scores to compare similar treatment and control groups
  - ML for model specification under selection on observables
  - Learning unobserved features from other features
  - Heterogeneous treatment effects

Causal estimands of interest

- For each covariate profile x, the conditional average treatment effect (CATE) is defined as:

$$\tau_{CATE}(x) = \mathbb{E}[Y(1) - Y(0)|X = x]$$

Causal estimands of interest

- For each covariate profile x, the conditional average treatment effect (CATE) is defined as:

$$\tau_{CATE}(x) = \mathbb{E}[Y(1) - Y(0)|X = x]$$

- More flexible marginal conditional average treatment effect (MCATE) defined as

$$\tau_{MCATE}(x) = \int \mathbb{E}[Y(1) - Y(0)|(X^1, X^2, ...X^S = x^S, ..X^d)]dF_{X^{-S}|X^S = x^S}$$

Causal estimands of interest

- For each covariate profile x, the conditional average treatment effect (CATE) is defined as:

$$\tau_{CATE}(x) = \mathbb{E}[Y(1) - Y(0)|X = x]$$

- More flexible marginal conditional average treatment effect (MCATE) defined as

$$\tau_{MCATE}(x) = \int \mathbb{E}[Y(1) - Y(0)|(X^1, X^2, ...X^S = x^S, ..X^d)]dF_{X^{-S}|X^S=x^S}$$

- New class which evaluates treatment effect by comparing potential outcomes' distribution functions: Distributional Average Treatment Effect (DATE):

$$\tau_{DATE}(S) = Div(F_{Y(1)|S}, F_{Y(0)|S})$$

ML methods for heterogeneous treatment effects

- Recent & forthcoming literature provide data-driven methods for investigation & potential discovery of sub-populations:
  - Sparse regression models (LASSO, ridge regression, elastic net)
    - Restrictive assumptions & limitations of (linear) regression

ML methods for heterogeneous treatment effects

- Recent & forthcoming literature provide data-driven methods for investigation & potential discovery of sub-populations:
  - Sparse regression models (LASSO, ridge regression, elastic net)
    - Restrictive assumptions & limitations of (linear) regression
  - Tree-based methods - recursively partitioning data into homogeneous sub-populations
    - Greedy partitioning; unstable; discontinuous approximations
    - Ensemble methods (BART, random forests) improve upon single tree model

ML methods for heterogeneous treatment effects

- Recent & forthcoming literature provide data-driven methods for investigation & potential discovery of sub-populations:
  - Sparse regression models (LASSO, ridge regression, elastic net)
    - Restrictive assumptions & limitations of (linear) regression
  - Tree-based methods - recursively partitioning data into homogeneous sub-populations
    - Greedy partitioning; unstable; discontinuous approximations
    - Ensemble methods (BART, random forests) improve upon single tree model
  - Treatment Effect Subset Scan (TESS) by Mcfowland et al
    - Frame identification as pattern detection problem; maximize a nonparametric scan statistic over all sub-populations, while being parsimonious in which effects to estimate

Detailed theoretical results in paper.

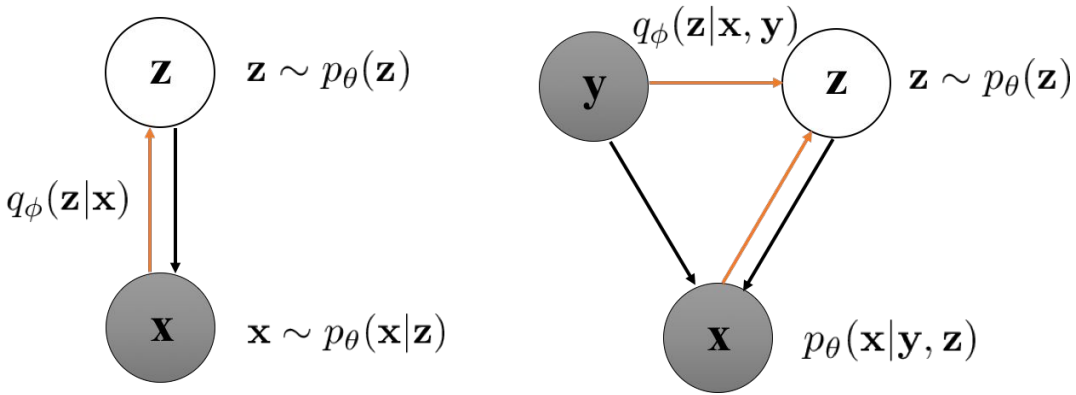
Thank you.

