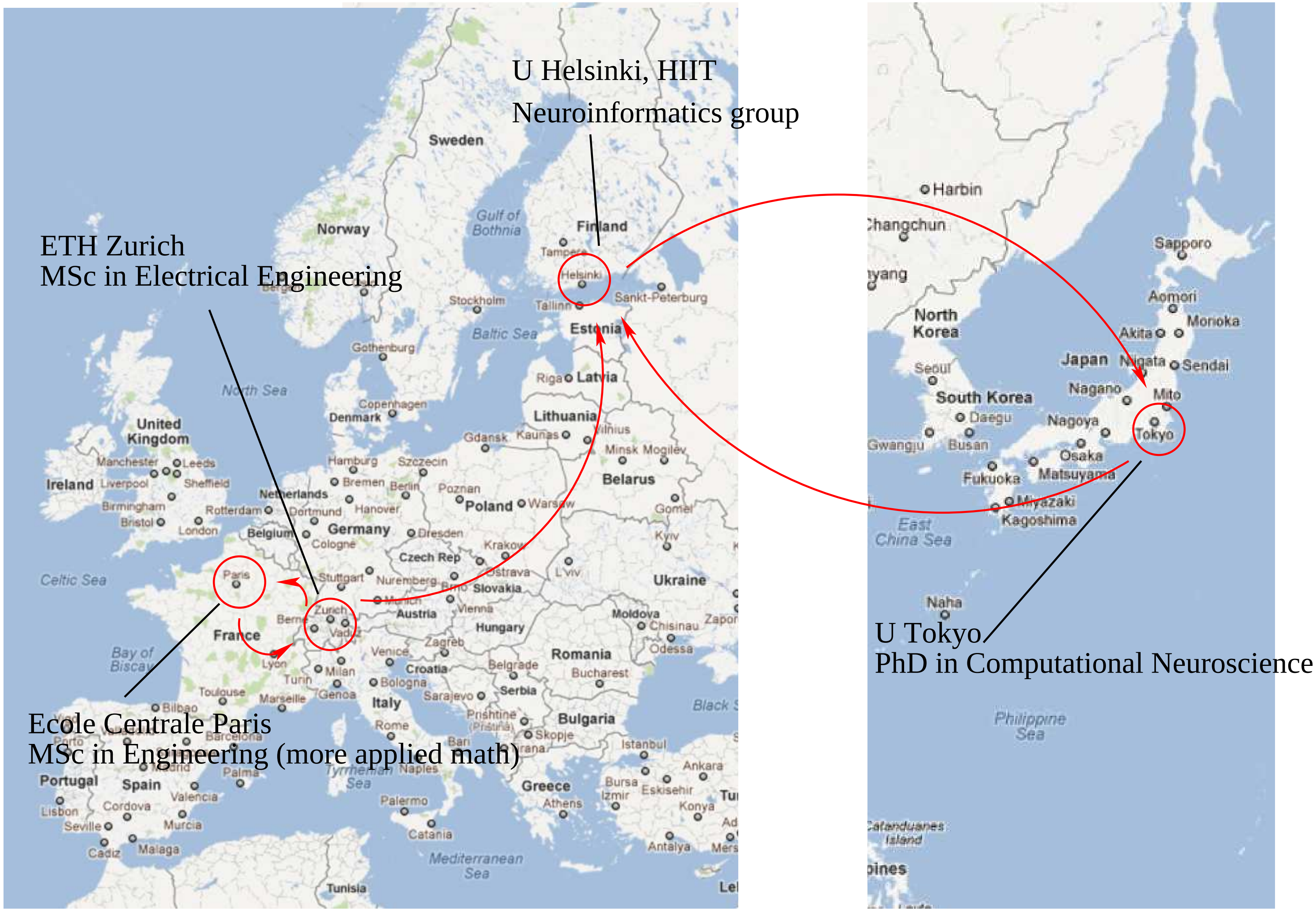
On the Estimation of Unnormalized Statistical Models

Michael U. Gutmann
Dept of Computer Science and HIIT
Dept of Mathematics and Statistics
University of Helsinki
michael.gutmann@helsinki.fi

My academic background on the map



Program

- Introduction: Background on unnormalized models, why they are hard to estimate, but why being able to estimate them is important
- Core part:
  - Noise-contrastive estimation: A new estimation method for unnormalized models (Gutmann and Hyvärinen, JMLR, 13(Feb):307–361, 2012.
  - Application in the modeling of natural images
- Closing: Some open questions and a summary

Introduction

The talk is about parametric estimation

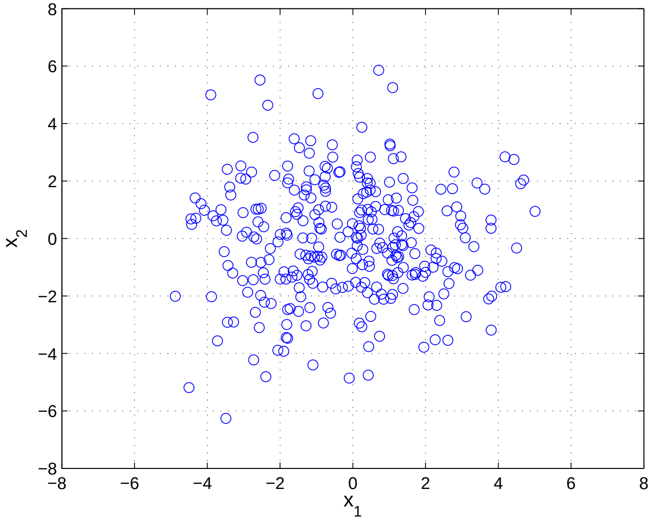
The big picture of parametric estimation is as follows:

