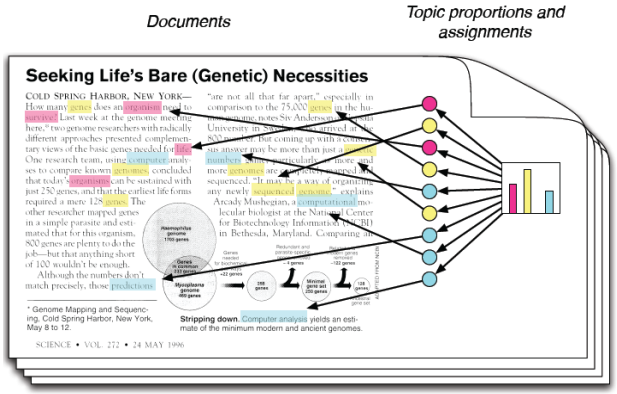
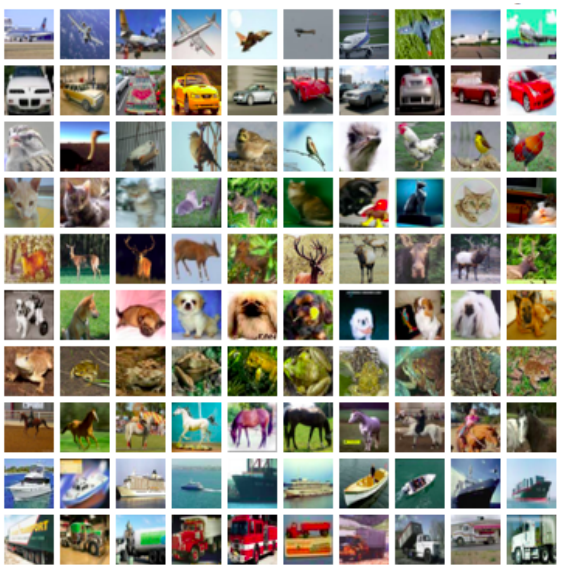
Distributed Estimation, Information Loss and Exponential Families

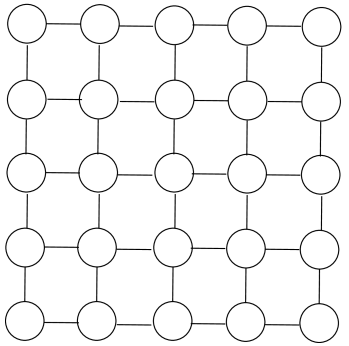
Qiang Liu
Department of Computer Science
Dartmouth College

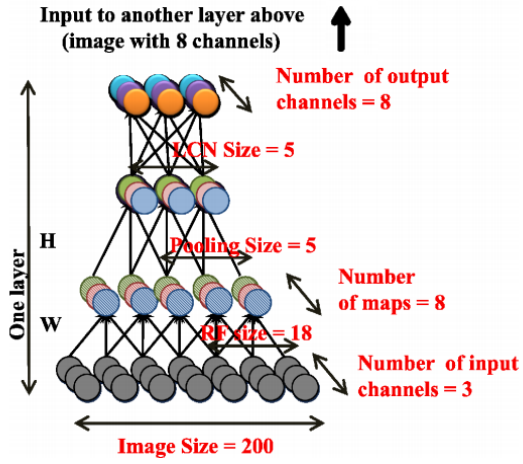
Statistical Learning / Estimation

- Learning generative models from data
  - Topic models (text), computer vision, bioinformatics …









Statistical Learning / Estimation

- Find a model (indexed by parameter) from a distribution family (or set) that fits the data best

Data (X), iid

$\{x^1, \ldots, x^n\}$

Best model $p(x|\theta^*)$ from

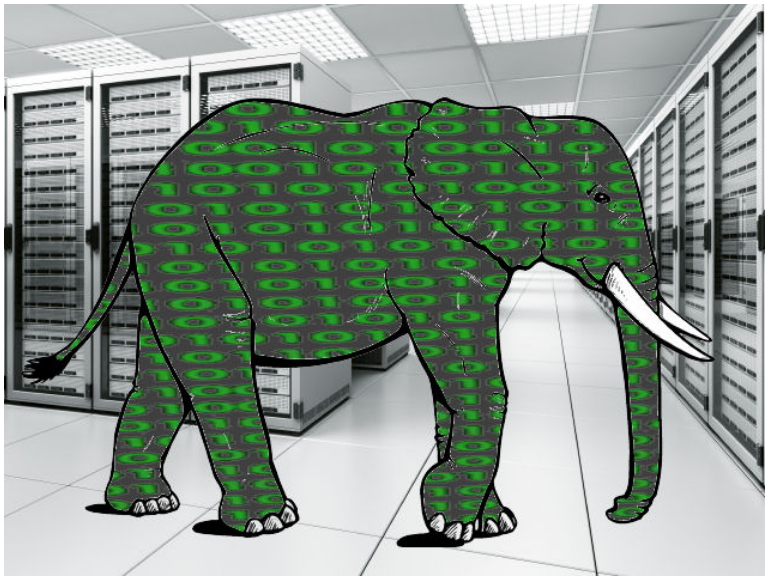
$\mathcal{P} = \{p(x|\theta), \theta \in \Theta\}$

- Maximum likelihood:

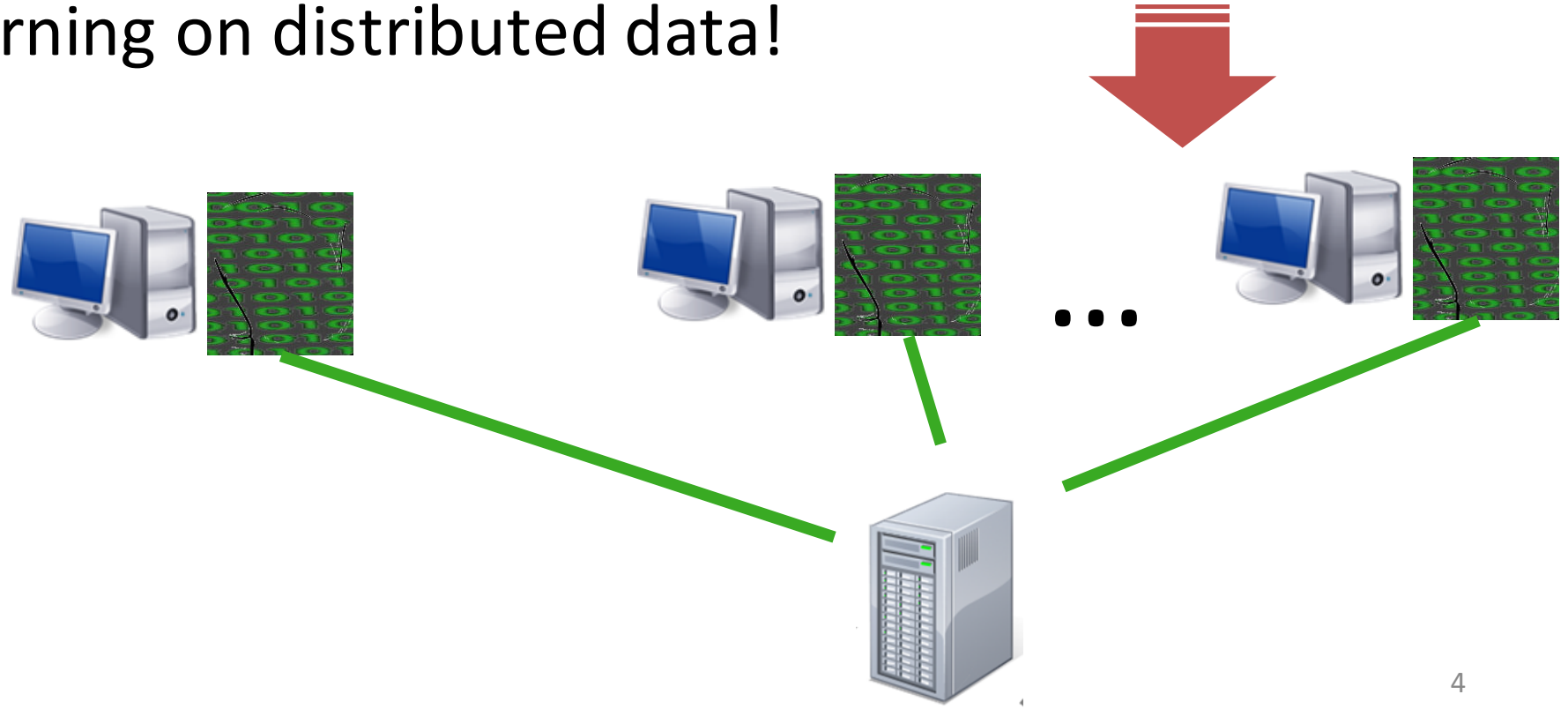
$$\hat{\theta}^{mle} = \arg\max_{\theta \in \Theta} \sum_{i}^{n} \log p(x^i|\theta)$$

