Representation Learning — Module 1

Christopher Zach

Department of Electrical Engineering
Chalmers University

Lecture, Sep 2020

Outline

1. Introduction
2. Background
3. Proper scoring rules
   - Noise-contrastive estimation
   - Score matching
4. Exercise 1

Organization

- Frontal lecture with breaks and discussion segments

Monday	9-12	Intro, background and proper scoring rules
	13-17	Noise-contrastive estimation, score matching
Tuesday	9-12	Latent variable models I
	13-17	Latent variable models II

  Fika breaks at around 10:00 and 15:00

- Homework
  - Small-scale numerical experiments for hands-on experience
  - Form groups (ideally two students in each group)
  - Submit little report describing how your thoughts, implementation & results
  - "Engineer's notes"
  - Submit via email (`zach@chalmers.se`) by end of November

`https://drive.google.com/file/d/1aKP85-4R3z8QTks7dRwmTVgxjL0NWY2l/view?usp=sharing`
`https://drive.google.com/file/d/1JFn44JGIXXoO_sc0AHRNl6R4MhAEfA_9/view?usp=sharing`

What is representation learning?

What is representation learning?

From Wikipedia:

In machine learning, feature learning or representation learning is a set of techniques that allows a system to automatically discover the representations needed for feature detection or classification from raw data. This replaces manual feature engineering and allows a machine to both learn the features and use them to perform a specific task.
[. . .]
Feature learning can be either supervised or unsupervised.

- In supervised feature learning, features are learned using labeled input data. Examples include supervised neural networks, multilayer perceptron and (supervised) dictionary learning.
- *In unsupervised feature learning, features are learned with unlabeled input data. Examples include dictionary learning, independent component analysis, autoencoders, matrix factorization and various forms of clustering.*

What is representation learning?

- I do not fully agree with Wiki's definition of supervised feature learning
    - Features from supervised learning are often too task-specific
    - You want to extract information about data beyond a particular task

Representing data $=$ encoding+decoding data?

- Supervised feature learning includes (IMHO)
    - Multi-task learning
        - Shared DNN backbone for multiple tasks
        - Extract representations useful for a variety of problems
    - Weakly supervised tasks with virtually unlimited training data
        - Image colorization
        - Image completion / inpainting
        - Solving visual puzzles
        - Siamese DNNs to predict image relations e.g. from videos
        - Feature learnign using contrastive losses

I will not talk about any of these approaches
I will talk about *energy-based models (EBMs)*

Focus on established methods and a few extensions

What is representation learning?

Energy-based model (EBM)

An EBM is a function $E_\theta : \mathcal{X} \to \mathbb{R}_{\geq 0}$ (depending on parameters θ) that assigns a scalar energy value to an input. The interpretations of E_θ is as follows:

- $E_\theta(x) \approx 0$: x is "correct" or "likely"
- $E_\theta(x) \gg 0$: x is "incorrect" or "unlikely"

- Connecting some EBMs with probabilities
$$E_\theta(x) = -\log p_\theta(x)$$
- Some EBMs are unnormalized likelihoods
$$E_\theta(x) = -\log p_\theta(x) + c$$
- We may not always be interested in c
- Learning: estimate θ from training data $\{x_i\}$ such that

  $E_\theta(x) \approx 0$ for $x = x_i$ $\qquad$ $E_\theta(x)$ is large for $x \not\approx x_i$ for any i

- Often $E_\theta(x) \to \infty$ with $\|x\| \to \infty$
  - "Likely" data is a bounded region in $\mathcal{X}$
  - Example: dictionary learning

Energy-based model (EBM)

An EBM is a function $E_\theta : \mathcal{X} \to \mathbb{R}_{\geq 0}$ (depending on parameters θ) that assigns a scalar energy value to an input. The interpretations of E_θ is as follows:

- $E_\theta(x) \approx 0$: x is "correct" or "likely"
- $E_\theta(x) \gg 0$: x is "incorrect" or "unlikely"

- Connecting some EBMs with probabilities

$$E_\theta(x) = -\log p_\theta(x)$$

- Some EBMs are unnormalized likelihoods

$$E_\theta(x) = -\log p_\theta(x) + c$$

- We may not always be interested in c
- Learning: estimate θ from training data $\{x_i\}$ such that

$$E_\theta(x) \approx 0 \text{ for } x = x_i \qquad E_\theta(x) \text{ is large for } x \not\approx x_i \text{ for any } i$$

- Often $E_\theta(x) \to \infty$ with $\|x\| \to \infty$
  - "Likely" data is a bounded region in $\mathcal{X}$
  - Example: dictionary learning

Energy-based model (EBM)

An EBM is a function $E_\theta : \mathcal{X} \to \mathbb{R}_{\geq 0}$ (depending on parameters θ) that assigns a scalar energy value to an input. The interpretations of E_θ is as follows:

- $E_\theta(x) \approx 0$: x is "correct" or "likely"
- $E_\theta(x) \gg 0$: x is "incorrect" or "unlikely"

- Connecting some EBMs with probabilities

$$E_\theta(x) = -\log p_\theta(x)$$

- Some EBMs are unnormalized likelihoods

$$E_\theta(x) = -\log p_\theta(x) + c$$

- We may not always be interested in c
- Learning: estimate θ from training data $\{x_i\}$ such that

$$E_\theta(x) \approx 0 \text{ for } x = x_i \qquad E_\theta(x) \text{ is large for } x \not\approx x_i \text{ for any } i$$

- Often $E_\theta(x) \to \infty$ with $\|x\| \to \infty$
  - "Likely" data is a bounded region in $\mathcal{X}$
  - Example: dictionary learning

What is representation learning?

Energy-based model (EBM)

An EBM is a function $E_\theta : \mathcal{X} \to \mathbb{R}_{\geq 0}$ (depending on parameters θ) that assigns a scalar energy value to an input. The interpretations of E_θ is as follows:

- $E_\theta(x) \approx 0$: x is "correct" or "likely"
- $E_\theta(x) \gg 0$: x is "incorrect" or "unlikely"

- Connecting some EBMs with probabilities

$$E_\theta(x) = -\log p_\theta(x)$$

- Some EBMs are unnormalized likelihoods

$$E_\theta(x) = -\log p_\theta(x) + c$$

- We may not always be interested in c
- Learning: estimate θ from training data $\{x_i\}$ such that

$E_\theta(x) \approx 0$ for $x = x_i$ $\qquad$ $E_\theta(x)$ is large for $x \not\approx x_i$ for any i

- Often $E_\theta(x) \to \infty$ with $\|x\| \to \infty$
  - "Likely" data is a bounded region in $\mathcal{X}$
  - Example: dictionary learning

What is representation learning?

Energy-based model (EBM)

An EBM is a function $E_\theta : \mathcal{X} \to \mathbb{R}_{\geq 0}$ (depending on parameters θ) that assigns a scalar energy value to an input. The interpretations of E_θ is as follows:

- $E_\theta(x) \approx 0$: x is "correct" or "likely"
- $E_\theta(x) \gg 0$: x is "incorrect" or "unlikely"

- Connecting some EBMs with probabilities

$$E_\theta(x) = -\log p_\theta(x)$$

- Some EBMs are unnormalized likelihoods

$$E_\theta(x) = -\log p_\theta(x) + c$$

- We may not always be interested in c
- Learning: estimate θ from training data $\{x_i\}$ such that

$$E_\theta(x) \approx 0 \text{ for } x = x_i \qquad\qquad E_\theta(x) \text{ is large for } x \not\approx x_i \text{ for any } i$$

- Often $E_\theta(x) \to \infty$ with $\|x\| \to \infty$
  - "Likely" data is a bounded region in $\mathcal{X}$
  - Example: dictionary learning

Contents

Quickly changing research topic, 100s of new publications every year

Focus on fundamentals and established theory

- Introduction
- Background: a short recap on probability theory
- Proper scoring rules
  - General theory
  - Noise contrastive estimation
  - Score matching
- Latent variable models
  - Variational Bayes (VAE & VNCE)
  - Boltzmann machines
  - Dictionary learning and sparse coding

Caveat: I will focus on image and image-like data

Contents

Quickly changing research topic, 100s of new publications every year

Focus on fundamentals and established theory

- Introduction
- Background: a short recap on probability theory
- Proper scoring rules
  - General theory
  - Noise contrastive estimation
  - Score matching
- Latent variable models
  - Variational Bayes (VAE & VNCE)
  - Boltzmann machines
  - Dictionary learning and sparse coding

Caveat: I will focus on image and image-like data

Outline

A recap of probability theory

Why?