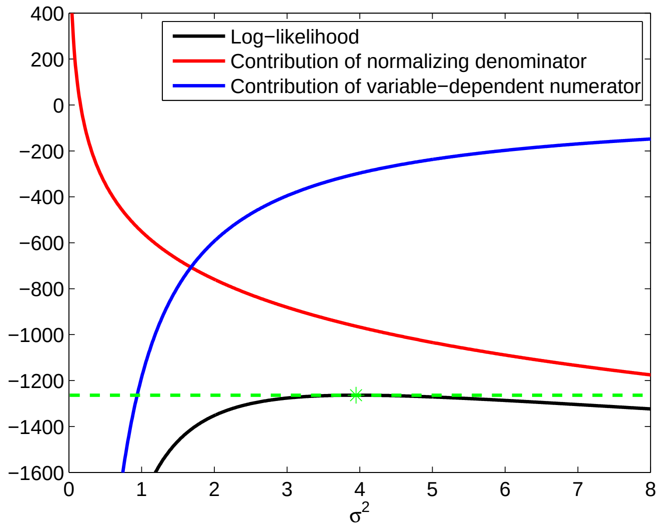
Observe a collection $X = (\mathbf{x}_1, \ldots, \mathbf{x}_{T_d})$ of continuous or discrete random variables $\mathbf{x}_t$.

Assume that the $\mathbf{x}_t$ are iid and that their distribution $p_d(\mathbf{x})$ belongs to the family of nonnegative functions $\{p_m(\mathbf{x}; \boldsymbol{\theta})\}_{\boldsymbol{\theta}}$ parameterized by $\boldsymbol{\theta} \in \mathbb{R}^m$. That is $p_d(\mathbf{x}) = p_m(\mathbf{x}; \boldsymbol{\theta}^\star)$.

Find $\boldsymbol{\theta}^\star$

Example 1





- Model of $\mathbf{x} \in \mathbb{R}^2$: $p_m(\mathbf{x}, \sigma^2) = \dfrac{\exp\left(-\frac{||\mathbf{x}||^2}{2\sigma^2}\right)}{2\pi\sigma^2}$
- Estimation of σ^2 by maximizing the log-likelihood ℓ

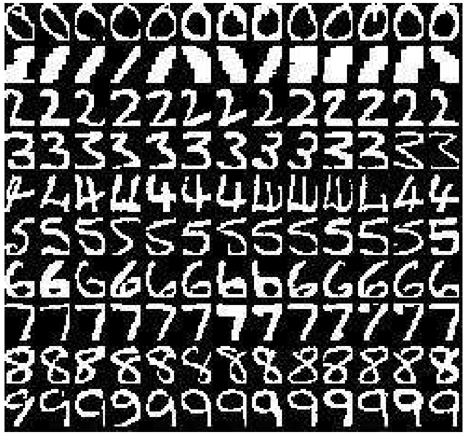
$$\ell(\sigma^2) = -T_d \ln(2\pi\sigma^2) + \frac{1}{\sigma^2}\sum_{t=1}^{T_d} \frac{-||\mathbf{x}_t||^2}{2}$$

- The term $2\pi\sigma^2$, which is such that $\int p_m(\mathbf{x};\sigma^2)\mathrm{d}\mathbf{x} = 1 \ \forall \sigma^2$, is important in maximum likelihood estimation (MLE).

Example 2

- Model of $\mathbf{x} \in \{-1, 1\}^{320}$: $p_m(\mathbf{x}; \boldsymbol{\alpha}) = \frac{p_m^0(\mathbf{x};\boldsymbol{\alpha})}{Z(\boldsymbol{\alpha})}$

  $p_m^0(\mathbf{x}; \boldsymbol{\alpha})$ is some complicated function which captures the shape of the data distribution very well.



- The normalizing partition function $Z(\boldsymbol{\alpha})$ is

$$Z(\boldsymbol{\alpha}) = \sum_{\mathbf{x} \in \{-1,1\}^{320}} p_m^0(\mathbf{x}; \boldsymbol{\alpha})$$

  The sum goes over $2^{320} \approx 10^{96}$ configurations.

- Estimation of $\boldsymbol{\alpha}$ by maximizing the log-likelihood ℓ,

$$\ell(\boldsymbol{\alpha}) = -T_d \ln Z(\boldsymbol{\alpha}) + \sum_{t=1}^{T_d} p_m^0(\mathbf{x}_t; \boldsymbol{\alpha}),$$

  is computationally *very* expensive (curse of dimensionality).

Important points so far

What I wanted to illustrate with the examples is:

- The normalizing term plays a key role in MLE.
- If we use MLE to estimate $\boldsymbol{\theta}$, $p_m(\mathbf{x}; \boldsymbol{\theta})$ must integrate to one for all $\boldsymbol{\theta}$. This imposes a condition on the model family: the model must be normalized.
- For MLE, having the “perfect” model for the *shape* of the data distribution does not yield much if we do not know the proper *scaling* of the model.
- But scaling (normalizing) a model may be analytically impossible or computationally expensive.