  - Many other traditional methods …

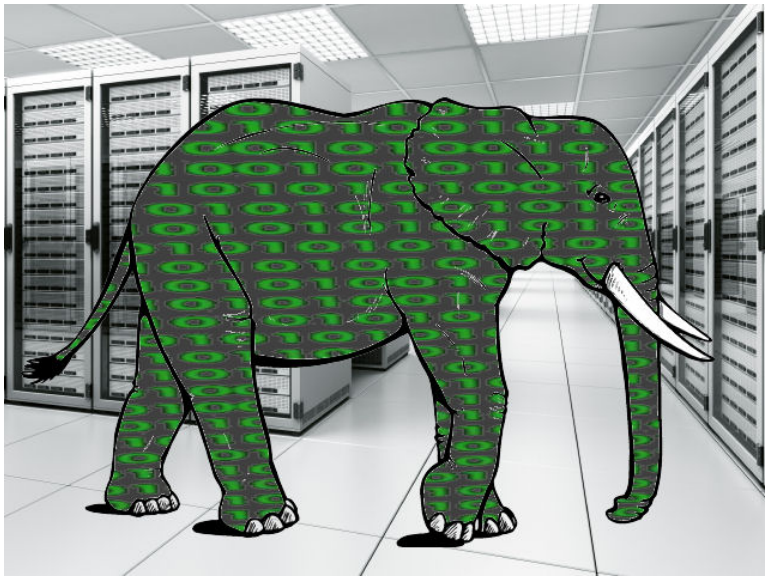
Challenge: Big Data



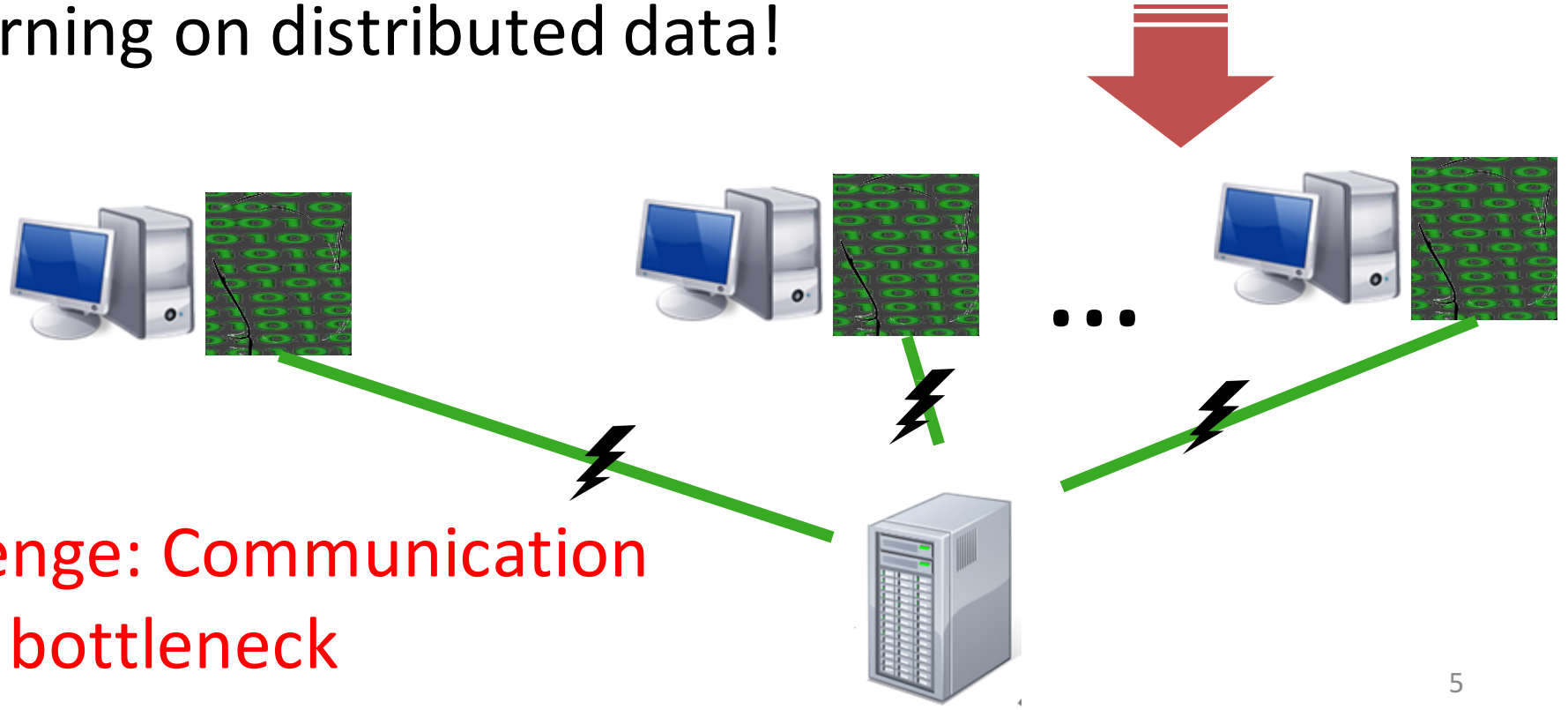
- Big data (very large n)
- Private data
- Can't be stored in a single machine!
- Learning on distributed data!



Challenge: Big Data



- Big data (very large n)
- Private data
- Can't be stored in a single machine!
- Learning on distributed data!



Challenge: Communication is the bottleneck

Big Data = Distributed Data

- Assumption: randomly partition the instances into d subsets evenly (n/d instances in each subset)

$$X = X^1 \cup X^2 \cup \cdots \cup X^d$$

- Traditional (centralized) MLE doesn't work in general

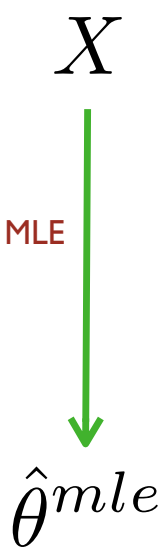
$$\hat{\theta}^{mle} = \arg\max_{\theta \in \Theta} \sum_{i}^{n} \log p(x^i|\theta)$$

- Existing methods: distributed implementation of MLE
  - Stochastic gradient descent (SGD), ADMM, etc…

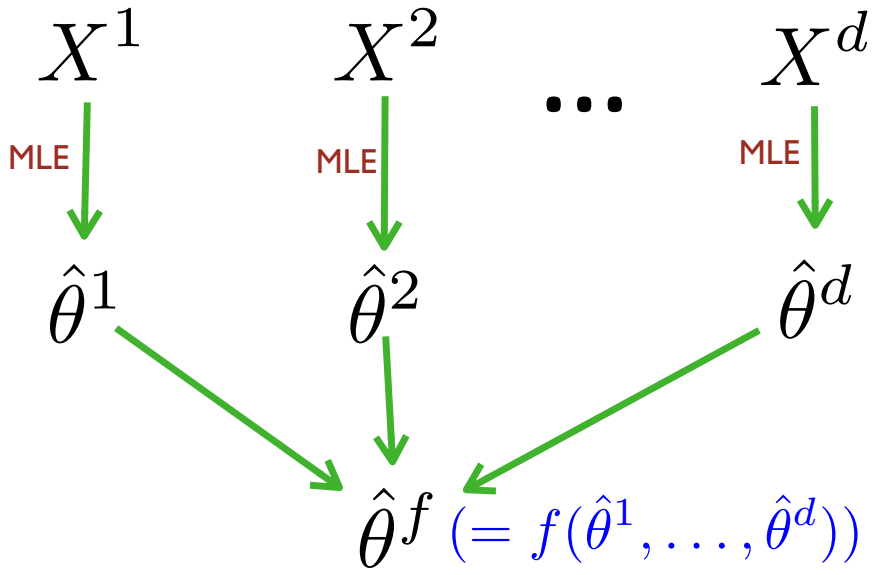
- **This talk: One-shot combine local MLEs**
  - Good approximation to global MLE
  - Much less communication

Combining local MLEs

Global MLE



Combine the local MLEs (this talk)



- Questions:
  - The best combination function f(…)?
  - How well the local MLEs approximates the global MLE?
  - Statistical loss by using local MLEs?

Linear Averaging (Naive)

- Take the linear averaging (*Zhang et al. 13; Scott et al.13*)

$$\hat{\theta}^{linear} = \frac{1}{d}\sum_{k}\hat{\theta}^{k}$$

Linear Averaging (Naive)

- Take the linear averaging (*Zhang et al. 13; Scott et al.13*)

$$\hat{\theta}^{linear} = \frac{1}{d}\sum_{k}\hat{\theta}^{k}$$

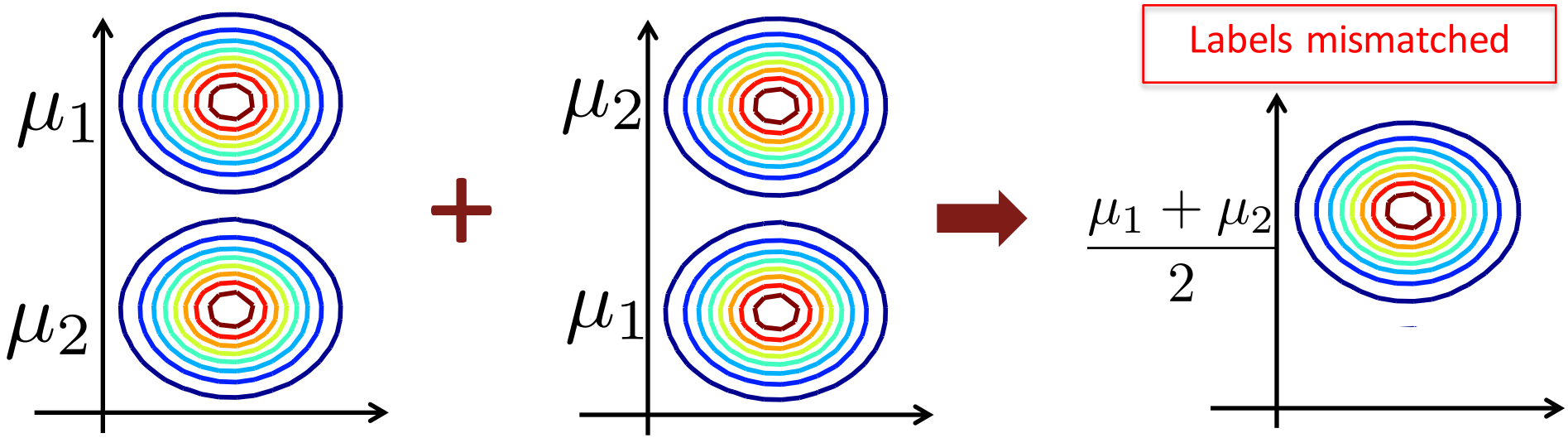
- Many disadvantages:
  - Doesn't work for discrete, non-additive parameters
  - Unidentifiable Parameters
    - Latent variables models: Gaussian mixture, topic models …

Linear Averaging (Naive)