A survey of Variational AutoEncoders (VAEs) and the variants

Yuanyuan Feng & Hongyu Zhu

Motivation

- Generative models in deep neural network -- GANs and VAEs.
- VAEs can be interpreted from both neural network formulation (encoder/decoder) and graphical model (inference) perspectives -- mathematically interesting.
- Efficiently approximate intractable (posterior) distribution -- Maximize lower bound objectives.



Related work I -- VAE & Conditional VAE

- VAE: Kingma & Welling (2013) -- The intractable posterior inference can be made especially efficient from a recognition model using a reparametrization trick.

$$\log p_\theta(\mathbf{x}_i) \geq \mathcal{L}(\mathbf{x}_i; \theta, \phi) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x}_i)} \left[\log p_\theta(\mathbf{x}_i|\mathbf{z})\right] - \mathrm{KL}\left(q_\phi(\mathbf{z}|\mathbf{x}_i)||p_\theta(\mathbf{z})\right).$$

- CVAE: Kingma, Rezende & Mahamed (2014) -- Semi-supervised learning with deep generative models and performs conditional generation on MNIST dataset.

$$\log p_\theta(\mathbf{y}|\mathbf{x}) \geq \mathcal{L}(\mathbf{x}, \mathbf{y}; \theta, \phi) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})} \left[\log p_\theta(\mathbf{y}|\mathbf{x}, \mathbf{z})\right] - \mathrm{KL}\left(q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y})||p_\theta(\mathbf{z}|\mathbf{x})\right).$$

Related work II -- Denoising VAE & Adversarial Autoencoders

- DVAE: Im, et. al (2015) -- Denoising VAEs are trained with noise injected in their stochastic hidden layer. A modified training criterion which corresponds to a tractable lower bound is proposed when the input data is corrupted.

$$\log p_\theta(\mathbf{x}) \geq \mathcal{L}_{dvae} \stackrel{def}{=} \mathbb{E}_{\tilde{q}_\phi(\mathbf{z}|\mathbf{x})}\left[\log \frac{p_\theta(\mathbf{x},\mathbf{z})}{q_\phi(\mathbf{z}|\tilde{\mathbf{x}})}\right].$$

- AAE: Makhzani, et. al (2015) -- uses GAN to perform variational inference by matching the aggregated posterior of the hidden layer with an arbitrary prior distribution.

$$q(\mathbf{z}|\mathbf{x}) = \int_\eta q(\mathbf{z}|\mathbf{x},\eta)p_\eta(\eta)d\eta \Rightarrow q(\mathbf{z}) = \int_{\mathbf{x}}\int_\eta q(\mathbf{z}|\mathbf{x},\eta)p_d(\mathbf{x})p_\eta(\eta)d\eta d\mathbf{x}.$$

Robust Estimation of Regression Coefficients

Fan Jiang, Yangyi Lu

May 2nd, 2017

Least Squares Estimator

Multiple linear regression model and the ordinary square estimator for **β**:

$$Y = X\beta + \epsilon \qquad \hat{\beta} = (X^T X)^{-1} X^T Y$$

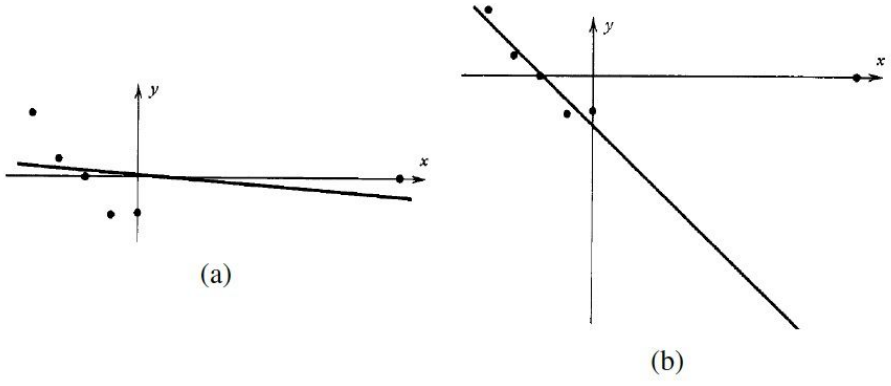


Figure 1: Comparison of two linear regression done by least square estimation

OLS works well only under strict conditions and assumptions.