- One major branch of unsupervised representation learning is tightly connected to estimating parameters of probability distributions
- These distributions can be continuous or discrete
  - Measure-theoretic introduction of probabilities unifies and generalizes notation.

$$Pr(A) = \sum_{x \in A} p(x) \qquad \text{vs.} \qquad Pr(A) = \int_A p(x)dx.$$

  - It is a beautiful theory
- Recap of conditional probabilities, Bayes' theorem etc.

Probability space

A probability space space is a triple $(\Omega, \mathcal{F}, P)$, where

1. Ω is the sample set (think of Ω as elementary random events)
2. $\mathcal{F}$ is the set of possible events and forms a σ-algebra
   1. With $A \in \mathcal{F}$ we also have $(\Omega \setminus A) \in \mathcal{F}$
   2. $A_i \in \mathcal{F}$ for all $i \in \mathbb{N}$ we also have $\bigcup_{i=1}^\infty A_i \in \mathcal{F}$
3. $P : \mathcal{F} \to [0,1]$ is the probability measure satisfying $P(\Omega) = 1$ and

$$P\left(\bigcup_{i=1}^\infty A_i\right) = \sum_{i=1}^\infty P(A_i) \qquad \text{if } \{A_i : A_i \in \mathcal{F}\}_{i\in\mathbb{N}} \text{ are pairwise disjoint}$$

Null set

A null set $A \in \mathcal{F}$ has $P(A) = 0$.

- $\emptyset$ is always a null set, but there can be $A \neq \emptyset$ such that $P(A) = 0$.
- If $P(A) = 1$, then $P(\Omega \setminus A) = 0$ and $\Omega \setminus A$ is a null set. The event A occurs almost surely (a.s.).

Example

- Infinite number of dice rolls: $\Omega = \otimes_{i=1}^{\infty}\{1, \dots, 6\}$ (uncountable set of sequences). Strong law of large numbers,

$$P\left(\lim_{N\to\infty} \frac{\sum_{i=1}^{N} X_i}{N} = \frac{7}{2}\right) = 1.$$

$\lim_{N\to\infty} \sum_{i=1}^{N} X_i/N$ does not hold for every sample sequence, e.g. $A_1 = (1)_{i\in\mathbb{N}}$, but $P(A_1) = 0$. The probability of all these “atypical” sequences is 0.

Probability measures

Null set

A null set $A \in \mathcal{F}$ has $P(A) = 0$.

- $\emptyset$ is always a null set, but there can be $A \neq \emptyset$ such that $P(A) = 0$.
- If $P(A) = 1$, then $P(\Omega \setminus A) = 0$ and $\Omega \setminus A$ is a null set. The event A occurs almost surely (a.s.).

Example

- Infinite number of dice rolls: $\Omega = \otimes_{i=1}^{\infty}\{1, \ldots, 6\}$ (uncountable set of sequences). Strong law of large numbers,

$$P\left(\lim_{N\to\infty} \frac{\sum_{i=1}^{N} X_i}{N} = \frac{7}{2}\right) = 1.$$

$\lim_{N\to\infty} \sum_{i=1}^{N} X_i/N$ does not hold for every sample sequence, e.g. $A_1 = (1)_{i\in\mathbb{N}}$, but $P(A_1) = 0$. The probability of all these “atypical” sequences is 0.

Probability measures

Examples

1. $\Omega \subseteq \mathbb{N}$, $\mathcal{F} = 2^\Omega$ and P is given by

$$P(A) = \sum_{\omega \in A} p(\omega) \qquad \forall A \subseteq \Omega,$$

where $p : \Omega \to [0, 1]$ is the *probability mass function* with $\sum_{\omega \in \Omega} p(\omega) = 1$.

2. $\Omega = [0, 1]$. $\mathcal{F} = 2^\Omega$? No, more complicated. 2^Ω is too large to find a P with the right properties. Most useful working example: Borel algebras and Lebesgue measures.
3. Borel algebra $\mathcal{B}$: smallest σ-algebra that contains intervals $[a, b] \subseteq [0, 1]$ (or $[a, b] \subseteq \mathbb{R}$).
4. Lebesgue measure λ: extension of $\lambda([a, b]) = a - b$ to all sets from $\mathcal{B}$. Generalization to higher dimensions via Cartesian products. The Lebesgue measure corresponds to our intuitive notions of length, area and volume.

Probability measures

Examples

1. $\Omega \subseteq \mathbb{N}$, $\mathcal{F} = 2^\Omega$ and P is given by

$$P(A) = \sum_{\omega \in A} p(\omega) \qquad \forall A \subseteq \Omega,$$

   where $p : \Omega \to [0, 1]$ is the *probability mass function* with $\sum_{\omega\in\Omega} p(\omega) = 1$.

2. $\Omega = [0, 1]$. $\mathcal{F} = 2^\Omega$? No, more complicated. 2^Ω is too large to find a P with the right properties. Most useful working example: Borel algebras and Lebesgue measures.

3. Borel algebra $\mathcal{B}$: smallest σ-algebra that contains intervals $[a, b] \subseteq [0, 1]$ (or $[a, b] \subseteq \mathbb{R}$).

4. Lebesgue measure λ: extension of $\lambda([a, b]) = a - b$ to all sets from $\mathcal{B}$. Generalization to higher dimensions via Cartesian products. The Lebesgue measure corresponds to our intuitive notions of length, area and volume.

Probability measures

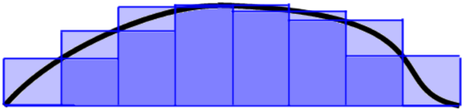
Examples

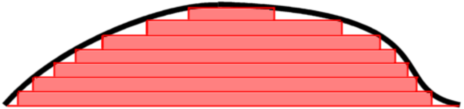
1. $\Omega \subseteq \mathbb{N}$, $\mathcal{F} = 2^{\Omega}$ and P is given by
$$P(A) = \sum_{\omega \in A} p(\omega) \qquad \forall A \subseteq \Omega,$$
where $p : \Omega \to [0, 1]$ is the *probability mass function* with $\sum_{\omega \in \Omega} p(\omega) = 1$.
2. $\Omega = [0, 1]$. $\mathcal{F} = 2^{\Omega}$? No, more complicated. 2^{Ω} is too large to find a P with the right properties. Most useful working example: Borel algebras and Lebesgue measures.
3. Borel algebra $\mathcal{B}$: smallest σ-algebra that contains intervals $[a, b] \subseteq [0, 1]$ (or $[a, b] \subseteq \mathbb{R}$).
4. Lebesgue measure λ: extension of $\lambda([a, b]) = a - b$ to all sets from $\mathcal{B}$. Generalization to higher dimensions via Cartesian products. The Lebesgue measure corresponds to our intuitive notions of length, area and volume.

Lebesgue integral

- Riemann integral: approximation of the area under a function via horizontal rectangles
- Lebesgue integral: approximation a function via superlevel sets (vertical slabs)
- Assume $f : \Omega \to [0, \infty)$, $A \in \mathcal{F}$, and a measure μ given

$$\int_A f \, d\mu = \int_0^\infty \mu(\{x \in A : f(x) > t\}) \, dt.$$





Densities

Absolute continuity

Let μ and ν be two measures on $(\Omega, \mathcal{F})$. We say ν is absolutely continuous w.r.t. μ ($\nu \ll \mu$) if $\mu(A) = 0$ implies $\nu(A) = 0$ for all $A \in \mathcal{F}$.

Radon-Nykodym

If $\nu \ll \mu$, then there exists a measureable function $f : \Omega \to \mathcal{F} \to [0, \infty)$ such that

$$\nu(A) = \int_A f(\omega) d\mu(\omega).$$

PMFs and PDFs

- Probability mass functions (pmfs) are densities w.r.t. the counting measure card(A)

$$P(A) = \sum_{x \in A} p(x)$$

- Probability density functions (pdfs) are densities w.r.t. the Lebesgue measure λ

$$\begin{aligned} P(A) &= \int_A f(\omega) d\lambda(\omega) \\ &= \int_A f(\omega) d\omega \qquad \text{if } f \text{ is continuous (Riemann-integrable)} \end{aligned}$$

TL;DR

Summation if the base measure is the counting measure, integral if the base measure is the Lebesgue measure (length of intervals).
I will use the integral notation for both discrete and continuous distributions.