- Take the linear averaging (*Zhang et al. 13; Scott et al.13*)

$$\hat{\theta}^{linear} = \frac{1}{d}\sum_{k}\hat{\theta}^{k}$$

- Many disadvantages:
  - Doesn't work for discrete, non-additive parameters
  - Unidentifiable Parameters
    - Latent variables models: Gaussian mixture, topic models …

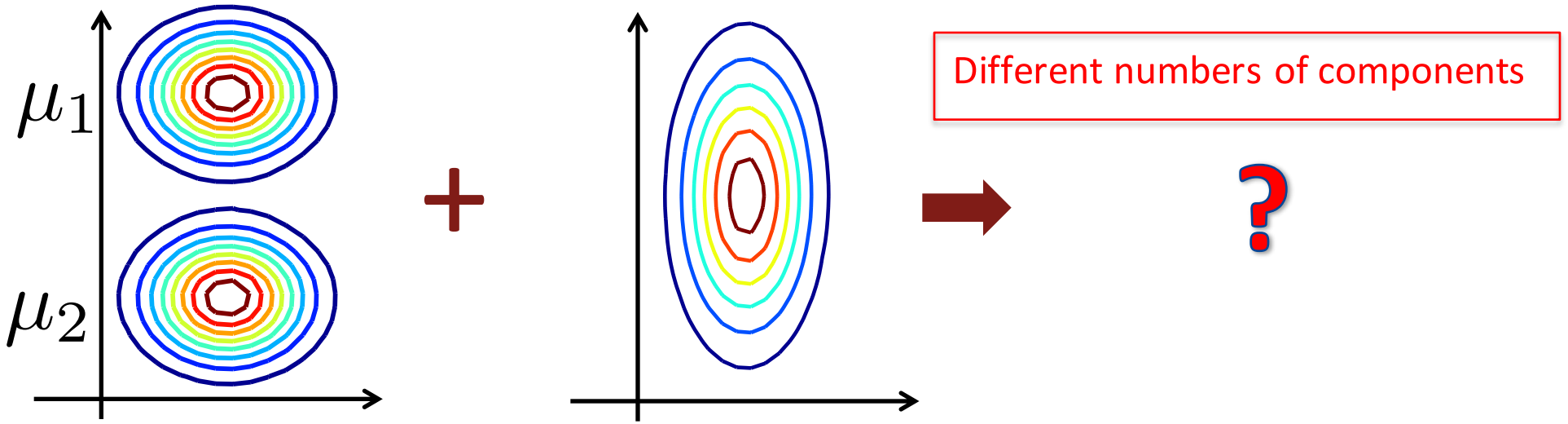


Linear Averaging (Naive)

- Take the linear averaging (*Zhang et al. 13; Scott et al.13*)

$$\hat{\theta}^{linear} = \frac{1}{d}\sum_{k}\hat{\theta}^{k}$$

- Many disadvantages:
  - Doesn't work for discrete, non-additive parameters
  - Parameter non-identifiability
    - Latent variables models: Gaussian mixture, topic models …



Linear Averaging (Naive)

- Take the linear averaging (*Zhang et al. 13; Scott et al.13*)

$$\hat{\theta}^{linear} = \frac{1}{d}\sum_{k}\hat{\theta}^{k}$$

- Many disadvantages:
  - Doesn't work for discrete, non-additive parameters
  - Parameter non-identifiability
    - Latent variables models: Gaussian mixture, LDA, deep nets
  - **Result changes with different parameterization**

For example:	$\theta = \sigma^2$	$\theta = \sigma$	$\theta = 1/\sigma$
	(variance)	(std)	(precision)
	$(\sigma_1^2 + \sigma_2^2)/2$	$(\sigma_1 + \sigma_2)/2$	$(1/\sigma_1 + 1/\sigma_2)/2$

Linear Averaging (Naive)

- Take the linear averaging (*Zhang et al. 13; Scott et al.13*)

$$\hat{\theta}^{linear} = \frac{1}{d}\sum_{k}\hat{\theta}^{k}$$

- Many disadvantages:
  - Doesn't work for discrete, non-additive parameters
  - Parameter non-identifiability
    - Latent variables models: Gaussian mixture, LDA, deep nets
  - Result changes with different parameterization
  - **Asymptotically: suboptimal error rate (explain later)**

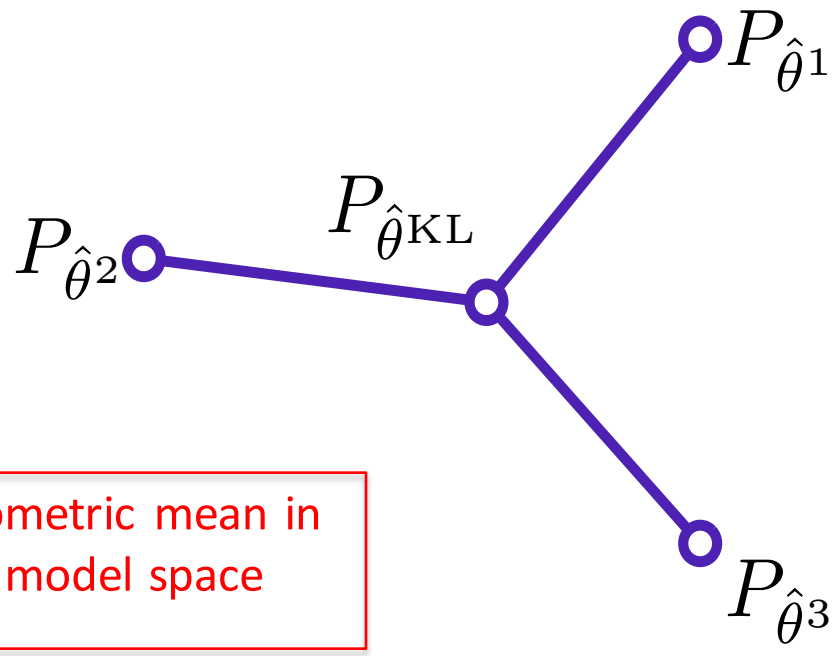
Our Algorithm: KL-Averaging

- We propose KL-averaging:

$$\hat{\theta}^{\mathrm{KL}} = \arg\min_{\theta \in \Theta} \sum_k \mathrm{KL}(P_{\hat{\theta}^k} || P_\theta)$$

Where KL divergence: $\mathrm{KL}(P_{\theta'} || P_\theta) = \int_x p(x|\theta') \log \frac{p(x|\theta')}{p(x|\theta)} dx$

– Idea: average the models, not the parameters!!



Geometric mean in the model space

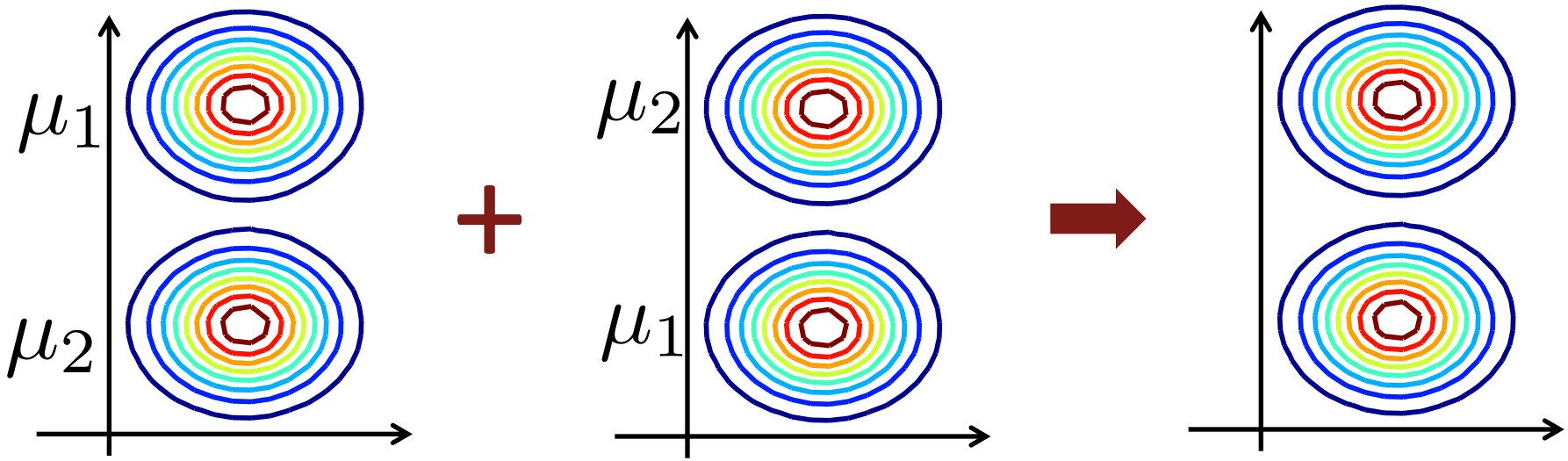
Our Algorithm: KL-Averaging

- We propose KL-averaging:

$$\hat{\theta}^{\mathrm{KL}} = \arg\min_{\theta \in \Theta} \sum_k \mathrm{KL}(P_{\hat{\theta}^k} || P_\theta)$$

Where KL divergence: $\mathrm{KL}(P_{\theta'} || P_\theta) = \int_x p(x|\theta') \log \frac{p(x|\theta')}{p(x|\theta)} dx$

– Addresses all the problems (non-additive, non-identifiabilities, reparameterization invariance…)