What if:

- wrong observations in the dataset occur?
- assumptions are incorrect ($E[X_i \epsilon_i] \neq 0$)

Misleading and totally damaged!

Robust Estimator Measure

Two measures for the robustness of estimators (T):

Influence function: the dependence of the estimator on the value of one of the points in the sample.

$$n \cdot (T_n(x_1, \ldots, x_{i-1}, x, x_{i+1}, \ldots, x_n) - T_n(x_1, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_n))$$

Breakdown points: the proportion of incorrect observations an estimator can handle before giving an incorrect result.

$$\epsilon_n^*(T, D) := \min \left\{ \frac{m}{n};\ bias(m, T, D)\ is\ infinite \right\}$$

Robust Estimator - M-estimator

M - estimator: automatically normal distributed

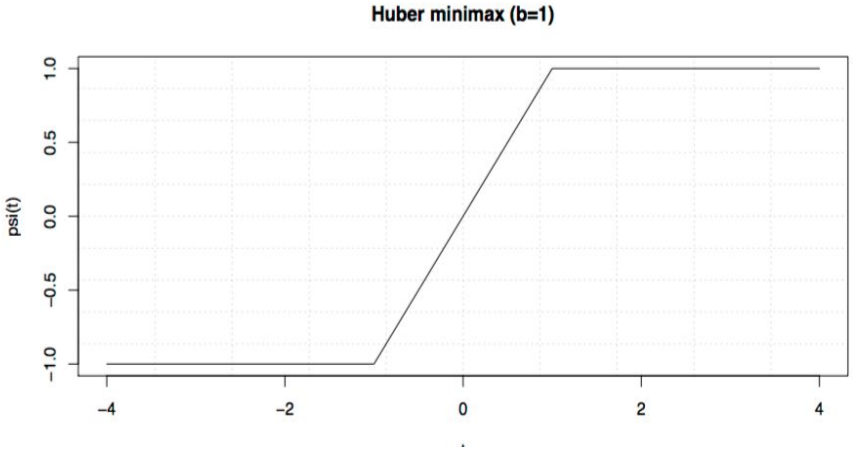
$$\hat{\beta}^{(M)} = argmin_{\beta \in R^P} \sum_{i=1}^{n} \rho\left(\frac{r_i(\beta)}{\hat{\sigma}}\right) \qquad r_i(\beta) = y_i - \sum_j x_{ij}\theta_j$$

σ: scale variable

Huber minimax M-estimator

$$\psi(t) = \begin{cases} t & \text{if } t < b \\ b\,\text{sgn}(t) & \text{if } t \geq b \end{cases}$$

where b is a constant.



e.g. for the robust function in M - estimator

M - estimator has bounded influence function according to Y, but the breakdown point is still super low, which is 0%.

Generalized m-estimator

Generalized M-estimators are introduced in order to bound the influence function of outlying X by means of some weight function w.

$$\hat{\beta}^{(GM)} = argmin_{\beta \in R^P} \sum_{i=1}^{n} w(X_i) \frac{\rho(r_i(\beta))}{\hat{\sigma}}$$

Computationally efficient by **iterated reweighted least square** method:

The first elementary estimate $\hat{\beta}^{(OLS)}$ of β^0.

Count the residuals $r_i(\hat{\beta}) = Y_i - \hat{Y}_i = Y_i - X_i^T \hat{\beta} \quad i = 1 \ldots n.$

Count the estimate $\hat{\sigma}$ of σ.

(e.g. *MAD*: $\hat{\sigma} = 1.4826 \underset{i}{\text{median}} \left(\left| r_i - \underset{j}{\text{median}}(r_j) \right| \right)$)

Count the weights w_i.

(e.g. Andrew's ψ function: $w_i = \frac{\psi(\frac{r_i}{\hat{\sigma}})}{\frac{r_i}{\hat{\sigma}}}$)

Update the estimate $\hat{\beta}$ by performing a weighted least squares with the weights w_i

Calculate $\hat{\beta}^{(WLS)} = (\mathbf{X}^T \mathbf{W} \mathbf{X})^{-1} \mathbf{X}^T \mathbf{W} \mathbf{Y}$

Finally, repeat from the second step until converge

Other Robust Estimator

- G-M-estimator has bounded influence function of outlying X, but no improvements to the breakdown points.