Random variables

A random variable (RV) X is a measurable function $\Omega \to \mathbb{R}$.

$$Pr(X \leq x) = P(\{\omega \in \Omega : X(\omega) \leq x\}).$$

That is why X needs to be measurable.

- Random variables map general, possibly non-numerical events to numerical values.
- Allows to talk about expected value etc.,

$$\mathbb{E}[X] = \int_\Omega X(\omega) dP(\omega)$$

Law of the unconscious statistician (LOTUS)

$$\mathbb{E}[g(X)] = \int_\Omega g(X(\omega)) dP(\omega) = \begin{cases} \sum_x p(x) g(x) & p \text{ is a pmf} \\ \int f(x) g(x) dx & f \text{ is a pdf} \end{cases}$$

Random variables

A random variable (RV) X is a measurable function $\Omega \to \mathbb{R}$.

$$Pr(X \leq x) = P(\{\omega \in \Omega : X(\omega) \leq x\}).$$

That is why X needs to be measurable.

- Random variables map general, possibly non-numerical events to numerical values.
- Allows to talk about expected value etc.,

$$\mathbb{E}[X] = \int_\Omega X(\omega) dP(\omega)$$

Law of the unconscious statistician (LOTUS)

$$\mathbb{E}[g(X)] = \int_\Omega g(X(\omega)) dP(\omega) = \begin{cases} \sum_x p(x) g(x) & p \text{ is a pmf} \\ \int f(x) g(x) dx & f \text{ is a pdf} \end{cases}$$

Conditional probabilities

Conditional probability

Given two events $A, B \in \mathcal{F}$, define

$$P(A|B) := \frac{P(A \cap B)}{P(B)} \qquad \text{or} \qquad P(A \cap B) := P(A|B)P(B)$$

Further, let X be a RV and define a new RV

$$P(A|X = x)(\omega) := P(A|X(\omega) = x).$$

$P(A|X = x)$ is technically a RV mapping events to real numbers from $[0, 1]$.

Bayes' theorem

$$P(B|A) = \frac{P(A|B)P(B)}{P(A)} = \frac{P(A \cap B)}{P(A)}$$

Conditional probabilities

Marginal distribution / law of total probability

$$P(A) = P(A|B)P(B) + P(A|\bar{B})P(\bar{B}) \qquad \bar{B} := \Omega \setminus B$$

$$P(A) = \sum_i P(A|B_i)P(B_i) = \sum_i P(A \cap B_i) \qquad \{B_i\}_{i\in\mathbb{N}} \text{ is a partition of } \Omega$$

For joint distribution $P_{X,Y}(\cdot,\cdot)$

$$P_X(A) = \int \mathbf{1}[x \in A] dP_{X,Y}(x,y).$$

$P_X(A)$ is the marginal distribution. Often used for pmfs and pdfs:

$$p(x) = \sum_y p(x,y) \qquad\qquad p(x) = \int p(x,y)\,dy$$

Convexity

Convex functions

A function $f : \mathcal{X} \to \mathbb{R}$ is convex iff for all $x, x' \in \mathcal{X}$ and all $\alpha \in [0, 1]$ we have

$$f(\alpha x + (1 - \alpha)x') \leq \alpha f(x) + (1 - \alpha)f(x')$$

- Convex functions can be written as supremum of affine functions

$$f \text{ convex} \iff f(x) = \sup_\tau a_\tau^T x + b_\tau$$

- Useful sufficient conditions

$$f''(x) \geq 0 \qquad\qquad \nabla^2 f(x) \succeq 0 \qquad \forall x \in \mathcal{X}$$

Jensen's inequality

Let x be a RV with distribution p. Then

$$f\left(\mathbb{E}_{x \sim p}[x]\right) \leq \mathbb{E}_{x \sim p}[f(x)]$$

Outline

Main scenario

Estimate distribution parameters to fit given data

Given data samples $\{x_1, \dots, x_N\} \sim p_d(x)$ and parametric distribution $p_\theta(\cdot)$

$$\theta^* = \arg\min_{\theta \in \mathbb{R}^n} \sum_{i=1}^{M} \ell(p_\theta(x), x_i)$$

What are good choices for ℓ?

Issues to consider

- We have only access to samples $x \sim p_d$, not p_d itself
- We would like that θ^* to approach the true parameters with $M \to \infty$
  - Consistency of the estimate (a.s.)
  - Requires that $p_d = p_{\hat{\theta}}$ for some $\hat{\theta}$
- Our model distribution p_θ is usually unnormalized

Main scenario

Discussion

1. What is the connection to representation / feature learning?
2. How would you estimate the parameters of p_θ from data samples?
3. Why will p_θ often be unnormalized?

Main scenario

Estimate distribution parameters to fit given data

Given data samples $\{x_1, \dots, x_N\} \sim p_d(x)$ and parametric distribution $p_\theta(\cdot)$

$$\theta^* = \arg\min_{\theta \in \mathbb{R}^n} \sum\nolimits_{i=1}^{M} \ell(p_\theta(x), x_i)$$

- Maximize log-likelihood: $\max_\theta \sum_i \log p_\theta(x_i)$
  - $\ell(p, x) = -\log p(x)$
  - Only works for “simple” distributions p_θ
- VAE (variational auto-encoder, auto-encoding variational Bayes)
  - ℓ is upper bound on $-\log p_\theta(x)$
- Proper scoring rules
  - In one line: minimize Bregman divergence between $p_d(x)$ and $p_\theta(x)$
  - Noise-contrastive estimation, score matching
- f-divergences (generative adversarial networks, GANs)
- Wasserstein distance
- Normalizing flows

Main scenario

Estimate distribution parameters to fit given data

Given data samples $\{x_1, \dots, x_N\} \sim p_d(x)$ and parametric distribution $p_\theta(\cdot)$

$$\theta^* = \arg\min_{\theta \in \mathbb{R}^n} \sum_{i=1}^{M} \ell(p_\theta(x), x_i)$$

- Maximize log-likelihood: $\max_\theta \sum_i \log p_\theta(x_i)$
  - $\ell(p, x) = -\log p(x)$
  - Only works for "simple" distributions p_θ
- VAE (variational auto-encoder, auto-encoding variational Bayes)
  - ℓ is upper bound on $-\log p_\theta(x)$
- Proper scoring rules
  - In one line: minimize Bregman divergence between $p_d(x)$ and $p_\theta(x)$
  - Noise-contrastive estimation, score matching
- f-divergences (generative adversarial networks, GANs)
- Wasserstein distance
- Normalizing flows

Main scenario

Estimate distribution parameters to fit given data

Given data samples $\{x_1, \dots, x_N\} \sim p_d(x)$ and parametric distribution $p_\theta(\cdot)$

$$\theta^* = \arg\min_{\theta \in \mathbb{R}^n} \sum\nolimits_{i=1}^{M} \ell(p_\theta(x), x_i)$$

- Maximize log-likelihood: $\max_\theta \sum_i \log p_\theta(x_i)$
  - $\ell(p, x) = -\log p(x)$
  - Only works for “simple” distributions p_θ
- VAE (variational auto-encoder, auto-encoding variational Bayes)
  - ℓ is upper bound on $-\log p_\theta(x)$
- Proper scoring rules
  - In one line: minimize Bregman divergence between $p_d(x)$ and $p_\theta(x)$
  - Noise-contrastive estimation, score matching
- f-divergences (generative adversarial networks, GANs)
- Wasserstein distance
- Normalizing flows

Main scenario

Estimate distribution parameters to fit given data

Given data samples $\{x_1, \dots, x_N\} \sim p_d(x)$ and parametric distribution $p_\theta(\cdot)$

$$\theta^* = \arg\min_{\theta \in \mathbb{R}^n} \sum_{i=1}^{M} \ell(p_\theta(x), x_i)$$

- Maximize log-likelihood: $\max_\theta \sum_i \log p_\theta(x_i)$
  - $\ell(p, x) = -\log p(x)$
  - Only works for “simple” distributions p_θ
- VAE (variational auto-encoder, auto-encoding variational Bayes)
  - ℓ is upper bound on $-\log p_\theta(x)$
- Proper scoring rules
  - In one line: minimize Bregman divergence between $p_d(x)$ and $p_\theta(x)$
  - Noise-contrastive estimation, score matching
- f-divergences (generative adversarial networks, GANs)
- Wasserstein distance
- Normalizing flows

Proper scoring rules

Overview

Proper scoring rules (PSRs) are one way to design losses ℓ with certain performance guarantees.