Focus of the talk

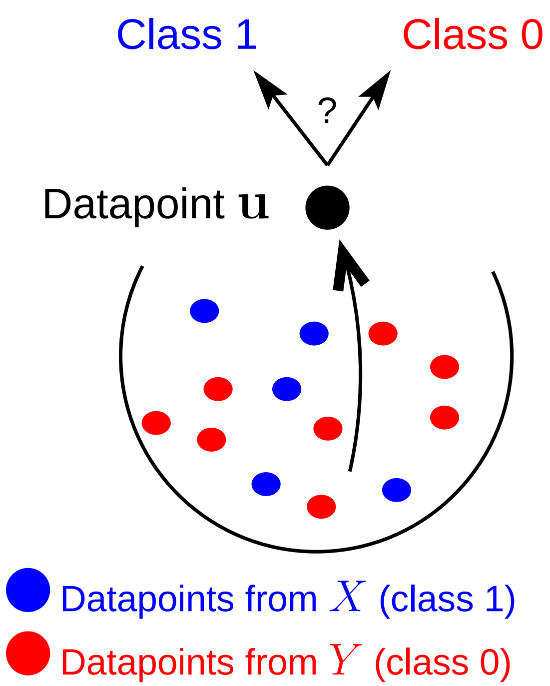
Focus of the talk: estimating $\boldsymbol{\theta}$ without requiring that the model $p_m(\mathbf{x};\boldsymbol{\theta})$ integrates to one for all possible values of the parameter $\boldsymbol{\theta}$

- Such models are said to be unnormalized. MLE is not applicable.
- Examples of unnormalized models:
  - Unnormalized Gaussian (a pairwise Markov network) :
    $\ln p_m(\mathbf{x};\boldsymbol{\theta}) = -1/2\mathbf{x}^T\boldsymbol{\Lambda}\mathbf{x} + \mathbf{b}^T\mathbf{x} + c$
    $\boldsymbol{\theta}$: upper triangular part of $\boldsymbol{\Lambda}$, $\mathbf{b}$, c
  - More general:
    $\ln p_m(\mathbf{x};\boldsymbol{\theta}) = \ln p_m^0(\mathbf{x};\boldsymbol{\alpha}) + c$
    $\boldsymbol{\theta}$: $\boldsymbol{\alpha}$, c
- Parameter $\boldsymbol{\alpha}$ is responsible for the shape of p_m, parameter c for the scaling of p_m. It is a normalizing parameter and takes the role of $\ln 1/Z$.

Noise-Contrastive Estimation

Logistic regression for classification

- Denote by $Y = \{\mathbf{y}_1, \dots \mathbf{y}_{T_n}\}$ a data set of iid observations of a random variable $\mathbf{y}$ with distribution p_n.
- Logistic regression can be used to discriminate between the two data sets X and Y.



- Let the regression function be

$$P(C = 1|\mathbf{u}; \boldsymbol{\theta}) = \frac{1}{1 + F(\mathbf{u}; \boldsymbol{\theta})}$$

with $F(\mathbf{u}; \boldsymbol{\theta}) \geq 0$

- Conditional log-likelihood $J_T(\boldsymbol{\theta})$

$$\sum_{t=1}^{T_d} \ln P(C = 1|\mathbf{x}_t; \boldsymbol{\theta}) + \sum_{t=1}^{T_n} \ln \left[P(C = 0|\mathbf{y}_t; \boldsymbol{\theta})\right]$$

can be used to learn $\boldsymbol{\theta}$.

- Classification rule: Class $C = 1$ if $P(C = 1|\mathbf{u}; \boldsymbol{\theta}) > 1/2$

Doing more with logistic regression

- Using Bayes' theorem we have that

$$P(C=1|\mathbf{u}) = \left(1 + \frac{T_n p_n(\mathbf{u})}{T_d p_d(\mathbf{u})}\right)^{-1}$$

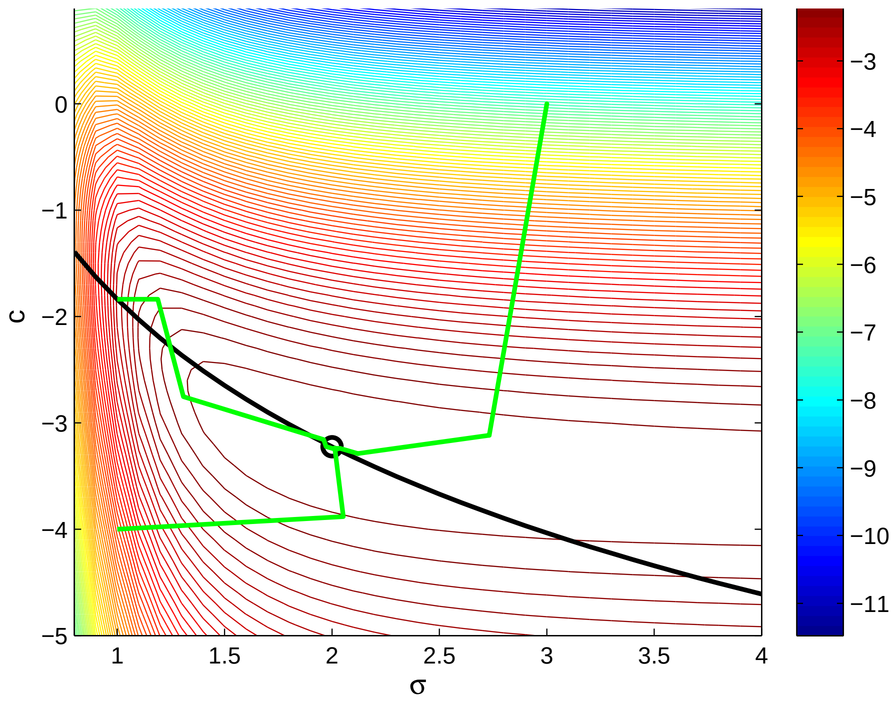
- We can show that $\hat{\boldsymbol{\theta}}$ satisfying $F(\mathbf{u}; \hat{\boldsymbol{\theta}}) = \frac{T_n p_n(\mathbf{u})}{T_d p_d(\mathbf{u})}$ is maximizing the conditional log-likelihood.
- Hence, if we choose ourselves p_n, create Y, and write $F(\mathbf{u}; \boldsymbol{\theta})$ as