- Other robust estimators with better breakdown points results:
  - **MM-estimators**: high-breakdown and high-efficiency estimators, where the initial estimate is obtained with an S-estimator, and it is then improved with an M-estimator.
  - **Least median of squares (LMS)**: 50% breakdown point estimator,

$$\hat{\beta}^{(LMS)} = argmin_{\beta \in R^P} \left(med_i(r_i^2(\beta)) \right)$$

Project report for 36-702: Adaptive smoothing spline

Manjari Das

Carnegie Mellon University
Department of Statistics

May 2, 2017

Introduction

Smoothing splines bring some flexibility to regression compared to linear and polynomial regression

Introduction

Smoothing splines bring some flexibility to regression compared to linear and polynomial regression

Conventionally the roughness is penalized uniformly

Introduction

Smoothing splines bring some flexibility to regression compared to linear and polynomial regression

Conventionally the roughness is penalized uniformly

Further flexibility can be incorporated by letting the penalty vary by roughness of the fitted curve

Recent works

[1] Wang, X., Du, P. & Shen, J. (2013), Śmoothing splines with varying smoothing parameter; *Biometrika* **100**, 4.

[2] Liu, Z. & Guo, W. (2010), Ḍata driven adaptive spline smoothing; *Statistica Sinica* **20**, 114363.

[3] Pintore, A., Spekman, P. & Holmes, C. C. (2006), Śpatially adaptive smoothing splines; *Biometrika* **93**, 112-25

Set up

$$y_i = f_0(t_i) + \sigma(t_i)\epsilon_i, i = 1, 2, ..., n$$

t_i are design points on [0,1]
ϵ_i are i.i.d D(0,1)
$\sigma(.)$ is variance function
f_0 is true regression function

Problem

Minimize

$$\frac{1}{n}\sum_{i=1}^{n}\sigma^{-2}(t_i)\{y_i - f(t_i)\}^2 + \lambda\int_0^1 \{f^{(m)}(t)\}^2 dx$$

Modified problem for adaptive penalty

$$\Psi(f) = \frac{1}{n}\sum_{i=1}^{n} \sigma^{-2}(t_i)\{y_i - f(t_i)\}^2 + \lambda \int_0^1 \rho(t)\{f^{(m)}(t)\}^2 dt,$$

$\lambda > 0$
$\rho : [0,1] \longrightarrow (0,\infty)$
$f^{(i)}$ absolutely continuous for i upto $m-1$
$f^{(m)} \in L_2[0,1]$

Necessary and sufficient condition

$$\omega_n(t) = \frac{1}{n} \sum_{i=1}^{n} I(t \leq t_i)$$

Necessary and sufficient condition

$$\omega_n(t) = \frac{1}{n}\sum_{i=1}^{n} I(t \le t_i)$$
$$\check{I}_I(f,t) = \int_0^t \sigma^{-2}(s) f(s) d\omega_n(s),$$

Necessary and sufficient condition

$$\omega_n(t) = \frac{1}{n}\sum_{i=1}^{n} I(t \le t_i)$$
$$\check{I}_l(f,t) = \int_0^t \sigma^{-2}(s) f(s) d\omega_n(s),$$
$$\check{I}_k(f,t) = \int_0^t \check{I}_{k-1}(f,s) ds,\ (2 \le k \le m)$$

Necessary and sufficient condition

$$\omega_n(t) = \frac{1}{n}\sum_{i=1}^{n} I(t \le t_i)$$
$$\check{I}_l(f,t) = \int_0^t \sigma^{-2}(s) f(s) d\omega_n(s),$$
$$\check{I}_k(f,t) = \int_0^t \check{I}_{k-1}(f,s) ds,\ (2 \le k \le m)$$
$$h(t_i) = y_i$$

Necessary and sufficient condition

$$\omega_n(t) = \frac{1}{n}\sum_{i=1}^{n} I(t \le t_i)$$
$$\check{I}_l(f,t) = \int_0^t \sigma^{-2}(s) f(s) d\omega_n(s),$$
$$\check{I}_k(f,t) = \int_0^t \check{I}_{k-1}(f,s) ds,\ (2 \le k \le m)$$
$$h(t_i) = y_i$$

Theorem 1. Necessary and sufficient conditions for $\hat{f} \in W_2^m$ to minimize ψ are that

$$(-1)^m \lambda \rho(t) \hat{f}^{(m)}(t) + \check{I}_m(\hat{f},t) = \check{I}_m(h,t),\ \text{a.e}$$

and

$$\check{I}_k(\hat{f},1) = \check{I}_k(h,1), (k = 1,...,m)$$

Corollary

Corollary 1. $n^{2m/(4m+1)}\{\hat{f}(t) - f_0(t)\}$ converges to

$$N[(-1)^{m-1} r(t)\{\rho(t) f_0^{(m)}(t)\}^{(m)}, L_0 r(t)^{1-1/(2m)} \rho(t)^{-1/(2m)}]$$

in distribution.