- PSRs unify a number of proposed methods for learning distributions
  - Maximum likelihood estimation
  - Score matching
  - Noise-contrastive estimation
  - Learning of graphical models using pseudo-likelihoods

- T. Gneiting & A.E. Raftery, “Strictly proper scoring rules, prediction, and estimation”
- A.P. Dawid & M. Musio, “Theory and applications of proper scoring rules”

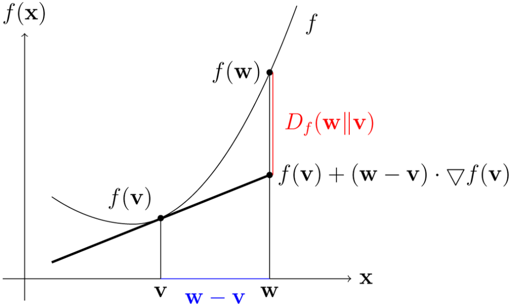
Bregman divergence

Bregman divergence

Let f be a diff'able convex function. The Bregman divergence $D_f(w\|v)$ is defined as

$$D_f(w\|v) := f(w) - \big(f(v) + \nabla f(v)^T (w - v)\big).$$

- $D_f(w\|v) \geq 0$.
- If f is strictly convex, then $D_f(w\|v) = 0$ iff $w = v$.
- $D_f(w\|v)$ can be interpreted as linearization error of a convex function
- f linearized at v linearization error measured at w



Proper scoring rules

- Let $\Omega = \{1, \dots, K\}$ be a finite set
- $p, q \in \mathbb{P}(\Omega)$ be probability measures

$$\mathbb{P}(\Omega) = \left\{ p \in [0,1]^K : \sum\nolimits_{x=1}^K p(x) = 1 \right\}$$

- $F : \mathbb{P}(\Omega) \to \mathbb{R}$ be a diff'able convex function

$$\begin{aligned} D_F(p\|q) &= F(p) - F(q) - \nabla F(q)^T (p - q) \\ &= F(p) - F(q) - \sum_{x=1}^K \frac{\partial F(q)}{\partial q(x)} (p(x) - q(x)) \end{aligned}$$

Minimize $D_F(p\|q)$ w.r.t. $q = (q_1, \dots, q_K)$:

$$\begin{aligned} \arg\min_q D_F(p\|q) &= \arg\min_q -F(q) - \sum_{x=1}^K \frac{\partial F(q)}{\partial q(x)} (p(x) - q(x)) \\ &= \arg\max_q F(q) + \sum_{x=1}^K \frac{\partial F(q)}{\partial q(x)} (p(x) - q(x)) \end{aligned}$$

Proper scoring rules

- Let $\Omega = \{1, \dots, K\}$ be a finite set
- $p, q \in \mathbb{P}(\Omega)$ be probability measures

$$\mathbb{P}(\Omega) = \left\{ p \in [0,1]^K : \sum\nolimits_{x=1}^{K} p(x) = 1 \right\}$$

- $F : \mathbb{P}(\Omega) \to \mathbb{R}$ be a diff'able convex function

$$\begin{aligned} D_F(p\|q) &= F(p) - F(q) - \nabla F(q)^T (p - q) \\ &= F(p) - F(q) - \sum_{x=1}^{K} \frac{\partial F(q)}{\partial q(x)} (p(x) - q(x)) \end{aligned}$$

Minimize $D_F(p\|q)$ w.r.t. $q = (q_1, \dots, q_K)$:

$$\begin{aligned} \arg\min_q D_F(p\|q) &= \arg\min_q -F(q) - \sum_{x=1}^{K} \frac{\partial F(q)}{\partial q(x)} (p(x) - q(x)) \\ &= \arg\max_q F(q) + \sum_{x=1}^{K} \frac{\partial F(q)}{\partial q(x)} (p(x) - q(x)) \end{aligned}$$

Proper scoring rules

Using $\sum p(x) = 1$:

$$q^* = \arg\max_q F(q) + \sum_{x=1}^K \frac{\partial F(q)}{\partial q(x)} (p(x) - q(x))$$

$$= \arg\max_q \sum_{x=1}^K p(x) \left(F(q) + \frac{\partial F(q)}{\partial q(x)} \right) - \sum_{x=1}^K \frac{\partial F(q)}{\partial q(x)} q(x)$$

$$= \arg\max_q \sum_{x=1}^K p(x) \underbrace{\left(F(q) + \frac{\partial F(q)}{\partial q(x)} - \sum_{x'=1}^K \frac{\partial F(q)}{\partial q(x')} q(x') \right)}_{=:S(q,x)}$$

$$= \arg\max_q \sum_{x=1}^K p(x) S(q,x) = \arg\max_q \mathbb{E}_{x\sim p} [S(q,x)]$$

Using $\sum p(x) = 1$:

$$q^* = \arg\max_q F(q) + \sum_{x=1}^{K} \frac{\partial F(q)}{\partial q(x)} (p(x) - q(x))$$

$$= \arg\max_q \sum_{x=1}^{K} p(x) \left(F(q) + \frac{\partial F(q)}{\partial q(x)} \right) - \sum_{x=1}^{K} \frac{\partial F(q)}{\partial q(x)} q(x)$$

$$= \arg\max_q \sum_{x=1}^{K} p(x) \underbrace{\left(F(q) + \frac{\partial F(q)}{\partial q(x)} - \sum_{x'=1}^{K} \frac{\partial F(q)}{\partial q(x')} q(x') \right)}_{=:S(q,x)}$$

$$= \arg\max_q \sum_{x=1}^{K} p(x) S(q,x) = \arg\max_q \mathbb{E}_{x\sim p}\left[S(q,x)\right]$$

Proper scoring rules

Using $\sum p(x) = 1$:

$$\begin{aligned}
q^* &= \arg\max_q F(q) + \sum_{x=1}^K \frac{\partial F(q)}{\partial q(x)} (p(x) - q(x)) \\
&= \arg\max_q \sum_{x=1}^K p(x) \left(F(q) + \frac{\partial F(q)}{\partial q(x)} \right) - \sum_{x=1}^K \frac{\partial F(q)}{\partial q(x)} q(x) \\
&= \arg\max_q \sum_{x=1}^K p(x) \underbrace{\left(F(q) + \frac{\partial F(q)}{\partial q(x)} - \sum_{x'=1}^K \frac{\partial F(q)}{\partial q(x')} q(x') \right)}_{=:S(q,x)} \\
&= \arg\max_q \sum_{x=1}^K p(x) S(q,x) = \arg\max_q \mathbb{E}_{x\sim p} [S(q,x)]
\end{aligned}$$

Proper scoring rules

Using $\sum p(x) = 1$:

$$
\begin{aligned}
q^* &= \arg\max_q F(q) + \sum_{x=1}^K \frac{\partial F(q)}{\partial q(x)} (p(x) - q(x)) \\
&= \arg\max_q \sum_{x=1}^K p(x) \left(F(q) + \frac{\partial F(q)}{\partial q(x)} \right) - \sum_{x=1}^K \frac{\partial F(q)}{\partial q(x)} q(x) \\
&= \arg\max_q \sum_{x=1}^K p(x) \underbrace{\left(F(q) + \frac{\partial F(q)}{\partial q(x)} - \sum_{x'=1}^K \frac{\partial F(q)}{\partial q(x')} q(x') \right)}_{=:S(q,x)} \\
&= \arg\max_q \sum_{x=1}^K p(x) S(q,x) = \arg\max_q \mathbb{E}_{x \sim p} \left[S(q,x) \right]
\end{aligned}
$$

Proper scoring rules

Proper scoring rule (PSR)

For a diff'able convex function F define

$$S(q,x) := F(q) + \frac{\partial F(q)}{\partial q(x)} - \sum_{x'=1}^{K} \frac{\partial F(q)}{\partial q(x')} q(x')$$

$$S(q,p) := \mathbb{E}_{x\sim p}\big[S(q,x)\big]$$

S is called a proper scoring rule (PSR). If F is strictly convex, then S is a strictly PSR.