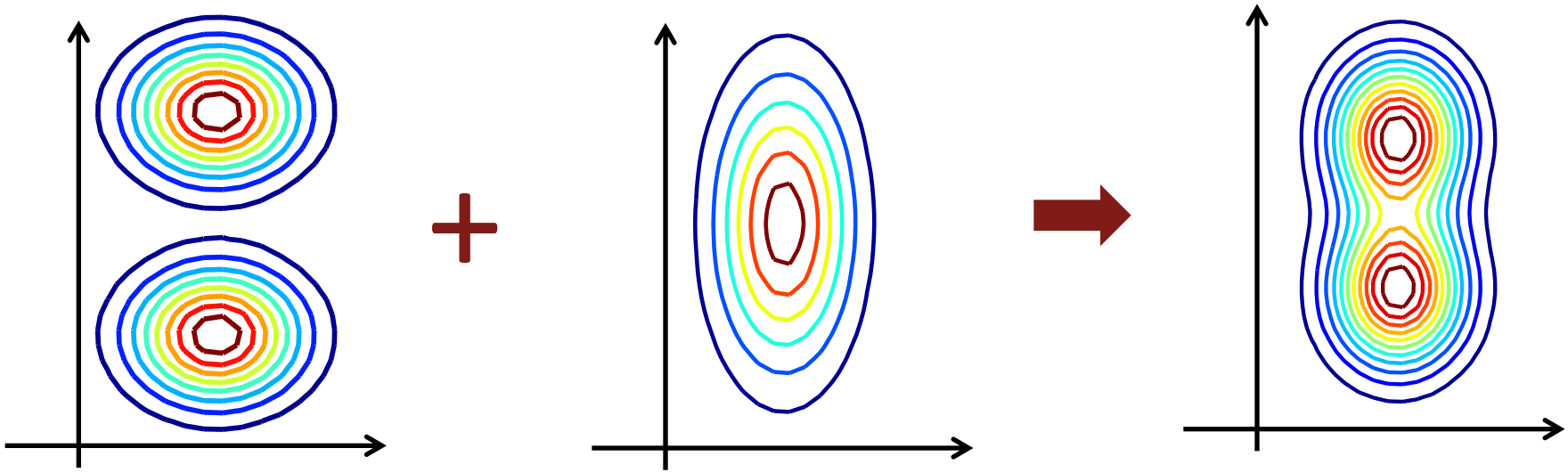
Our Algorithm: KL-Averaging

- We propose KL-averaging:

$$\hat{\theta}^{\mathrm{KL}} = \arg\min_{\theta \in \Theta} \sum_k \mathrm{KL}(P_{\hat{\theta}^k} || P_\theta)$$

Where KL divergence: $\mathrm{KL}(P_{\theta'} || P_\theta) = \int_x p(x|\theta') \log \frac{p(x|\theta')}{p(x|\theta)} dx$

– Addresses all the problems (non-additive, non-identifiabilities, reparameterization invariance…)



Parametric Bootstrap Interpretation

- A parametric bootstrap procedure:
  - Resample from the local model $p(x|\hat{\theta}^k)$

$$\tilde{X}^k \sim p(x|\hat{\theta}^k)$$

  - Maximum likelihood based on $\tilde{X} = [\tilde{X}^1, \ldots, \tilde{X}^d]$

$$\hat{\theta} = \arg\max_{\theta \in \Theta} \sum_k \log p(\tilde{X}^k|\theta)$$

- Reduces to $\hat{\theta}^{\mathrm{KL}}$ when the resample size goes to infinite

KL-averaging on Exponential Families

- (Full) exponential families

$$p(x|\theta) = \frac{1}{Z(\theta)} \exp[\theta^T \phi(x)] \qquad \Theta = \{\theta : Z(\theta) < \infty\}$$

θ: natural parameter

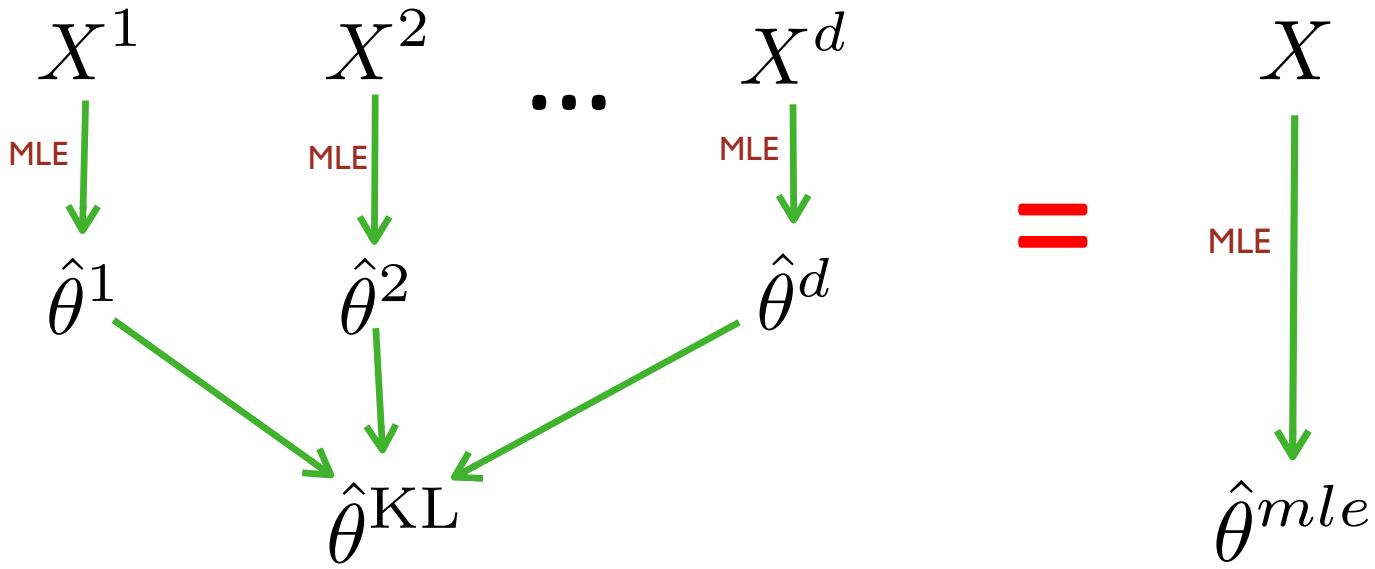
$\phi(x)$: sufficient statistics

$Z(\theta)$: normalization constant

- Include:
  - Gaussian, exponential, Gamma, Dirichlet, Bernoulli …
  - Most undirected graphical models (e.g., Ising model) …
- Not include:
  - Latent variables models (Gaussian mixture, topic models)
  - Hierarchical models, Gaussian process, (Parametric) Bayes nets …

KL-averaging on Exponential Families

- KL-averaging **exactly recovers** the global MLE under exponential families ($\hat{\theta}^{\mathrm{KL}} = \hat{\theta}^{mle}$)!
- Linear-averaging is not exact in general



KL-averaging on Exponential Families

- KL-averaging **exactly recovers** the global MLE under exponential families ($\hat{\theta}^{\mathrm{KL}} = \hat{\theta}^{mle}$)!

  – Because the local MLE $\hat{\theta}^k$ is a **sufficient statistic** of X^k

  – Can be viewed as a nonlinear averaging:

$$\hat{\theta}^{\mathrm{KL}} = g^{-1}\Big(\frac{g(\hat{\theta}^1) + \cdots + g(\hat{\theta}^d)}{d}\Big)$$

where $g: \theta \mapsto \mu \equiv \mathbb{E}_\theta[\phi(x)]$ (one-to-one map)

μ : The moment parameter

Non-Exponential Families

- Non-exponential families:
  - $\hat{\theta}^{\mathrm{KL}}$ doesn't equal $\hat{\theta}^{mle}$ in general
  - Error rate = how nearly "exponential family" they are

- *Statistical Curvature* (Efron 75): the mathematical measure of "exponential-family-ness".

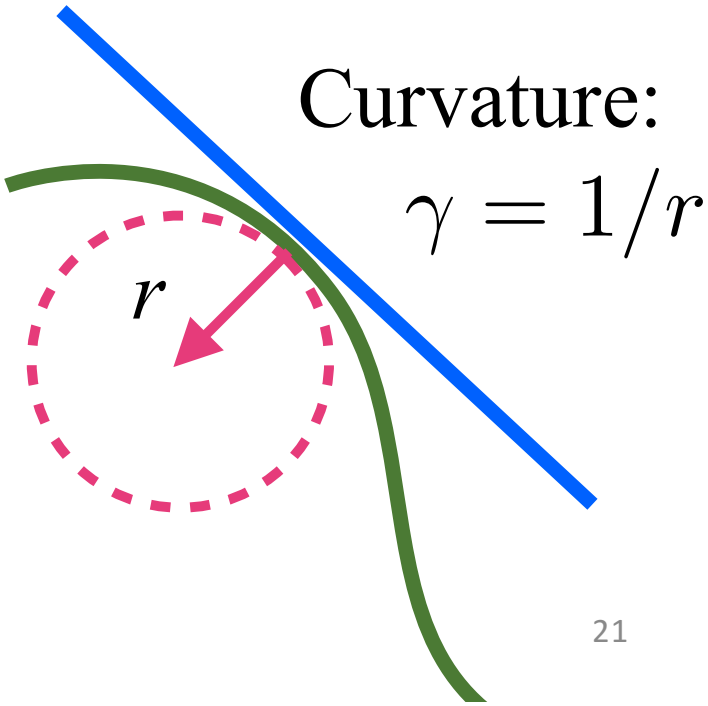
- Geometric intuition:

  Exponential families

  = lines / linear subspaces

  Non-exponential families

  = curves / curved surfaces



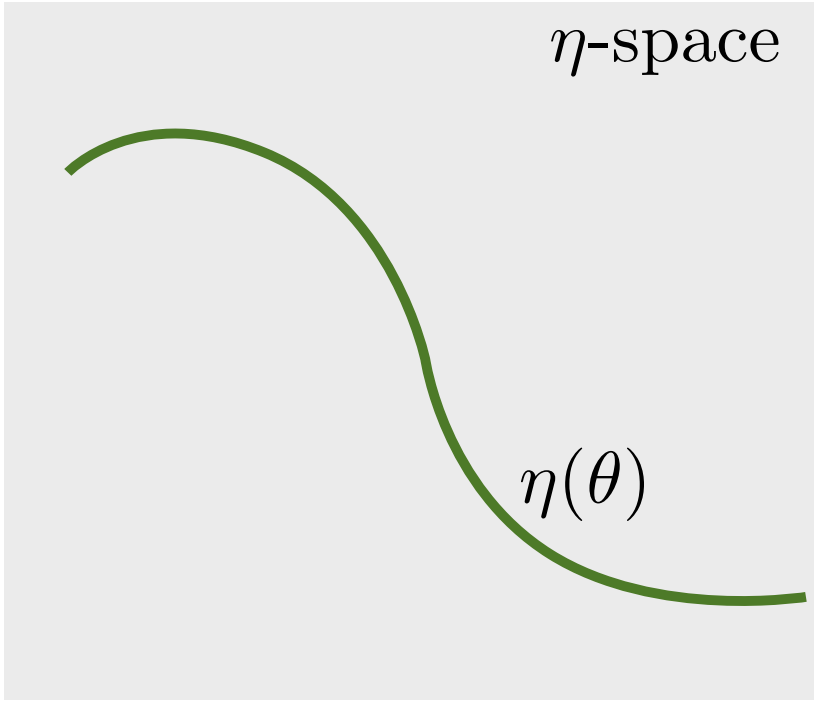
Curved Exponential Families

- Curved exponential families:

$$p(x|\theta) = \frac{1}{Z(\eta(\theta))} \exp\left(\eta(\theta)^T \phi(x)\right)$$

– Assume: dimension of θ is smaller than η (and $\phi(x)$)

– $\eta(\theta)$: lower dimensional curved surface in η -space



Curved Exponential Families

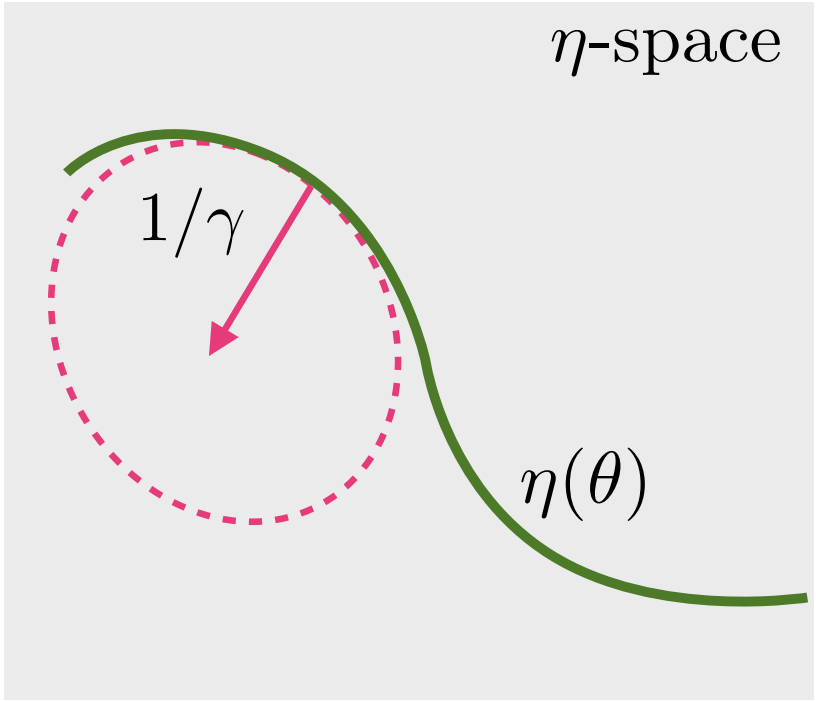
- Statistical curvature of $p(x|\theta)$: (Efron 75)
  - Geometric curvature of $\eta(\theta)$
  - with inner product via Fisher information matrix:

$$\langle \eta, \eta' \rangle \stackrel{def}{=} \eta^T \Sigma \eta'$$

where $\Sigma = \mathrm{cov}[\phi(x)]$

Curved exponential families:

$$p(x|\theta) = \frac{1}{Z(\eta(\theta))} \exp\left(\eta(\theta)^T \phi(x)\right)$$



Statistical Curvature &Information Loss

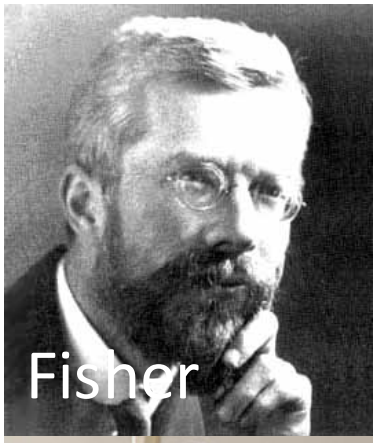
- Information loss of MLE (Fisher, Rao, Efron):
  - Statistical curvature = Loss of information when summarizing the data using MLE

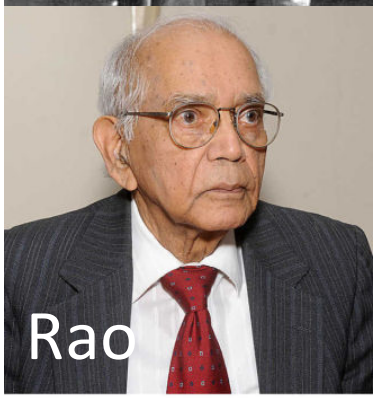
$$\gamma^2 = \lim_{n\to\infty} \mathcal{I}_1^{-1}(\mathcal{I}_n - \mathcal{I}_{\hat{\theta}^{mle}})$$

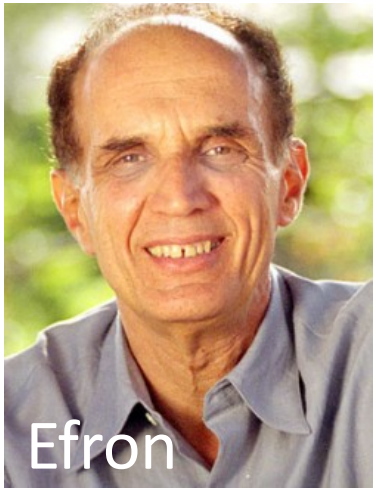
Statistical curvature

Fisher information in a simple of size n: $\mathcal{I}_n = n\mathcal{I}_1$

Fisher information in $\hat{\theta}^{mle}$ based on a sample of size n







  - **Full exponential family:**
    - => Zero curvature ($\gamma = 0$)
    - => No information loss
    - => MLE is sufficient statistics

Statistical Curvature &Information Loss

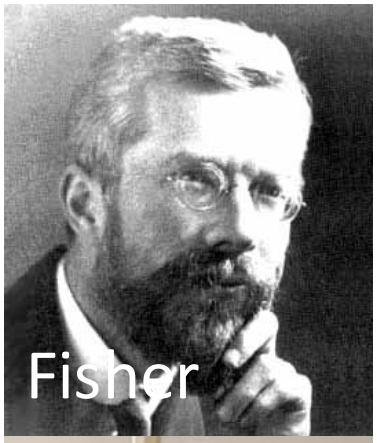
- Information loss of MLE (Fisher, Rao, Efron):
  - Statistical curvature = Loss of information when summarizing the data using MLE

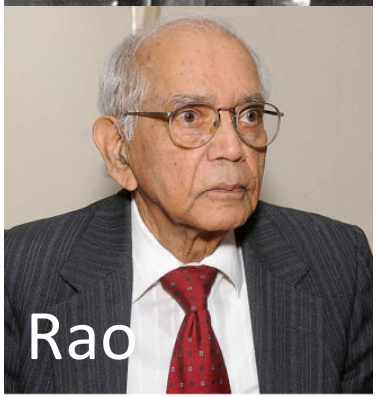
$$\gamma^2 = \lim_{n\to\infty} \mathcal{I}_1^{-1}(\mathcal{I}_n - \mathcal{I}_{\hat{\theta}^{mle}})$$