$$F(\mathbf{u}; \boldsymbol{\theta}) = \frac{T_n p_n(\mathbf{u})}{T_d p_m(\mathbf{u}; \boldsymbol{\theta})}$$

  we can estimate the model $p_m(\mathbf{x}; \boldsymbol{\theta})$ via logistic regression.
- We call this procedure to estimate $\boldsymbol{\theta}$ "noise-contrastive estimation". (Gutmann and Hyvärinen, JMLR, 13(Feb):307–361, 2012.)
- The next slide shows that in noise-contrastive estimation p_m does not need to be normalized.

Simple Example

- Observed data X: Zero mean Gaussian with standard deviation $\sigma = 2$; Contrastive noise Y: standard Gaussian
- Unnormalized model: $\ln p_m(\mathbf{x}; \sigma, c) = -\frac{||\mathbf{x}||^2}{2\sigma^2} + c$
- Contour plot of $J_T(\sigma, c)$ (to be maximized)
  black: $c^\star = \ln 1/Z(\sigma)$ (location of properly normalized models),
  green: optimization trajectories



Statistical properties of noise-contrastive estimation

- Denote by $\hat{\boldsymbol{\theta}}_T$ the parameter vector which maximizes $J_T(\boldsymbol{\theta})$, the objective where T_d observations of $\mathbf{x} \sim p_m(\mathbf{x}; \boldsymbol{\theta}^\star)$ are used.
- Property 1 (consistency): As T_d increases $\hat{\boldsymbol{\theta}}_T$ converges in probability to $\boldsymbol{\theta}^\star$.

  For proof and (mild) conditions, see Gutmann and Hyvärinen, JMLR, 13(Feb):307–361, 2012.
- Property 2: For normalized models, as $\nu = T_n/T_d$ increases, for any valid choice of p_n, noise-contrastive estimation tends to “perform as well” as MLE (more formally: it is asymptotically Fisher efficient).
- We have also studied other properties like the distribution of $\hat{\boldsymbol{\theta}}_T$ when T_d is large, see the article above.

Validating the properties with toy data (1/2)

- Let the data follow the ICA model $\mathbf{x} = \mathbf{A}\mathbf{s}$ with 4 sources.
- The distribution of $\mathbf{x}$ is

$$\ln p_m(\mathbf{x}; \boldsymbol{\theta}^\star) = -\sum_{i=1}^{4} \sqrt{2}|\mathbf{b}_i^\star \mathbf{x}| + c^\star$$

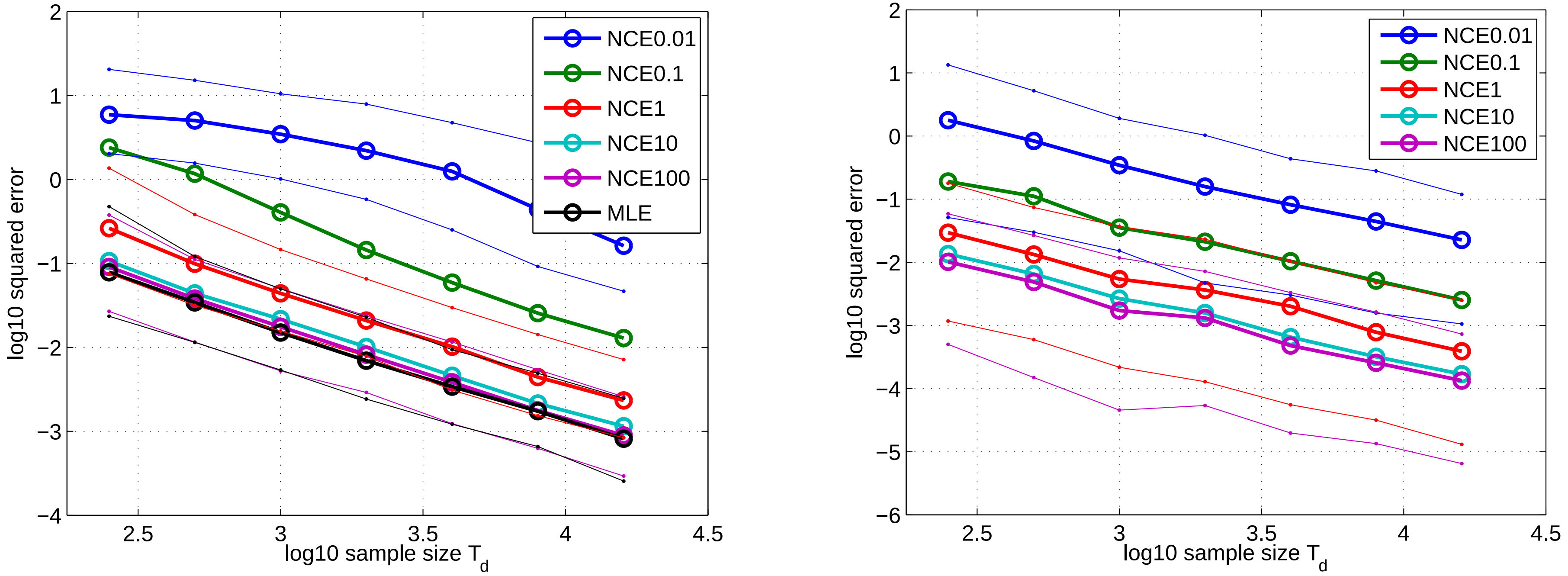
  with $c^\star = \ln |\text{det}\mathbf{B}^\star| - \frac{4}{2}\ln 2$ and $\mathbf{B}^\star = \mathbf{A}^{-1}$.
- For this toy data, we could formulate a properly normalized model. To validate our method, let us estimate the unnormalized model

$$\ln p_m(\mathbf{x}; \boldsymbol{\theta}) = -\sum_{i=1}^{4} \sqrt{2}|\mathbf{b}_i \mathbf{x}| + c$$

  with parameters $\boldsymbol{\theta} = (\mathbf{b}_1, \ldots, \mathbf{b}_4, c)$.
- Contrastive noise p_n: Gaussian with the same covariance as the data.