Corollary

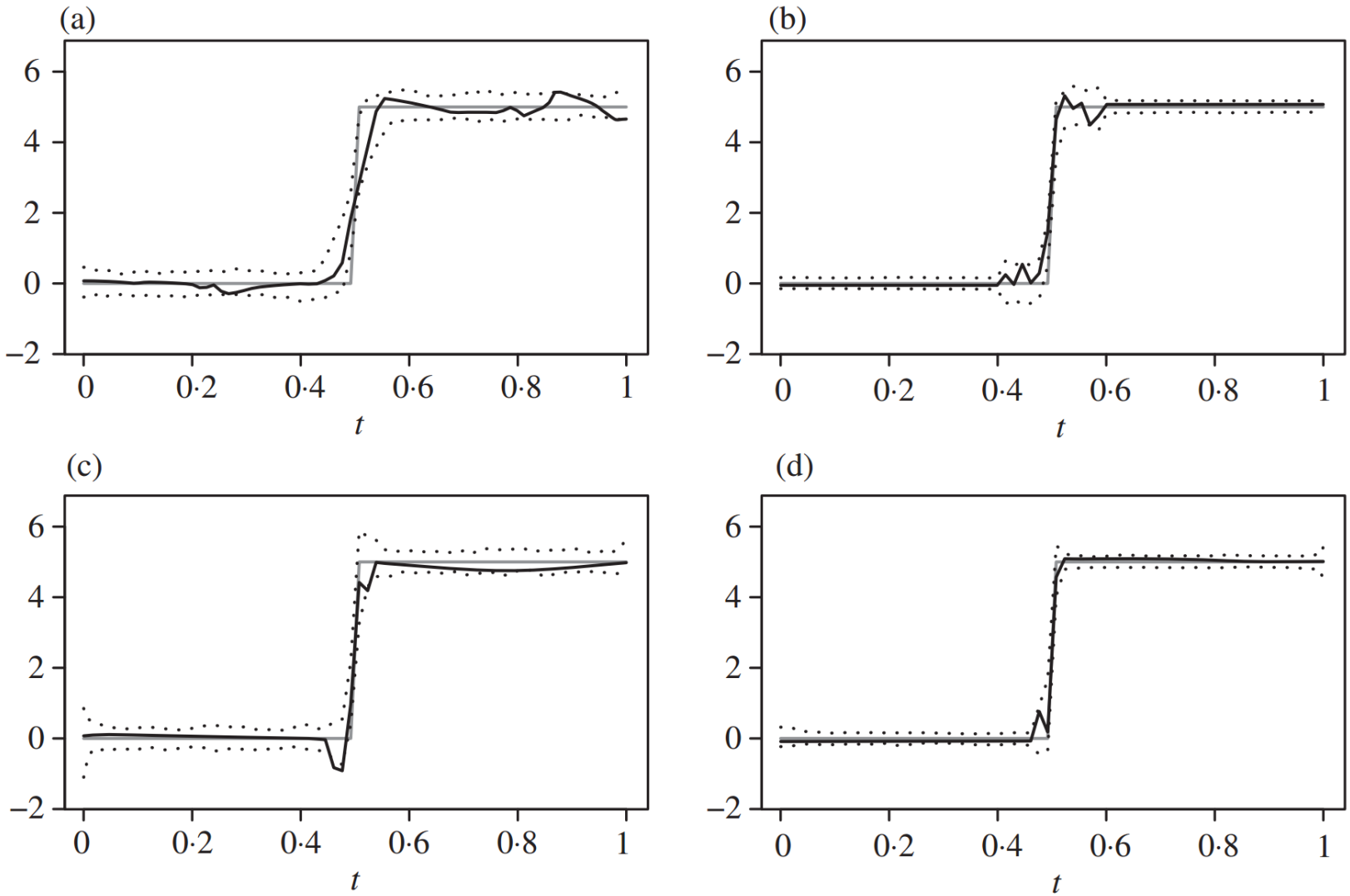
Corollary 1. $n^{2m/(4m+1)}\{\hat{f}(t) - f_0(t)\}$ converges to

$$N[(-1)^{m-1} r(t)\{\rho(t) f_0^{(m)}(t)\}^{(m)}, L_0 r(t)^{1-1/(2m)} \rho(t)^{-1/(2m)}]$$

in distribution.