- Historically, the convention is to maximize PSRs (higher scores are better)
- What about $\Omega = \mathbb{R}^D$?

$$S(q,x) := F(q) + \frac{\partial F(q)}{\partial q(x)} - \int \frac{\partial F(q)}{\partial q(x')} q(x')\, dx'$$

Proper scoring rules

- F can be recovered from S via

$$
\begin{aligned}
S(q,q) &= \mathbb{E}_{x\sim q}\big[S(q,x)\big] \\
&= \sum_x q(x)\left(F(q) + \frac{\partial F(q)}{\partial q(x)} - \sum_{x'} \frac{\partial F(q)}{\partial q(x')} q(x')\right) \\
&= F(q) + \sum_x q(x) \frac{\partial F(q)}{\partial q(x)} - \sum_x q(x) \sum_{x'} \frac{\partial F(q)}{\partial q(x')} q(x') \\
&= F(q) + \sum_x q(x) \frac{\partial F(q)}{\partial q(x)} - \sum_{x'} \frac{\partial F(q)}{\partial q(x')} q(x') \\
&= F(q)
\end{aligned}
$$

- $S(q,q) = F(q)$ can also be interpreted as generalized (negated) entropy

Proper scoring rules

- F can be recovered from S via

$$\begin{aligned} S(q,q) &= \mathbb{E}_{x\sim q}\big[S(q,x)\big] \\ &= \sum_x q(x)\left(F(q) + \frac{\partial F(q)}{\partial q(x)} - \sum_{x'} \frac{\partial F(q)}{\partial q(x')} q(x')\right) \\ &= F(q) + \sum_x q(x)\frac{\partial F(q)}{\partial q(x)} - \sum_x q(x) \sum_{x'} \frac{\partial F(q)}{\partial q(x')} q(x') \\ &= F(q) + \sum_x q(x)\frac{\partial F(q)}{\partial q(x)} - \sum_{x'} \frac{\partial F(q)}{\partial q(x')} q(x') \\ &= F(q) \end{aligned}$$

- $S(q,q) = F(q)$ can also be interpreted as generalized (negated) entropy

Proper scoring rules

- F can be recovered from S via

$$
\begin{aligned}
S(q,q) &= \mathbb{E}_{x\sim q}\big[S(q,x)\big] \\
&= \sum_x q(x)\left(F(q) + \frac{\partial F(q)}{\partial q(x)} - \sum_{x'} \frac{\partial F(q)}{\partial q(x')} q(x')\right) \\
&= F(q) + \sum_x q(x)\frac{\partial F(q)}{\partial q(x)} - \sum_x q(x) \sum_{x'} \frac{\partial F(q)}{\partial q(x')} q(x') \\
&= F(q) + \sum_x q(x)\frac{\partial F(q)}{\partial q(x)} - \sum_{x'} \frac{\partial F(q)}{\partial q(x')} q(x') \\
&= F(q)
\end{aligned}
$$

- $S(q,q) = F(q)$ can also be interpreted as generalized (negated) entropy

Proper scoring rules

- F can be recovered from S via

$$\begin{aligned}
S(q,q) &= \mathbb{E}_{x\sim q}\left[S(q,x)\right] \\
&= \sum_x q(x)\left(F(q) + \frac{\partial F(q)}{\partial q(x)} - \sum_{x'} \frac{\partial F(q)}{\partial q(x')} q(x')\right) \\
&= F(q) + \sum_x q(x)\frac{\partial F(q)}{\partial q(x)} - \sum_x q(x) \sum_{x'} \frac{\partial F(q)}{\partial q(x')} q(x') \\
&= F(q) + \sum_x q(x)\frac{\partial F(q)}{\partial q(x)} - \sum_{x'} \frac{\partial F(q)}{\partial q(x')} q(x') \\
&= F(q)
\end{aligned}$$

- $S(q,q) = F(q)$ can also be interpreted as generalized (negated) entropy

Proper scoring rules

- F can be recovered from S via

$$
\begin{aligned}
S(q,q) &= \mathbb{E}_{x\sim q}\left[S(q,x)\right] \\
&= \sum_x q(x)\left(F(q) + \frac{\partial F(q)}{\partial q(x)} - \sum_{x'} \frac{\partial F(q)}{\partial q(x')} q(x')\right) \\
&= F(q) + \sum_x q(x)\frac{\partial F(q)}{\partial q(x)} - \sum_x q(x) \sum_{x'} \frac{\partial F(q)}{\partial q(x')} q(x') \\
&= F(q) + \sum_x q(x)\frac{\partial F(q)}{\partial q(x)} - \sum_{x'} \frac{\partial F(q)}{\partial q(x')} q(x') \\
&= F(q)
\end{aligned}
$$

- $S(q,q) = F(q)$ can also be interpreted as generalized (negated) entropy

Why are PSRs useful?

Monte Carlo approximation of $D_F(p\|q)$

$$-D_F(p\|q) \doteq \mathbb{E}_{x\sim p}\left[S(q,x)\right] \approx \frac{1}{N}\sum\nolimits_i S(q,x_i)$$

We only need iid samples $(x_i)_{i=1}^N$ with $x_i \sim p$

- Let S be a strictly PSR. With $N \to \infty$:

$$\frac{1}{N}\sum\nolimits_i S(q,x_i) \overset{\text{a.s.}}{\to} \mathbb{E}_{x\sim p}\left[S(q,x)\right] \doteq -D_F(q\|p)$$

Since $D_F(p\|q) = 0$ iff $q = p$ (modulo null sets)

- $p = q^* = \arg\max_q -D_F(q\|p) = \arg\min_q D_F(q\|p)$ unique solution
- q^* consistent estimator of p

Proper scoring rules: examples

- Logarithmic scoring rule
- Let $F(p) = \sum_x p(x) \log p(x)$ be the (negated) Shannon entropy

$$
\begin{aligned}
S(q,x) &= F(q) + \frac{\partial F(q)}{\partial q(x)} - \sum_{x'=1}^{K} \frac{\partial F(q)}{\partial q(x')} q(x') \\
&= F(q) + 1 + \log q(x) - \sum_{x'=1}^{K} q(x') \left(1 + \log q(x')\right) \\
&= \log q(x) + F(q) - \underbrace{\sum_{x'=1}^{K} q(x') \log q(x')}_{=F(q)} + 1 - \underbrace{\sum_{x'=1}^{K} q(x)}_{=1} \\
&= \log q(x)
\end{aligned}
$$

- Maximum likelihood!
- $S(q,x) = \log q(x)$ depends only on $q(x)$ but not on $q(x')$ for any $x' \neq x$
- *Local* proper scoring rule

Proper scoring rules: examples

- Logarithmic scoring rule
- Let $F(p) = \sum_x p(x) \log p(x)$ be the (negated) Shannon entropy

$$\begin{aligned}
S(q,x) &= F(q) + \frac{\partial F(q)}{\partial q(x)} - \sum_{x'=1}^{K} \frac{\partial F(q)}{\partial q(x')} q(x') \\
&= F(q) + 1 + \log q(x) - \sum_{x'=1}^{K} q(x') \left(1 + \log q(x')\right) \\
&= \log q(x) + F(q) - \underbrace{\sum_{x'=1}^{K} q(x') \log q(x')}_{=F(q)} + 1 - \underbrace{\sum_{x'=1}^{K} q(x)}_{=1} \\
&= \log q(x)
\end{aligned}$$

- Maximum likelihood!
- $S(q,x) = \log q(x)$ depends only on $q(x)$ but not on $q(x')$ for any $x' \neq x$
- *Local* proper scoring rule

Proper scoring rules: examples

- Logarithmic scoring rule
- Let $F(p) = \sum_x p(x) \log p(x)$ be the (negated) Shannon entropy

$$
\begin{aligned}
S(q,x) &= F(q) + \frac{\partial F(q)}{\partial q(x)} - \sum_{x'=1}^{K} \frac{\partial F(q)}{\partial q(x')} q(x') \\
&= F(q) + 1 + \log q(x) - \sum_{x'=1}^{K} q(x') \left(1 + \log q(x')\right) \\
&= \log q(x) + F(q) - \underbrace{\sum_{x'=1}^{K} q(x') \log q(x')}_{=F(q)} + 1 - \underbrace{\sum_{x'=1}^{K} q(x)}_{=1} \\
&= \log q(x)
\end{aligned}
$$

- Maximum likelihood!
- $S(q,x) = \log q(x)$ depends only on $q(x)$ but not on $q(x')$ for any $x' \neq x$
- *Local* proper scoring rule

Proper scoring rules: examples

- Logarithmic scoring rule
- Let $F(p) = \sum_x p(x) \log p(x)$ be the (negated) Shannon entropy

$$
\begin{aligned}
S(q,x) &= F(q) + \frac{\partial F(q)}{\partial q(x)} - \sum_{x'=1}^{K} \frac{\partial F(q)}{\partial q(x')} q(x') \\
&= F(q) + 1 + \log q(x) - \sum_{x'=1}^{K} q(x') \left(1 + \log q(x')\right) \\
&= \log q(x) + F(q) - \underbrace{\sum_{x'=1}^{K} q(x') \log q(x')}_{=F(q)} + 1 - \underbrace{\sum_{x'=1}^{K} q(x)}_{=1} \\
&= \log q(x)
\end{aligned}
$$