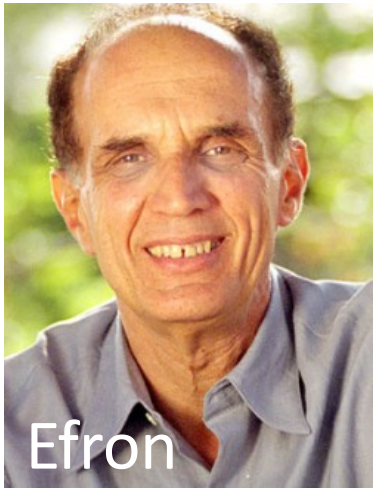
Statistical curvature

Fisher information in a simple of size n: $\mathcal{I}_n = n\mathcal{I}_1$

Fisher information in $\hat{\theta}^{mle}$ based on a sample of size n

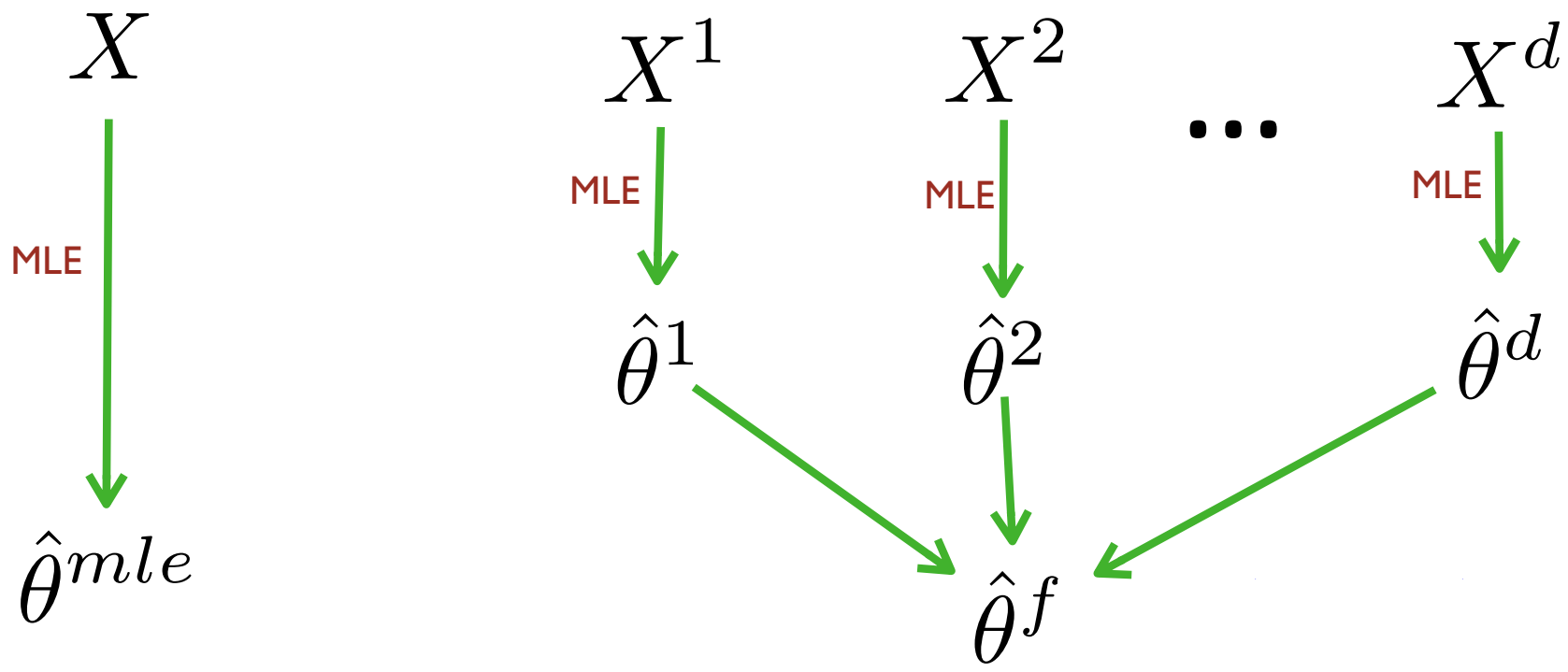






- Doesn't violate the **first order efficiency** of MLE: $\lim_{n\to\infty} \mathcal{I}_{\hat{\theta}^{mle}} / \mathcal{I}_n = 1$
- **Second order efficiency** of MLE: γ^2 is the lower bound

Information loss in Distributed Estimation



Information loss = γ^2

Information loss >= $d \cdot \gamma^2$

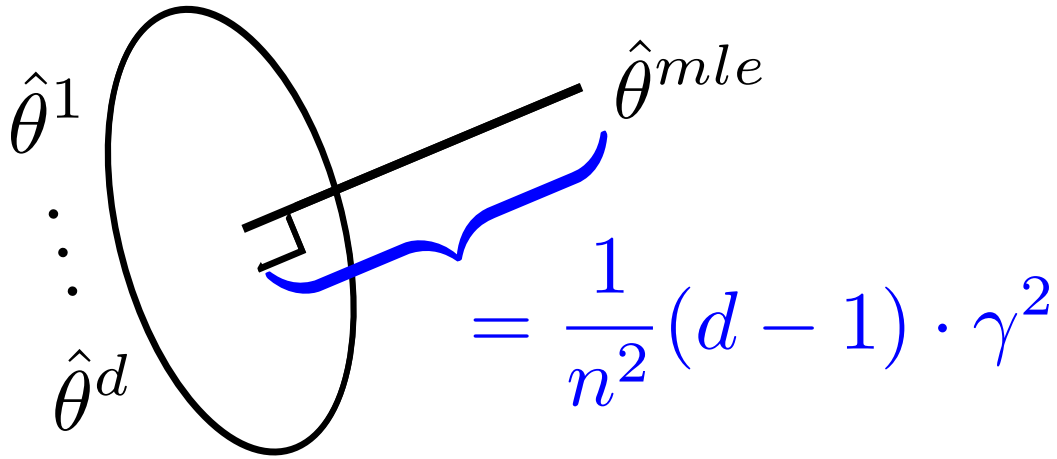
⇨ Relative Information loss >= $(d-1) \cdot \gamma^2$

Information Loss & Distributed Estimation

- **Lower bound**: For any $\hat{\theta}^f = f(\hat{\theta}^1, \ldots, \hat{\theta}^d)$, we have

$$\mathbb{E}||\sqrt{I}(\hat{\theta}^f - \hat{\theta}^{mle})||^2] \geq \frac{1}{n^2}(d-1) \cdot \gamma^2 + o(n^{-2})$$

  - Intuition: project $\hat{\theta}^{mle}$ into the space of spanned by $\hat{\theta}^1, \ldots, \hat{\theta}^d$



Information Loss & Distributed Estimation

- **Lower bound**: For any $\hat{\theta}^f = f(\hat{\theta}^1, \ldots, \hat{\theta}^d)$, we have

$$\mathbb{E}||\sqrt{I}(\hat{\theta}^f - \hat{\theta}^{mle})||^2] \geq \frac{1}{n^2}(d-1)\cdot\gamma^2 + o(n^{-2})$$

- **KL-averaging**: achieves the lower bound

$$\mathbb{E}||\sqrt{I}(\hat{\theta}^{\mathrm{KL}} - \hat{\theta}^{mle})||^2] = \frac{1}{n^2}(d-1)\cdot\gamma^2 + o(n^{-2})$$

- **Linear–averaging**: doesn't achieve the lower bound (even on exponential families)

$$\mathbb{E}||\sqrt{I}(\hat{\theta}^{\mathrm{linear}} - \hat{\theta}^{mle})||^2] = \frac{1}{n^2}(d-1)\cdot(\gamma^2 + (d+1)\beta^2) + o(n^{-2})$$

$$(\beta \neq 0 \text{ in general})$$

Error to the true parameters

- The error w.r.t. the *true parameters* is related to the error w.r.t. the global MLE

$$\mathbb{E}||\sqrt{I}(\hat{\theta}^{\mathrm{KL}} - \theta^*)||^2] \approx \mathrm{MSE}_{mle} + \frac{1}{n^2}(d-1)\cdot\gamma^2$$

$$\mathbb{E}||\sqrt{I}(\hat{\theta}^{\mathrm{linear}} - \theta^*)||^2] \approx \mathrm{MSE}_{mle} + \frac{1}{n^2}(d-1)\cdot(\gamma^2 + 2\beta^2)$$

- But note that

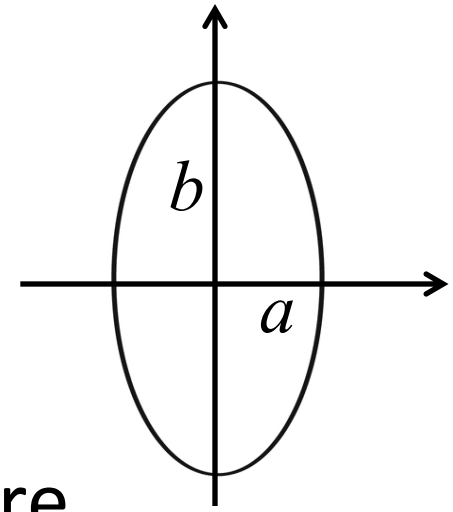
$$\mathrm{MSE}_{mle} \approx \frac{1}{n}\mathcal{I}_1^{-1} \gg \frac{1}{n^2}$$

  - The different is small, theoretically and asymptotically
  - But practically, KL-averaging is more robust in non-asymptotic or non-regular settings

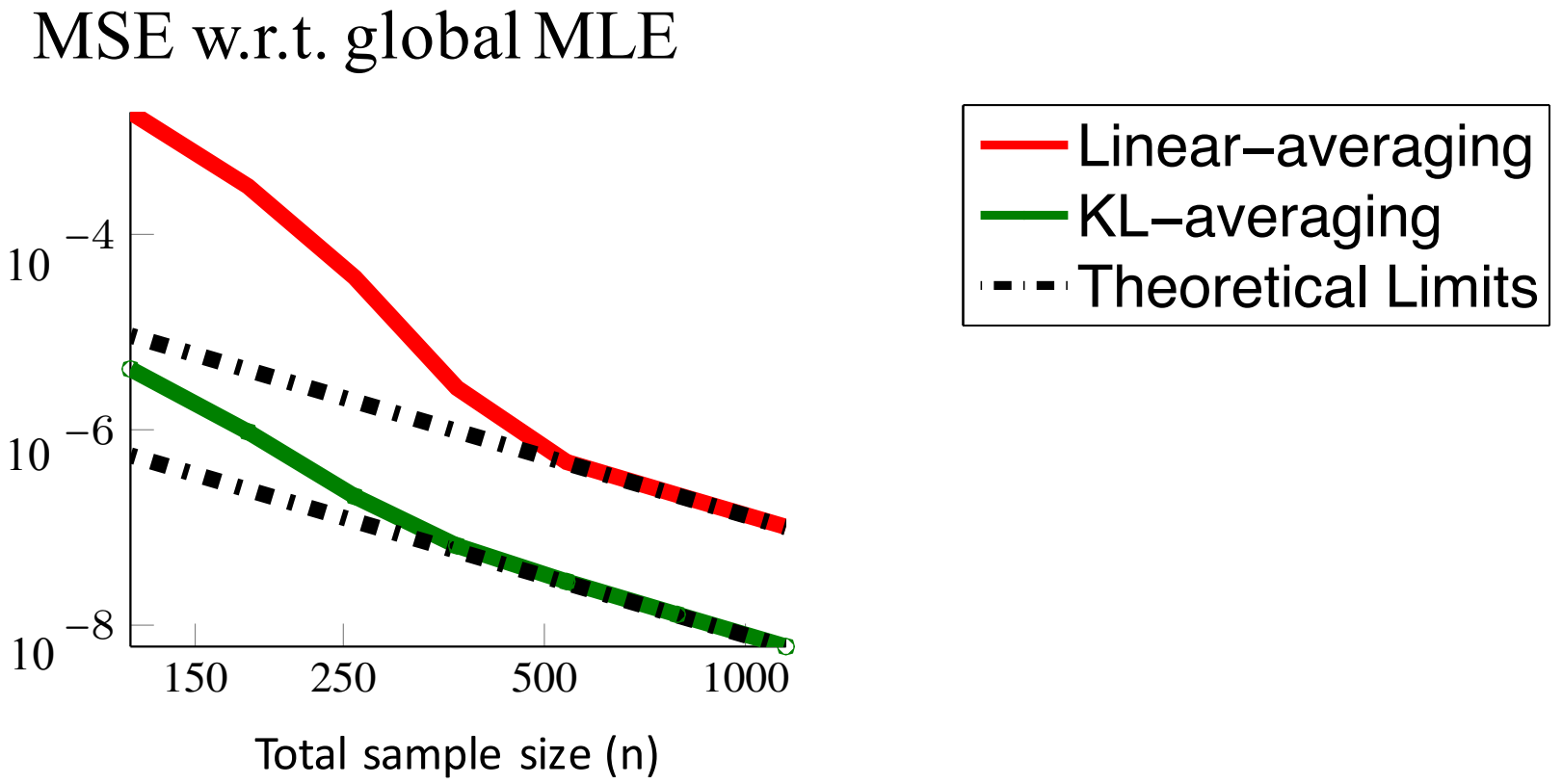
Experiments

- 2D Gaussian distribution on an ellipse

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} a\cos(\theta) \\ b\sin(\theta) \end{bmatrix}, \sigma^2 I \right)$$