$\lambda^{opt} = n^{-2m/(4m+1)}$
Minimize integrated MSE $\longrightarrow \rho(t)$

Justification of method with simulation



Thank You

On monotonic improvement guarantees and sampling efficiency for approximate policy gradient methods

Lisa Lee (lslee@cs.cmu.edu)

Carnegie Mellon University

May 2, 2017

Motivation

We focus on **policy gradient (PG) methods**, a popular class of reinforcement learning algorithms that have been at the heart of significant advances in AI and robotics.

A **policy** π describes how an agent will act when in some state s. The goal is to find an optimal policy that maximizes the **expected cumulative reward**,

$$\eta(\pi) := \mathbb{E}_{\tau \sim \pi} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t) \right]$$

where the expectation is taken over trajectories $\tau := (s_0, a_0, s_1, a_1, \ldots)$.

- PG methods directly optimize $\eta(\pi_\theta)$ by estimating its gradient w.r.t. the policy parameters θ.
- They are appealing because they **reduce RL to stochastic gradient descent**.

Motivation

Main challenges of PG methods:

1. Difficulty of obtaining **stable and steady improvement** despite the nonstationarity of the incoming data
2. High **sample complexity**

Monotonic improvement guarantees

Natural gradient method (Kakade, 2001)

Kakade (2001) provided a **natural gradient method** for policy iteration that has **guaranteed performance improvement**.

- Moves toward choosing a greedy optimal action rather than just a good action.
- Represents the steepest descent direction based on the underlying structure of the parameter space.

Other works (Kakade 2002, Schulman 2015) build off of this paper.

Monotonic improvement guarantees

Optimizing a lower bound on η (Schulman et al., 2015))

Thm (Schulman et al., 2015)

For stochastic policies π, $\tilde{\pi}$,

$$\eta(\tilde{\pi}) \geq L_{\pi}(\tilde{\pi}) - CD_{KL}^{\max}(\pi,\ \tilde{\pi}) \tag{1}$$

where $C := \frac{4\epsilon\gamma}{(1-\gamma)^2}$ and $\epsilon := \max_{s,a} |A_{\pi}(s,a)|$ is the max expected advantage.

Thus, we are guaranteed to improve the true objective η by optimizing the lower bound

$$\max_{\theta} \left[L_{\pi_{\theta_{\text{old}}}}(\pi_{\theta}) - CD_{\text{KL}}^{\max}(\pi_{\theta_{\text{old}}},\ \pi_{\theta}) \right] \tag{2}$$

Trust Region Policy Optimization (Schulman et al., 2015) is an approximate algorithm for optimizing Eq. (2).

High sample complexity problem

Bias vs. variance tradeoff

Bias vs. variance tradeoff of the policy gradient estimator $\hat{g} \approx \nabla_\theta \eta(\pi_\theta)$:

$$\theta \leftarrow \theta + \epsilon \hat{g}$$

- High variance necessitates using more samples.
- High bias can cause the algorithm to fail to converge, or to converge to a poor solution that is not even a local optimum.

High sample complexity problem

Variance reduction techniques

Overview:

- Unbiased estimators (Williams, 1992, Sutton et al., 1999; Baxter & Bartlett, 2000) exhibit variance that scales unfavorably with the time horizon.
- Actor-critic methods (another class of PG methods) use a value function rather than the empirical returns, obtaining a $\hat{g}$ with lower variance but more bias.
- Using an exponentially weighted estimator of the advantage function (Schulman et al. 2016) has shown to significantly reduce variance while maintaining a tolerable level of bias.

$$\hat{g} = \frac{1}{N} \sum_{n=1}^{N} \sum_{t=0}^{\infty} \hat{A}_t^n \nabla_\theta \log \pi_\theta(a_t^n \mid s_t^n)$$

Thank you

Definitions

Define the following distance between two policies π, $\tilde{\pi}$:

$$D_{\mathrm{KL}}^{\max}(\pi,\ \tilde{\pi}) := \max_{s} D_{\mathrm{KL}}\left[\pi(\cdot \mid s) \,\|\, \tilde{\pi}(\cdot \mid s)\right]$$

where $D_{\mathrm{KL}}[p \,\|\, q] := \sum_i p_i \log \frac{p_i}{q_i}$ is the Kullback-Leibler divergence for discrete probability distributions p, q.