- Maximum likelihood!
- $S(q,x) = \log q(x)$ depends only on $q(x)$ but not on $q(x')$ for any $x' \neq x$
- *Local* proper scoring rule

Proper scoring rules: examples

- Quadratic (or Brier) scoring rule

$$F(p) = \frac{1}{2}\|p\|^2 = \frac{1}{2}\sum_x p(x)^2$$

- Exercise for now: what is $S(q,x)$?
- Recall

$$S(q,x) = F(q) + \frac{\partial F(q)}{\partial q(x)} - \sum_{x'=1}^{K} \frac{\partial F(q)}{\partial q(x')} q(x')$$

Proper scoring rules: examples

- Quadratic (or Brier) scoring rule: $F(p) = \frac{1}{2}\|p\|^2$
  - Yields $S(q,x) = q(x) - \|q\|^2/2$
  - Non-local PSR

$$\|q\|^2 = \sum_{x'} q(x)^2 = \langle q, q\rangle_{\mathbb{R}^K} \qquad \|q\|_{L_2}^2 = \int q(x)^2\, dx = \langle q, q\rangle_{L_2}$$

- Spherical scoring rule with $S(q,x) = q(x)/\|q\|$
- *Local PSR* only depend on $q(x)$, not on $q(x')$ for $x' \neq x$
  - Logarithmic score
  - Hyvärinen score (score matching): depends on derivatives of $q(x)$
- $S(q,x) = q(x)$ is *not* a PSR!
  - Discussion: optimal parameter μ of a Gaussian (with $\sigma = 1$)

$$\max_{\mu \in \mathbb{R}} \frac{1}{\sqrt{2\pi}} \sum_i \exp\left(-\frac{(x_i - \mu)^2}{2}\right)$$

How is this different from a MLE?

Proper scoring rules: examples

- Quadratic (or Brier) scoring rule: $F(p) = \frac{1}{2}\|p\|^2$
  - Yields $S(q,x) = q(x) - \|q\|^2/2$
  - Non-local PSR

$$\|q\|^2 = \sum_{x'} q(x)^2 = \langle q, q \rangle_{\mathbb{R}^K} \qquad \|q\|_{L_2}^2 = \int q(x)^2\, dx = \langle q, q \rangle_{L_2}$$

- Spherical scoring rule with $S(q,x) = q(x)/\|q\|$
- *Local PSR* only depend on $q(x)$, not on $q(x')$ for $x' \neq x$
  - Logarithmic score
  - Hyvärinen score (score matching): depends on derivatives of $q(x)$
- $S(q,x) = q(x)$ is *not* a PSR!
  - Discussion: optimal parameter μ of a Gaussian (with $\sigma = 1$)

$$\max_{\mu \in \mathbb{R}} \frac{1}{\sqrt{2\pi}} \sum_i \exp\left(-\frac{(x_i - \mu)^2}{2}\right)$$

How is this different from a MLE?

Proper scoring rules: examples

- Quadratic (or Brier) scoring rule: $F(p) = \frac{1}{2}\|p\|^2$
  - Yields $S(q,x) = q(x) - \|q\|^2/2$
  - Non-local PSR

$$\|q\|^2 = \sum_{x'} q(x)^2 = \langle q, q\rangle_{\mathbb{R}^K} \qquad \|q\|_{L_2}^2 = \int q(x)^2\,dx = \langle q, q\rangle_{L_2}$$

- Spherical scoring rule with $S(q,x) = q(x)/\|q\|$
- *Local PSR* only depend on $q(x)$, not on $q(x')$ for $x' \neq x$
  - Logarithmic score
  - Hyvärinen score (score matching): depends on derivatives of $q(x)$
- $S(q,x) = q(x)$ is *not* a PSR!
  - Discussion: optimal parameter μ of a Gaussian (with $\sigma = 1$)

$$\max_{\mu \in \mathbb{R}} \frac{1}{\sqrt{2\pi}} \sum_i \exp\left(-\frac{(x_i - \mu)^2}{2}\right)$$

How is this different from a MLE?

Proper scoring rules: examples

- Quadratic (or Brier) scoring rule: $F(p) = \frac{1}{2}\|p\|^2$
  - Yields $S(q,x) = q(x) - \|q\|^2/2$
  - Non-local PSR

$$\|q\|^2 = \sum_{x'} q(x)^2 = \langle q, q \rangle_{\mathbb{R}^K} \qquad \|q\|_{L_2}^2 = \int q(x)^2 \, dx = \langle q, q \rangle_{L_2}$$

- Spherical scoring rule with $S(q,x) = q(x)/\|q\|$
- *Local PSR* only depend on $q(x)$, not on $q(x')$ for $x' \neq x$
  - Logarithmic score
  - Hyvärinen score (score matching): depends on derivatives of $q(x)$
- $S(q,x) = q(x)$ is *not* a PSR!
  - Discussion: optimal parameter μ of a Gaussian (with $\sigma = 1$)

$$\max_{\mu \in \mathbb{R}} \frac{1}{\sqrt{2\pi}} \sum_i \exp\left(-\frac{(x_i - \mu)^2}{2}\right)$$

How is this different from a MLE?

Proper scoring rules for Bernoulli RVs

- Let z be a Bernoulli RV

$$p_\theta(z=1)=\theta \qquad\qquad p_\theta(z=0)=1-\theta$$

  for a $\theta \in [0,1]$
- N training samples $\{z_i\}$
- ηN ones in the training set with $\eta \in (0,1)$
- Logarithmic PSR

$$\max_{\theta\in(0,1)} \eta\log\theta + (1-\eta)\log(1-\theta)$$

- First order optimality

$$\frac{\eta}{\theta} - \frac{1-\eta}{1-\theta} \stackrel{!}{=} 0 \iff \frac{\eta}{\theta} = \frac{1-\eta}{1-\theta} \iff \eta(1-\theta) = (1-\eta)\theta \iff \eta=\theta$$

Proper scoring rules for Bernoulli RVs

- Let z be a Bernoulli RV

$$p_\theta(z=1)=\theta \qquad\qquad p_\theta(z=0)=1-\theta$$

  for a $\theta \in [0,1]$
- N training samples $\{z_i\}$
- ηN ones in the training set with $\eta \in (0,1)$
- Quadratic PSR

$$\begin{aligned} &\max_{\theta\in(0,1)} \eta\left(\theta - \frac{1}{2}\left\|\begin{pmatrix}\theta\\1-\theta\end{pmatrix}\right\|^2\right) + (1-\eta)\left(1-\theta-\frac{1}{2}\left\|\begin{pmatrix}\theta\\1-\theta\end{pmatrix}\right\|^2\right) \\ &= \max_{\theta\in(0,1)} \eta\theta + (1-\eta)(1-\theta) - \frac{1}{2}\left\|\begin{pmatrix}\theta\\1-\theta\end{pmatrix}\right\|^2 \end{aligned}$$

- First order optimality

$$0 \stackrel{!}{=} \eta - (1-\eta) - \theta - (\theta - 1) \iff 2\eta = 2\theta$$

Proper scoring rules

Recall our starting point:

Estimate distribution parameters to fit given data

Given data samples $\{x_1, \dots, x_N\} \sim p_d(x)$ and parametric distribution $p_\theta(\cdot)$

$$\theta^* = \arg\min_{\theta \in \mathbb{R}^n} \sum_{i=1}^{M} \ell(p_\theta(\cdot), x_i)$$

What are good choices for ℓ?

We have now one answer:

Choose a strictly convex and diff'able function F and set

$$\ell(p_\theta, x) = -S(p_\theta, x),$$

where S is the PSR induced by F.

Big assumption

There exists a $\hat{\theta}$ such that $p_d = p_{\hat{\theta}}$.

Discussions

- Mis-specification: what happens if p_d cannot be represented as $p_{\hat{\theta}}$?
- Robustness: which PSRs might be more robust when the training data is contaminated by outliers?
- Ideal setting: why are we not done yet?

Proper scoring rules

Big assumption

There exists a $\hat{\theta}$ such that $p_d = p_{\hat{\theta}}$.