Validating the properties with toy data (2/2)

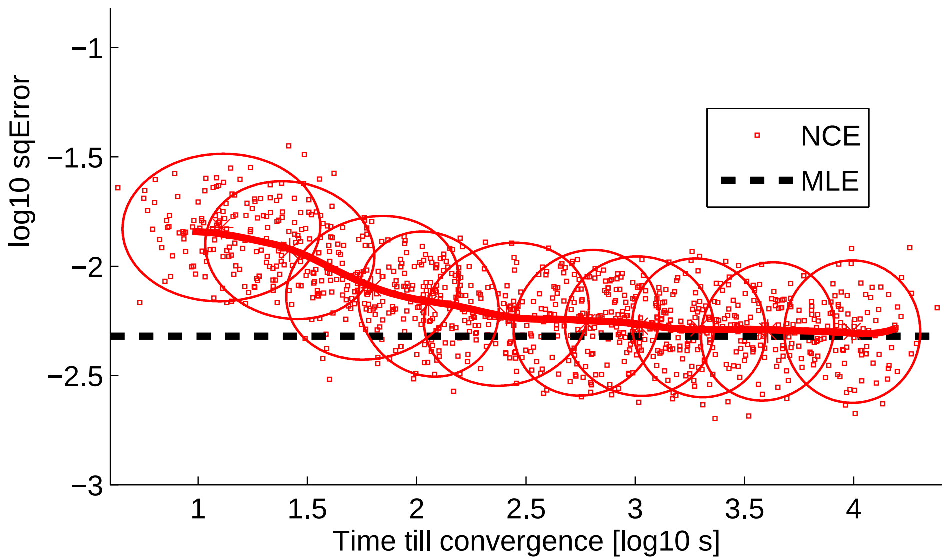
Results for 500 estimation problems with random $\mathbf{A}$, for $\nu \in \{0.01, 0.1, 1, 10, 100\}$. For the MLE results, we used the properly normalized model.



(a) Mixing matrix (b) Normalizing parameter

Computational aspects (1/3)

- The estimation accuracy improves as the number of noise samples T_n increases.
- With more noise samples, more computations are needed.
  $\rightarrow$ There is a trade-off between computational and statistical performance.
- Example: ICA model as before but with 10 sources.
  $T_d = 8000$, $\nu \in \{1, 2, 5, 10, 20, 50, 100, 200, 400, 1000\}$.
  Performance for 100 random estimation problems:



Computational aspects (2/3)

How good is the trade-off? Let's compare with other estimation methods.

1. MLE where partition function is evaluated with importance sampling. Maximization of

$$J_{\mathsf{IS}}(\boldsymbol{\alpha}) = \frac{1}{T_d}\sum_{t=1}^{T_d} \ln p_m^0(\mathbf{x}_t;\boldsymbol{\alpha}) - \ln\left(\frac{1}{T_n}\sum_{t=1}^{T_n}\frac{p_m^0(\mathbf{n}_t;\boldsymbol{\alpha})}{p_{\mathsf{IS}}(\mathbf{n}_t)}\right)$$

   $p_{\mathsf{IS}} = p_n$ is the proposal distribution and
   $\ln p_m^0(\mathbf{x};\boldsymbol{\alpha}) = -\sum_{i=1}^{10}\sqrt{2}|\mathbf{b}_i\mathbf{x}|$, $\boldsymbol{\alpha} = (\mathbf{b}_1,\ldots,\mathbf{b}_{10})$

2. Score matching: minimization of

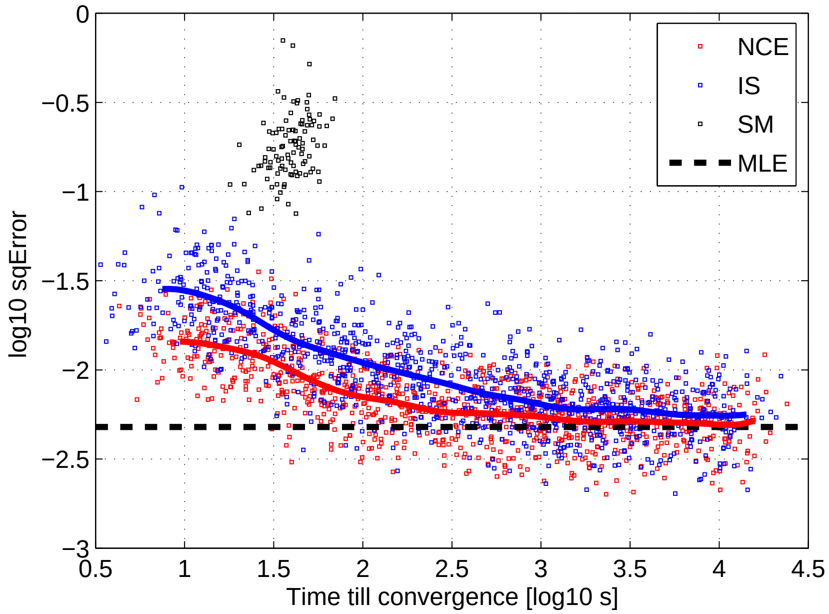
$$J_{\mathsf{SM}}(\boldsymbol{\alpha}) = \frac{1}{T_d}\sum_{t=1}^{T_d}\sum_{i=1}^{10}\frac{1}{2}\Psi_i^2(\mathbf{x}_t;\boldsymbol{\alpha}) + \Psi_i'(\mathbf{x}_t;\boldsymbol{\alpha})$$