Definitions

We use the following standard definitions of the state-action value function Q_π, the value function V_π, and the advantage function A_π:

$$Q_\pi(s_t, a_t) = \mathbb{E}_{s_{t+1}, a_{t+1}, \ldots} \left[\sum_{l=0}^{\infty} \gamma^l r(s_{t+l}) \right] \tag{3}$$

$$V_\pi(s_t) = \mathbb{E}_{a_t, s_{t+1}, \ldots} \left[\sum_{l=0}^{\infty} \gamma^l r(s_{t+l}) \right] \tag{4}$$

$$A_\pi(s, a) = Q_\pi(s, a) - V_\pi(s) \tag{5}$$

where $a_t \sim \pi(a_t \mid s_t)$ and $s_{t+1} \sim P(s_{t+1} \mid s_t, a_t)$.

A useful identity for η

The following useful identity expresses the expected return of another policy $\tilde{\pi}$ in terms of the advantage over π, accumlated over timesteps:

Lem

Given two policies π, $\tilde{\pi}$,

$$\eta(\tilde{\pi}) = \eta(\pi) + \mathbb{E}_{\tau \sim \tilde{\pi}} \left[\sum_{t=0}^{\infty} \gamma^t A_\pi(s_t, a_t) \right] \tag{6}$$

$$= \eta(\pi) + \sum_s \rho_{\tilde{\pi}}(s) \sum_a \tilde{\pi}(a \mid s) A_\pi(s, a) \tag{7}$$

where $\rho_\pi(s) = \sum_{t=0}^{\infty} \gamma^t P(s_t = s \mid \tilde{\pi})$ is the (unnormalized) discounted visitation frequencies.

A local approximation to η

However, in the approximate setting, there will be some states s for which the expected advantage is negative, i.e., $\sum_a \tilde{\pi}(a \mid s) A_\pi(s, a) < 0$, due to estimation and approximation error. We introduce the following local approximation to $\eta(\tilde{\pi})$:

$$L_\pi(\tilde{\pi}) = \eta(\pi) + \sum_s \rho_\pi(s) \sum_a \tilde{\pi}(a \mid s) \gamma^t A_\pi(s, a) \tag{8}$$

which uses the visitation frequency ρ_π rather than $\rho_{\tilde{\pi}}$, ignoring changes in state visitation density due to changes in the policy.

- Difficult to choose a stepsize that works for the entire course of the optimization, especially because the statistics of the states and rewards changes
- Often, the policy prematurely converges to a nearly-deterministic policy with a suboptimal behavior.

We survey results on strong theoretical performance guarantees for reliable monotonic improvement.

Graph Based Semisupervised Learning

Mochong Duan[1]

[1]Carnegie Mellon University

May 2, 2017

This project aims to review some of the previous and current work that may give insights into how spectral graph theory works when applied in semi-supervised learning, particularly, graph Laplacian.

Bulllet points

- performance of graph laplacian
- the performance when hyperparameters of the similarity graph,
- transformation of graph laplacian and noise model need to be regularized which constrain the performance.

Graph representation of the data.

List of doing

- Graph Construction
- Injecting labels on a subset of vertices
- Infer labels on unlabeled vertices on the graph

Problem Setup

While we have $G = (V, E)$ which are vertices and edges,the observations as $\{x_i\}_{i=1}^{n}$,the edges denoted as W, which W_{ij} connected the points x_i and x_j. $\mathcal{Y}$ denote the label. And the vertices V can be partitioned as two sets as a label set V_{label} and an unlabel set $V_{unlabel}$ .The goal is to predict the label $\hat{\mathcal{Y}}_u$ for the unlabel vertices. And for the weighted matrix, we have assumptionsas

- $W_{ij} \geq 0, \forall i, j$ and $W_{ij} = W_{ji}, \forall i, j$
- no edge means $W_{ij} = 0$
- no self loops, which means $W_{ii} = 0, \forall 1 \leq i \leq n$