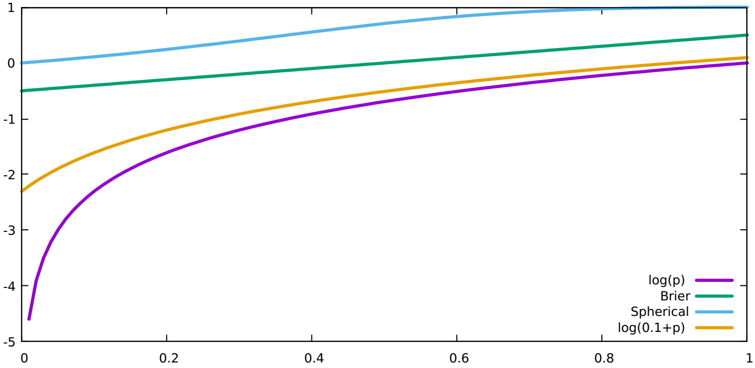
Discussions

- Mis-specification: what happens if p_d cannot be represented as $p_{\hat{\theta}}$?
  - *Different PSRs lead to different estimates* θ^*
- Robustness: which PSRs might be more robust when the training data is contaminated by outliers?
- Ideal setting: why are we not done yet?

Proper scoring rules

Discussion: Robustness

Which PSRs might be more robust when the training data is contaminated by outliers?



Proper scoring rules

Discussion

Why are we not done yet?

- Non-local PSRs only easy to use for discrete (categorical) RVs
  - For continuous RVs we need to compute e.g., $\int q(x)^2\, dx$
  - At least as difficult as computing $Z = \int q(x)\, dx$
- Non-local PSRs also not tractable for discrete RVs with many states
  - Quantized images: $256^{3\times 1000\times 1000}$ possible values
- Logarithmic PSR is maximum likelihood
  - We need to work with normalized models
- PSRs do not work directly with latent variable models

Proper scoring rules

We give examples addressing some issues:

- Working with unnormalized models: NCE and score matching
- Working with latent variable models: variational Bayes and VAE
- Working with unnormalized and latent variable models: VNCE

Outline

Noise-contrastive estimation

Overview

Noise-contrastive estimation (NCE)

- casts estimation of distribution parameters as supervised learning problem
- jointly estimates the unknown partition function / normalization constant
- applies logarithmic PSR to estimate parameters of a binary RV

- T. Hastie, R. Tibshirani & J. Friedman, “The Elements of Statistical Learning”, Sec. 14.2.4
- M. Gutmann & A. Hyvärinen, “Noise-contrastive estimation: A new estimation principle for unnormalized statistical models”

Noise-contrastive estimation

Given

- Samples $\{x_1, \dots, x_N\}$ from unknown data distribution p_d
- Fully known noise distribution p_n (e.g. multi-variate Gaussian)
  - We can evaluate $p_n(x)$ for any x and sample from p_n easily

Goal: estimate parameters of p_θ that models/approximates p_d

Idea:

- Randomly choose $z \in \{0, 1\}$ whether to draw a sample from p_d or p_n

$$p_{d,n}(x|z=0) = p_d(x)$$
$$p_{d,n}(x|z=1) = p_n(x)$$

- We use a prior $p(z)$ on z: select $\eta \in (0, 1)$

$$p(z=0) = \eta \qquad\qquad p(z=1) = 1 - \eta$$

Exercise now: what is $p_{d,n}(z|x)$?

Noise-contrastive estimation

- Recall Bayes' theorem

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

- In our setting

$$p_{d,n}(z|x) = \frac{p_{d,n}(x|z)p(z)}{p_{d,n}(x)}$$

- From the previous slide

$$p_{d,n}(x|z=0) = p_d(x) \qquad p(z=0) = \eta$$
$$p_{d,n}(x|z=1) = p_n(x) \qquad p(z=1) = 1-\eta$$

- Recall law of total probability

$$\begin{aligned} p_{d,n}(x) &= p_{d,n}(x|z=0)p(z=0) + p_{d,n}(x|z=1)p(z=1) \\ &= p_d(x)\eta + p_n(x)(1-\eta) \end{aligned}$$

Noise-contrastive estimation

- Recall Bayes' theorem

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

- In our setting

$$p_{d,n}(z|x) = \frac{p_{d,n}(x|z)p(z)}{p_{d,n}(x)}$$

- From the previous slide

$$p_{d,n}(x|z=0) = p_d(x) \qquad\qquad p(z=0) = \eta$$
$$p_{d,n}(x|z=1) = p_n(x) \qquad\qquad p(z=1) = 1-\eta$$

- Recall law of total probability

$$\begin{aligned} p_{d,n}(x) &= p_{d,n}(x|z=0)p(z=0) + p_{d,n}(x|z=1)p(z=1) \\ &= p_d(x)\eta + p_n(x)(1-\eta) \end{aligned}$$

Noise-contrastive estimation

- Recall Bayes' theorem

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

- In our setting

$$p_{d,n}(z|x) = \frac{p_{d,n}(x|z)p(z)}{p_{d,n}(x)}$$

- From the previous slide

$$p_{d,n}(x|z=0) = p_d(x) \qquad p(z=0) = \eta$$
$$p_{d,n}(x|z=1) = p_n(x) \qquad p(z=1) = 1-\eta$$

- Recall law of total probability

$$\begin{aligned} p_{d,n}(x) &= p_{d,n}(x|z=0)p(z=0) + p_{d,n}(x|z=1)p(z=1) \\ &= p_d(x)\eta + p_n(x)(1-\eta) \end{aligned}$$

Noise-contrastive estimation

- Recall Bayes' theorem

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

- In our setting

$$p_{d,n}(z|x) = \frac{p_{d,n}(x|z)p(z)}{p_{d,n}(x)}$$

- From the previous slide

$$p_{d,n}(x|z=0) = p_d(x) \qquad\qquad p(z=0) = \eta$$
$$p_{d,n}(x|z=1) = p_n(x) \qquad\qquad p(z=1) = 1-\eta$$

- Recall law of total probability

$$\begin{aligned} p_{d,n}(x) &= p_{d,n}(x|z=0)p(z=0) + p_{d,n}(x|z=1)p(z=1) \\ &= p_d(x)\eta + p_n(x)(1-\eta) \end{aligned}$$

Noise-contrastive estimation

- From previous slide

$$p_{d,n}(x) = \eta p_d(x) + (1-\eta)p_n(x)$$

- Apply Bayes

$$p_{d,n}(z=0|x) = \frac{p_{d,n}(x|z=0)p(z=0)}{p_{d,n}(x)} = \frac{\eta p_d(x)}{\eta p_d(x) + (1-\eta)p_n(x)}$$

$$p_{d,n}(z=1|x) = \frac{p_{d,n}(x|z=1)p(z=1)}{p_{d,n}(x)} = \frac{(1-\eta)p_n(x)}{\eta p_d(x) + (1-\eta)p_n(x)}$$

- We rewrite $\eta = 1/(1+\nu)$ (hence $1-\eta = \nu/(1+\nu)$) for a $\nu > 0$

$$p_{d,n}(z=0|x) = \frac{p_d(x)}{p_d(x) + \nu p_n(x)} \qquad p_{d,n}(z=1|x) = \frac{\nu p_n(x)}{p_d(x) + \nu p_n(x)}$$

- Can be unified into (observe that $z \in \{0,1\}$)

$$p_{d,n}(z|x) = \frac{(1-z)p_d(x) + \nu z p_n(x)}{p_d(x) + \nu p_n(x)}$$

Noise-contrastive estimation

- From previous slide

$$p_{d,n}(x) = \eta p_d(x) + (1-\eta)p_n(x)$$

- Apply Bayes

$$p_{d,n}(z=0|x) = \frac{p_{d,n}(x|z=0)p(z=0)}{p_{d,n}(x)} = \frac{\eta p_d(x)}{\eta p_d(x) + (1-\eta)p_n(x)}$$
$$p_{d,n}(z=1|x) = \frac{p_{d,n}(x|z=1)p(z=1)}{p_{d,n}(x)} = \frac{(1-\eta)p_n(x)}{\eta p_d(x) + (1-\eta)p_n(x)}$$

- We rewrite $\eta = 1/(1+\nu)$ (hence $1-\eta = \nu/(1+\nu)$) for a $\nu > 0$

$$p_{d,n}(z=0|x) = \frac{p_d(x)}{p_d(x) + \nu p_n(x)} \qquad p_{d,n}(z=1|x) = \frac{\nu p_n(x)}{p_d(x) + \nu p_n(x)}$$