   with $\Psi_i(\mathbf{x};\boldsymbol{\alpha}) = \frac{\partial \ln p_m^0(\mathbf{x};\boldsymbol{\alpha})}{\partial \mathbf{x}(i)}$ (smoothing needed!)

(see JMLR2012 paper for more comparisons)

Computational aspects (3/3)

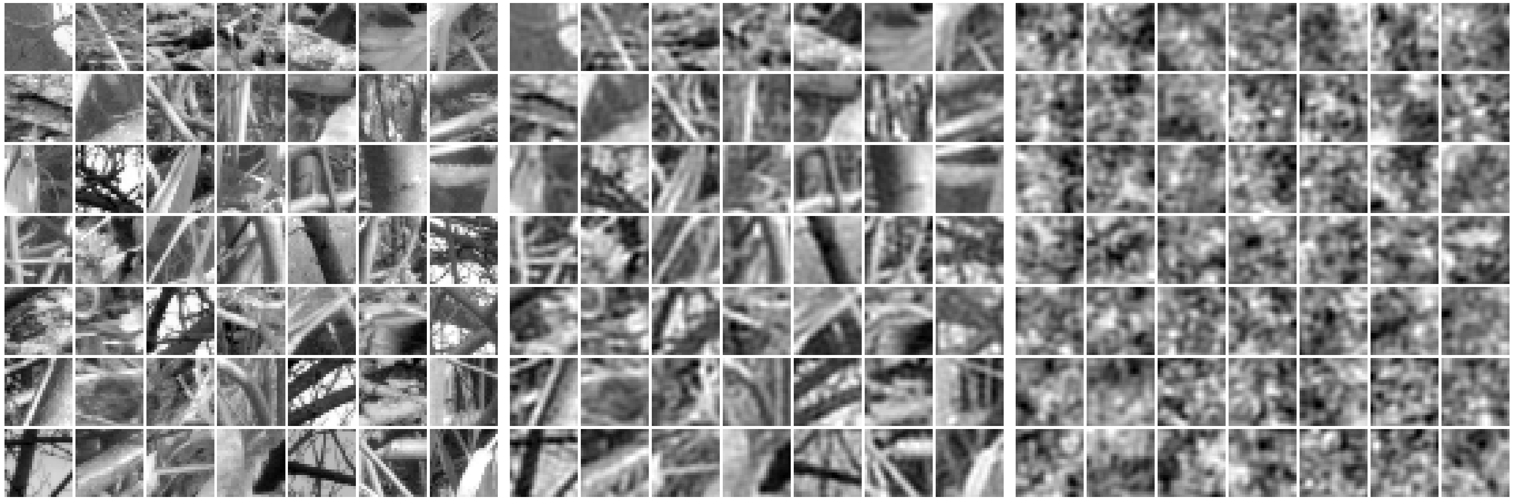
- Compared to the importance sampling approach (IS), noise-contrastive estimation (NCE) is less sensitive to the mismatch of data and noise distribution.
- Score matching (SM) does not perform well if the data distribution is not smooth.
- NCE seems suitable for data with heavy tails or non-smooth distribution.



Application in the Modeling of Natural Images

Natural image data

- Image patches: 32×32 pixel subregions of larger images
- Preprocessing: PCA dimension reduction from 625 to 160 (93% of variance retained), cancelling illumination condition by centering each patch.
- Data is clearly structured. Its modeling is important for image processing (e.g. denoising) and for understanding the visual processing in the brain.