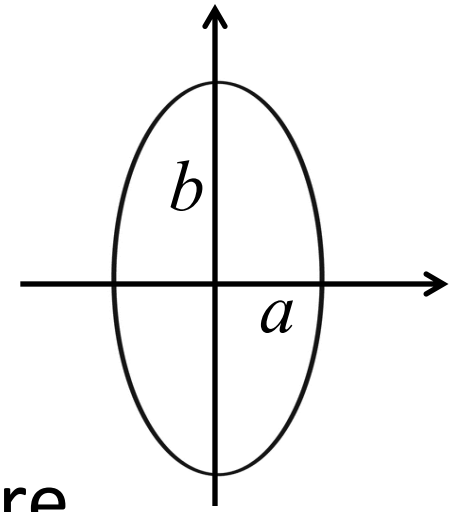
- Statistical curvature = geometric curvature



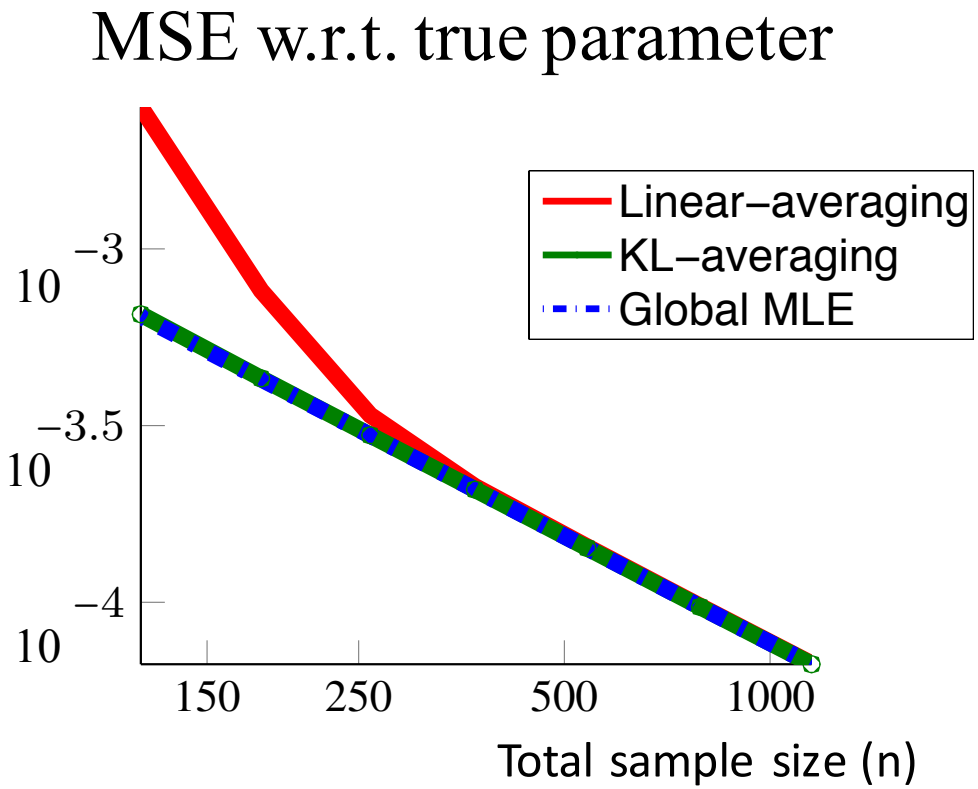
Experiments

- 2D Gaussian distribution on an ellipse

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} a\cos(\theta) \\ b\sin(\theta) \end{bmatrix}, \sigma^2 I \right)$$

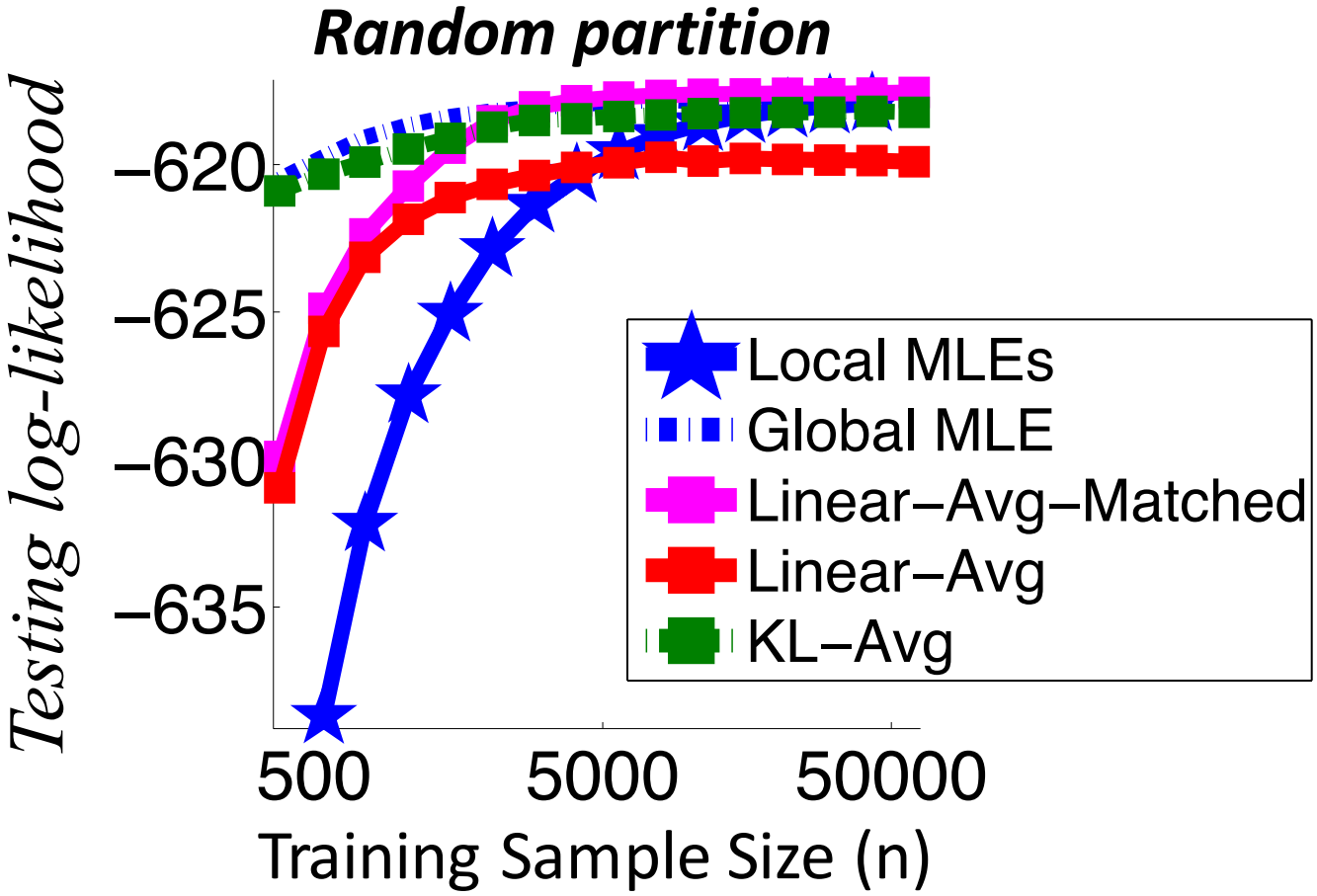


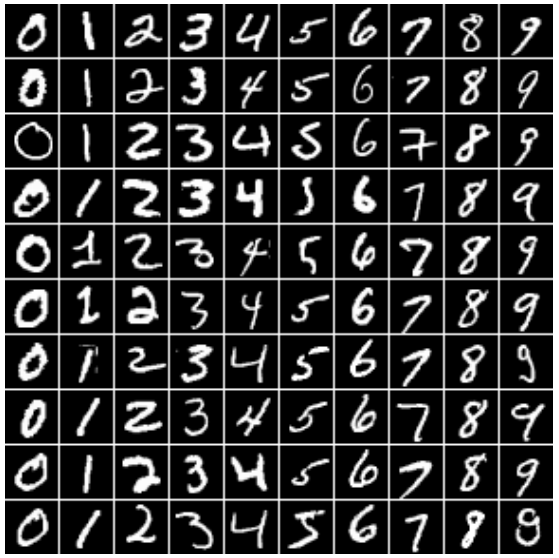
- Statistical curvature = geometric curvature



Gaussian mixture on MNIST data:

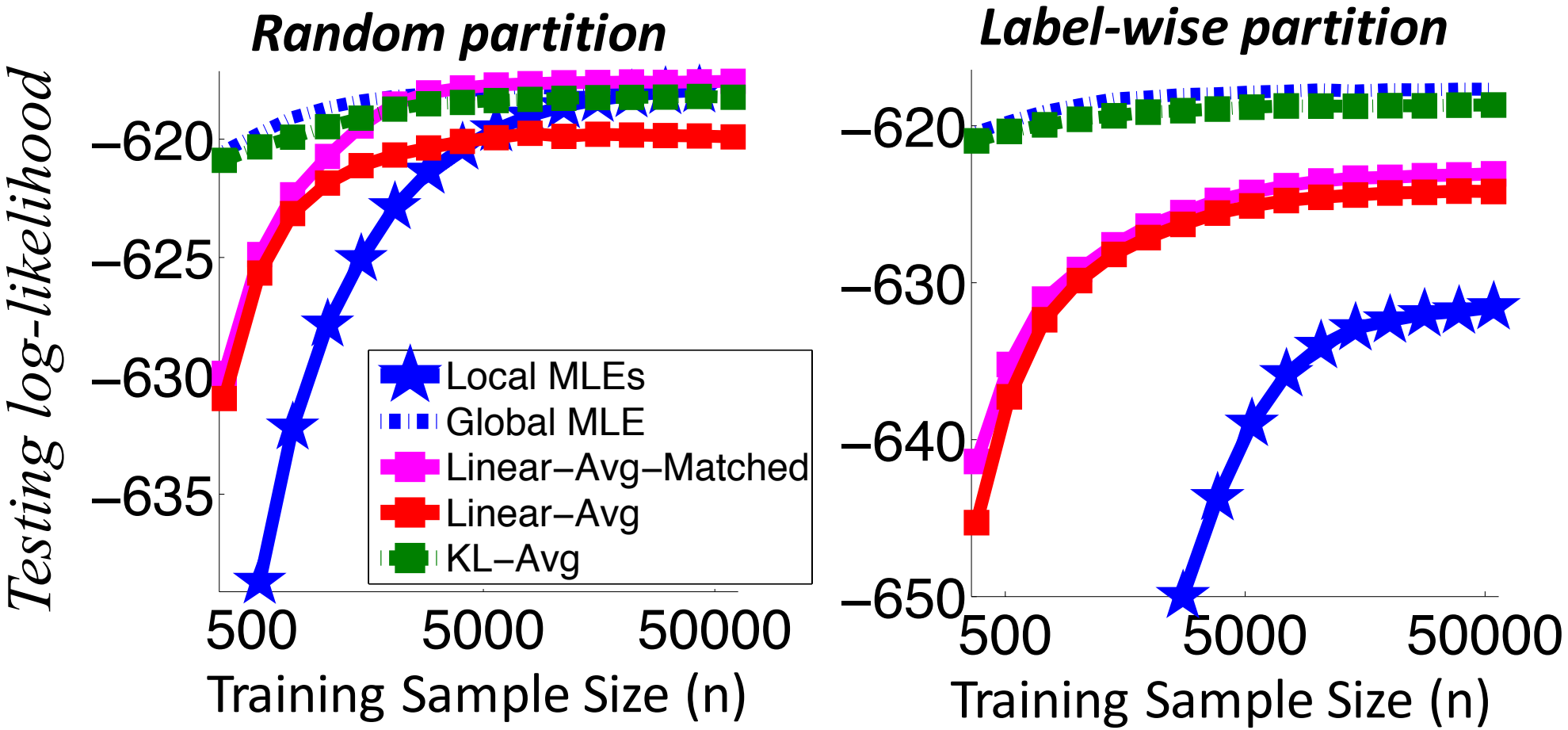
- Training data partitioned into 10 groups
- Calculate KL divergence by Monte Carlo (i.e., the parametric bootstrap procedure)
- Show testing likelihood for different methods





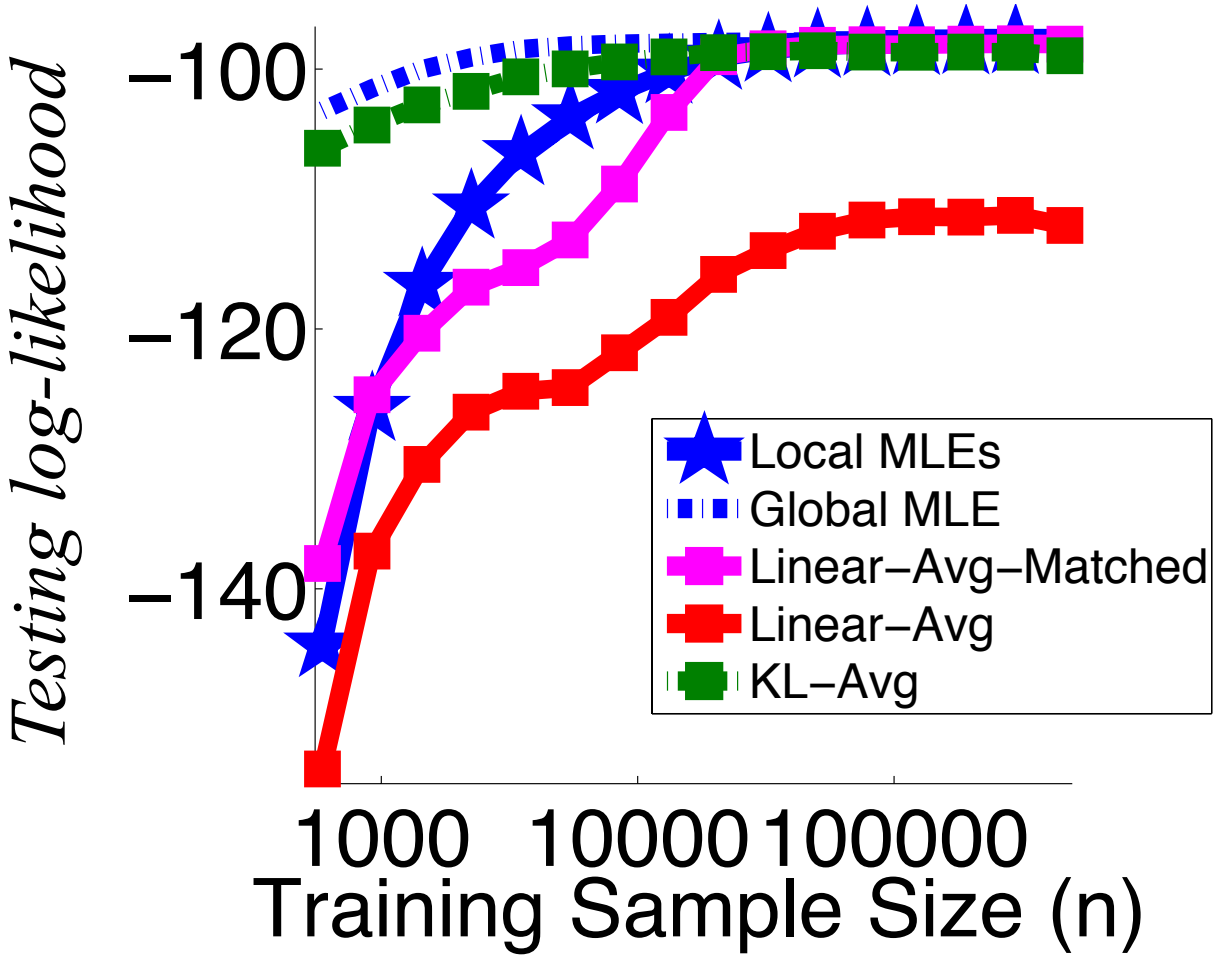
Gaussian mixture on MNIST data:

- Training data partitioned into 10 groups
- Calculate KL divergence by Monte Carlo (i.e., the parametric bootstrap procedure)
- Show testing likelihood for different methods



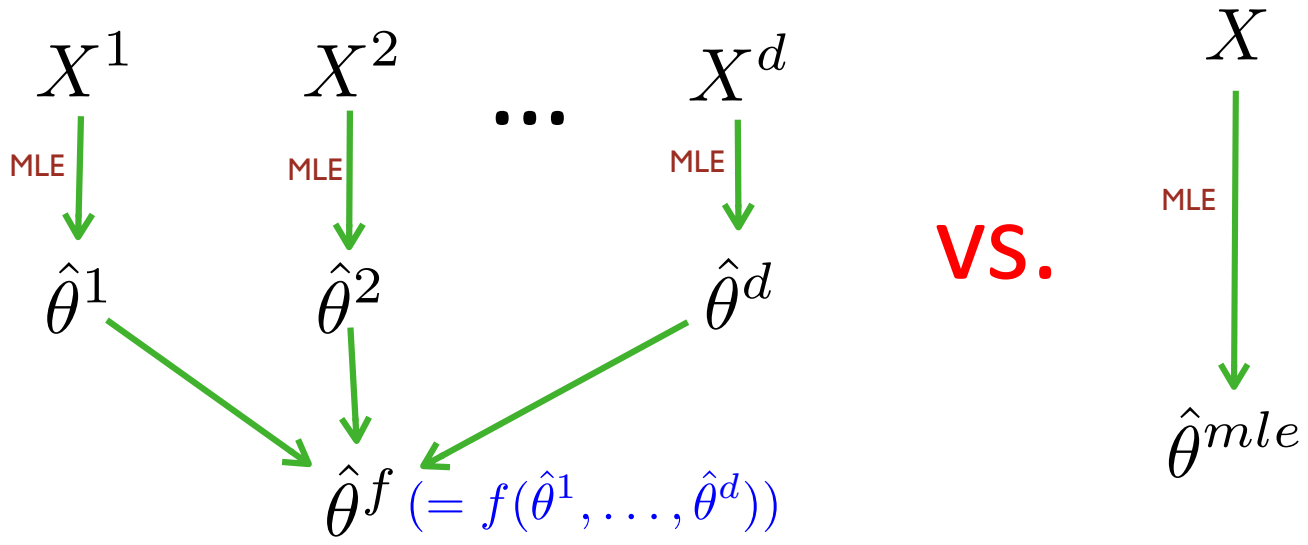
More datasets

- YearPredictionMSD dataset from UCI repository
- Similar setting as before (random partition)



Conclusion

- Distributed Learning:



- Combination function f()?
  - KL-averaging vs. linear-averaging
- Statistical loss?
  - Exponential / non-exponential families
  - statistical curvature = Information loss

Thanks!