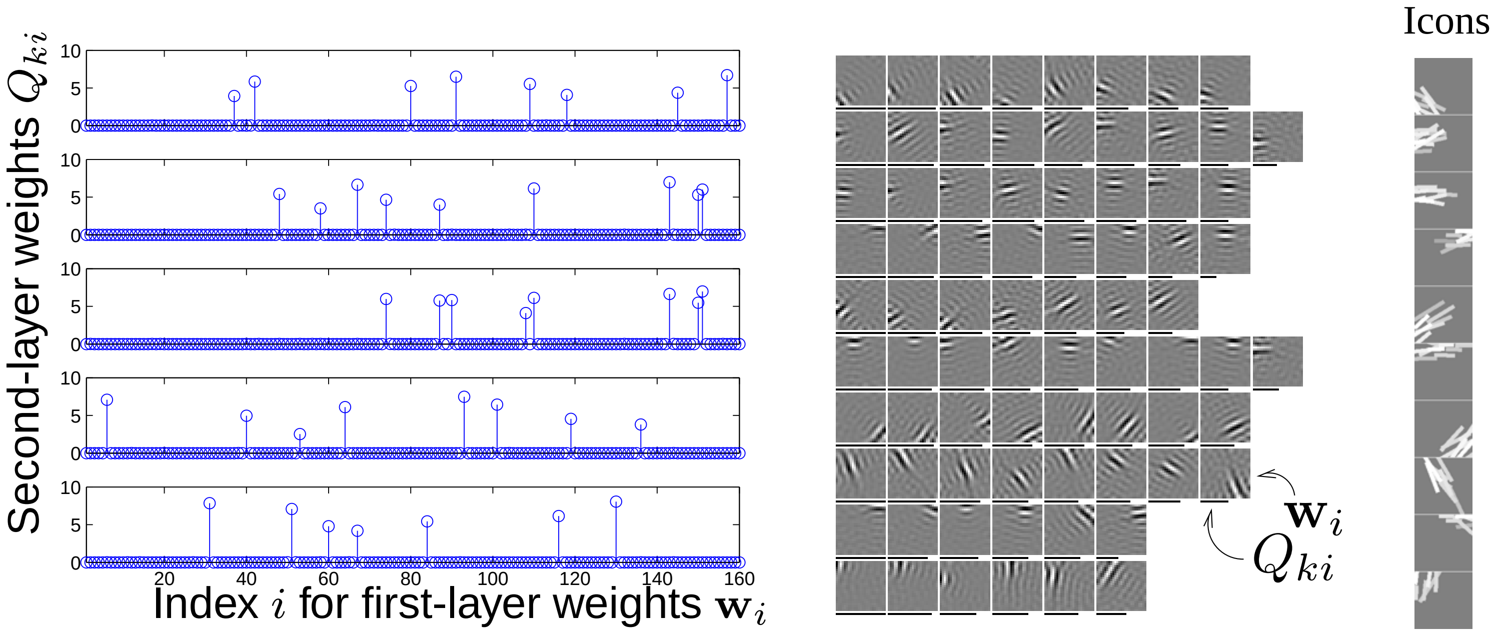
(a) Image patches (b) After preprocessing (c) Noise

Two-layer model

- Build a multi-layer network which takes as input an image $\mathbf{x}$ and outputs the pdf at $\mathbf{x}$.
  - First layer: compute feature outputs $\mathbf{w}_i^T\mathbf{x}$ for $i = 1, \dots, 160$
  - Then: compute "energies" $(\mathbf{w}_i^T\mathbf{x})^2$
  - Second layer: pooling of energies: $y_k = \sum_i Q_{ki}(\mathbf{w}_i^T\mathbf{x})^2$, $Q_{ki} > 0$, $k = 1, \dots, 160$
  - Output of the network: $\ln p_m(\mathbf{x}; \boldsymbol{\theta}) = \sum_{k=1}^{160} f(y_k) + c$ where f is a spline nonlinearity
- The parameters are $\boldsymbol{\theta} = (\mathbf{w}_i, Q_{ki}, f, c)$. There are more than 50'000.
- The model p_m is unnormalized. We estimate it with noise-contrastive estimation.

Results of the estimation: features

- The $\mathbf{w}_i$ are “Gabor-like”.
- The second layer is more interesting: Five different summations $\sum_{i=1}^{n} Q_{ki}(\mathbf{w}_i^T\mathbf{x})^2$ are shown.

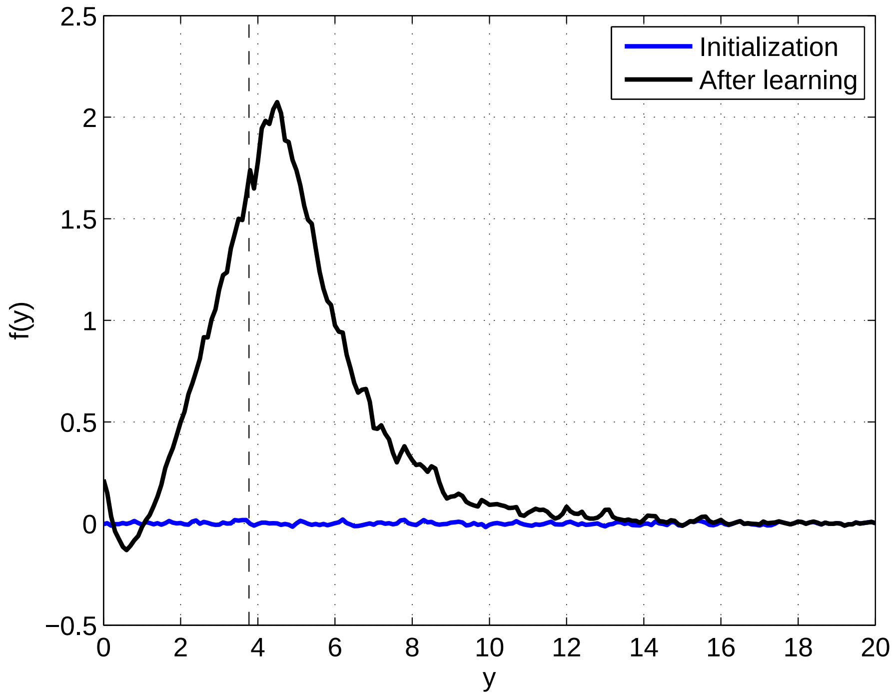


(a) Raw result

(b) Graphical visualization

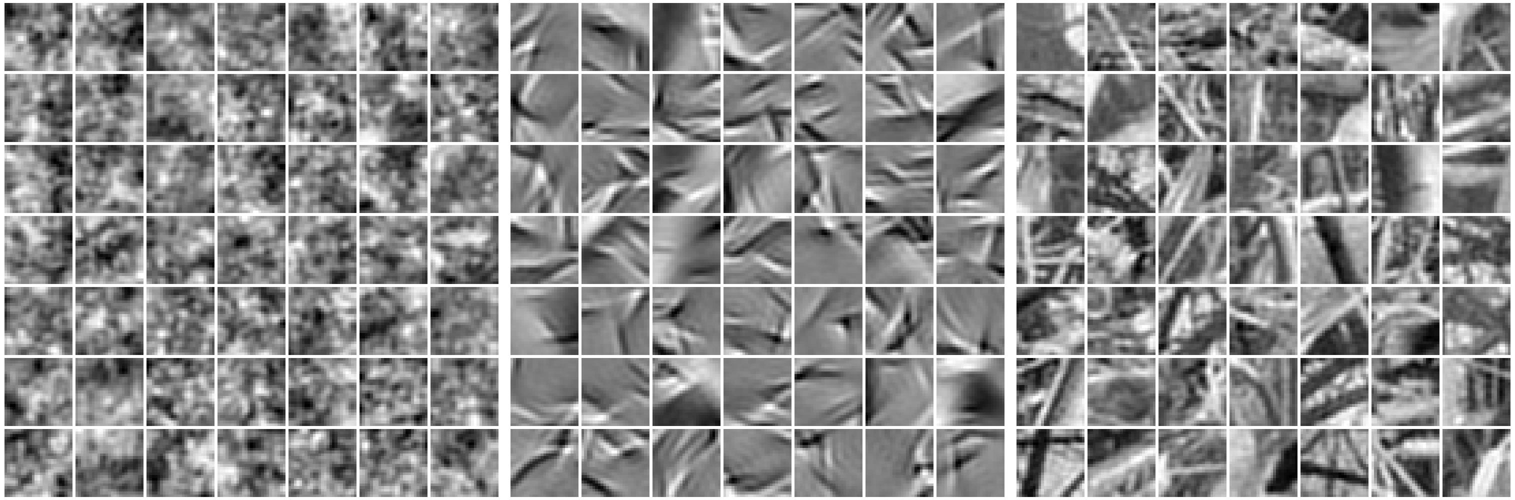
Results of the estimation: nonlinearity

- The nonlinearity f at the beginning and end of the learning.
- For natural images 99% of the second layer outputs y_k fall to the left of the dashed line. The learned f is only valid in that region.
- Nonlinearity f assigns high probabilities to either very small or large y_k. $\rightarrow$ sparsity of the feature outputs



Results of the estimation: likely points of the model

- We visualize here the behavior of the model by showing what kind of structure the model considers likely.
- Initialize $\mathbf{x}$ randomly and find $\hat{\mathbf{x}} = \text{argmax}_{\mathbf{u}} p_m(\mathbf{u}; \hat{\boldsymbol{\theta}})$. We call the resulting local maximum a “likely point”.



(a) Noise (b) Likely points (c) Training data

Closing

Some research directions for the estimation part

- Noise-contrastive estimation uses noise (auxiliary) samples. How to select their distribution p_n?
- Noise-contrastive estimation (NCE) is a special instance of a large family of estimators (Gutmann and Hirayama, UAI 2011). Minimizing

$$
\begin{aligned}
L_\Psi(\boldsymbol{\theta}) \quad = \quad & \frac{1}{T_d}\left\{\sum_{t=1}^{T_n} -\Psi\left(\frac{p_m(\mathbf{y}_t;\boldsymbol{\theta})}{\nu p_n(\mathbf{y}_t)}\right) + \Psi'\left(\frac{p_m(\mathbf{y}_t;\boldsymbol{\theta})}{\nu p_n(\mathbf{y}_t)}\right)\frac{p_m(\mathbf{y}_t;\boldsymbol{\theta})}{\nu p_n(\mathbf{y}_t)} - \right.\\
& \left.\sum_{t=1}^{T_d}\Psi'\left(\frac{p_m(\mathbf{x}_t;\boldsymbol{\theta})}{\nu p_n(\mathbf{x}_t)}\right)\right\},
\end{aligned}
$$

  where Ψ is strictly convex, gives a consistent estimator.
- $\Psi(u) = u\ln u - (1+u)\ln(1+u)$ gives NCE. Are other Ψ more appropriate?
- The estimation is performed via optimization. Can we increase the computational efficiency by better optimization techniques?

Summary

- Introduction
  - What unnormalized models are and why being able to estimate them is important
  - Unnormalized models cannot be estimated by MLE (without approximations)
- Noise-contrastive estimation
  - Estimating unnormalized models by discriminating the observed data from artificial data with known distribution
  - Statistical and computational properties
- Application to the modeling of natural images
  - Formulated a two-layer model with a spline nonlinearity
  - In the second layer, sparse pooling of similarly oriented Gabor features emerged. The shape of the learned spline matches the sparsity of the feature outputs.

Three important points to retain

The big picture of parametric estimation is as follows:

Observe a collection $X = (\mathbf{x}_1, \ldots, \mathbf{x}_{T_d})$ of continuous or discrete random variables $\mathbf{x}_t$.

Assume that the $\mathbf{x}_t$ are iid and that their distribution $p_d(\mathbf{x})$ belongs to the family of nonnegative functions $\{p_m(\mathbf{x}; \boldsymbol{\theta})\}_{\boldsymbol{\theta}}$ parameterized by $\boldsymbol{\theta} \in \mathbb{R}^m$. That is $p_d(\mathbf{x}) = p_m(\mathbf{x}; \boldsymbol{\theta}^\star)$.

Find $\boldsymbol{\theta}^\star$

1. Many models p_m are unnormalized: only the shape of the pdf is modeled and not its scale. Such models do not integrate to one.
2. Normalizing them is problematic, but MLE is only applicable to normalized models.
3. Noise-contrastive estimation yields consistent estimates for unnormalized models and seems suitable for data with heavy tails.[a]

[a] Code available at `https://sites.google.com/site/michaelgutmann/code`.