- Can be unified into (observe that $z \in \{0,1\}$)

$$p_{d,n}(z|x) = \frac{(1-z)p_d(x) + \nu z p_n(x)}{p_d(x) + \nu p_n(x)}$$

Noise-contrastive estimation

- From previous slide

$$p_{d,n}(x) = \eta p_d(x) + (1-\eta)p_n(x)$$

- Apply Bayes

$$p_{d,n}(z=0|x) = \frac{p_{d,n}(x|z=0)p(z=0)}{p_{d,n}(x)} = \frac{\eta p_d(x)}{\eta p_d(x) + (1-\eta)p_n(x)}$$
$$p_{d,n}(z=1|x) = \frac{p_{d,n}(x|z=1)p(z=1)}{p_{d,n}(x)} = \frac{(1-\eta)p_n(x)}{\eta p_d(x) + (1-\eta)p_n(x)}$$

- We rewrite $\eta = 1/(1+\nu)$ (hence $1-\eta = \nu/(1+\nu)$) for a $\nu > 0$

$$p_{d,n}(z=0|x) = \frac{p_d(x)}{p_d(x) + \nu p_n(x)} \qquad p_{d,n}(z=1|x) = \frac{\nu p_n(x)}{p_d(x) + \nu p_n(x)}$$

- Can be unified into (observe that $z \in \{0,1\}$)

$$p_{d,n}(z|x) = \frac{(1-z)p_d(x) + \nu z p_n(x)}{p_d(x) + \nu p_n(x)}$$

Noise-contrastive estimation

- From previous slide

$$p_{d,n}(x) = \eta p_d(x) + (1-\eta)p_n(x)$$

- Apply Bayes

$$p_{d,n}(z=0|x) = \frac{p_{d,n}(x|z=0)p(z=0)}{p_{d,n}(x)} = \frac{\eta p_d(x)}{\eta p_d(x) + (1-\eta)p_n(x)}$$
$$p_{d,n}(z=1|x) = \frac{p_{d,n}(x|z=1)p(z=1)}{p_{d,n}(x)} = \frac{(1-\eta)p_n(x)}{\eta p_d(x) + (1-\eta)p_n(x)}$$

- We rewrite $\eta = 1/(1+\nu)$ (hence $1-\eta = \nu/(1+\nu)$) for a $\nu > 0$

$$p_{d,n}(z=0|x) = \frac{p_d(x)}{p_d(x)+\nu p_n(x)} \qquad p_{d,n}(z=1|x) = \frac{\nu p_n(x)}{p_d(x)+\nu p_n(x)}$$

- Can be unified into (observe that $z \in \{0,1\}$)

$$p_{d,n}(z|x) = \frac{(1-z)p_d(x) + \nu z p_n(x)}{p_d(x)+\nu p_n(x)}$$

Noise-contrastive estimation

- Posterior induced by true data distribution p_d

$$p_{d,n}(z|x) = \frac{(1-z)p_d(x) + \nu z p_n(x)}{p_d(x) + \nu p_n(x)}$$

- Posterior induced by model distribution p_θ

$$p_{\theta,n}(z|x) = \frac{(1-z)p_\theta(x) + \nu z p_n(x)}{p_\theta(x) + \nu p_n(x)}$$

  p_θ has our parameters of interest
- NCE: use logarithmic PSR to align $p_{\theta,n}$ with $p_{d,n}$
  - We need only samples from $p_{d,n}(x,z)$
  - Construction of training data $\{(x_i, z_i)\}$

$$z_i \sim p(z) \qquad\qquad x_i \sim \begin{cases} p_d(x) & \text{if } z = 0 \\ p_n(x) & \text{if } z = 1 \end{cases}$$

Noise-contrastive estimation

- Posterior induced by true data distribution p_d

$$p_{d,n}(z|x) = \frac{(1-z)p_d(x) + \nu z p_n(x)}{p_d(x) + \nu p_n(x)}$$

- Posterior induced by model distribution p_θ

$$p_{\theta,n}(z|x) = \frac{(1-z)p_\theta(x) + \nu z p_n(x)}{p_\theta(x) + \nu p_n(x)}$$

  p_θ has our parameters of interest

- NCE: use logarithmic PSR to align $p_{\theta,n}$ with $p_{d,n}$
  - We need only samples from $p_{d,n}(x, z)$
  - Construction of training data $\{(x_i, z_i)\}$

$$z_i \sim p(z) \qquad\qquad x_i \sim \begin{cases} p_d(x) & \text{if } z = 0 \\ p_n(x) & \text{if } z = 1 \end{cases}$$

Noise-contrastive estimation

- Posterior induced by true data distribution p_d

$$p_{d,n}(z|x) = \frac{(1-z)p_d(x) + \nu z p_n(x)}{p_d(x) + \nu p_n(x)}$$

- Posterior induced by model distribution p_θ

$$p_{\theta,n}(z|x) = \frac{(1-z)p_\theta(x) + \nu z p_n(x)}{p_\theta(x) + \nu p_n(x)}$$

  p_θ has our parameters of interest
- NCE: use logarithmic PSR to align $p_{\theta,n}$ with $p_{d,n}$
  - We need only samples from $p_{d,n}(x,z)$
  - Construction of training data $\{(x_i, z_i)\}$

$$z_i \sim p(z) \qquad\qquad x_i \sim \begin{cases} p_d(x) & \text{if } z = 0 \\ p_n(x) & \text{if } z = 1 \end{cases}$$

Noise-contrastive estimation

- Let S be any (strictly) PSR
- NCE objective by taking expectation w.r.t. $p_{d,n}$

$$
\begin{aligned}
J(\theta) &= \mathbb{E}_{(x,z)\sim p_{d,n}(x,z)}\left[S(p_{\theta,n}(z|x), z)\right] \\
&= \mathbb{E}_{(x,z)\sim p_{d,n}(x,z)}\left[S\left(\frac{(1-z)p_\theta(x)+\nu z p_n(x)}{p_\theta(x)+\nu p_n(x)}, z\right)\right] \\
&= \mathbb{E}_{z\sim p(z),x\sim p_{d,n}(x|z)}\left[S\left(\frac{(1-z)p_\theta(x)+\nu z p_n(x)}{p_\theta(x)+\nu p_n(x)}, z\right)\right] \\
&= \eta\mathbb{E}_{x\sim p_d}\left[S\left(\frac{p_\theta(x)}{p_\theta(x)+\nu p_n(x)}, 0\right)\right] + (1-\eta)\mathbb{E}_{x\sim p_n}\left[S\left(\frac{\nu p_n(x)}{p_\theta(x)+\nu p_n(x)}, 1\right)\right] \\
&\propto \mathbb{E}_{x\sim p_d}\left[S\left(\frac{p_\theta(x)}{p_\theta(x)+\nu p_n(x)}, 0\right)\right] + \nu\mathbb{E}_{x\sim p_n}\left[S\left(\frac{\nu p_n(x)}{p_\theta(x)+\nu p_n(x)}, 1\right)\right]
\end{aligned}
$$

Noise-contrastive estimation

- Let S be any (strictly) PSR
- NCE objective by taking expectation w.r.t. $p_{d,n}$

$$
\begin{aligned}
J(\theta) &= \mathbb{E}_{(x,z)\sim p_{d,n}(x,z)}\left[S(p_{\theta,n}(z|x), z)\right] \\
&= \mathbb{E}_{(x,z)\sim p_{d,n}(x,z)}\left[S\left(\frac{(1-z)p_\theta(x)+\nu z p_n(x)}{p_\theta(x)+\nu p_n(x)}, z\right)\right] \\
&= \mathbb{E}_{z\sim p(z), x\sim p_{d,n}(x|z)}\left[S\left(\frac{(1-z)p_\theta(x)+\nu z p_n(x)}{p_\theta(x)+\nu p_n(x)}, z\right)\right] \\
&= \eta\mathbb{E}_{x\sim p_d}\left[S\left(\frac{p_\theta(x)}{p_\theta(x)+\nu p_n(x)}, 0\right)\right] + (1-\eta)\mathbb{E}_{x\sim p_n}\left[S\left(\frac{\nu p_n(x)}{p_\theta(x)+\nu p_n(x)}, 1\right)\right] \\
&\propto \mathbb{E}_{x\sim p_d}\left[S\left(\frac{p_\theta(x)}{p_\theta(x)+\nu p_n(x)}, 0\right)\right] + \nu\mathbb{E}_{x\sim p_n}\left[S\left(\frac{\nu p_n(x)}{p_\theta(x)+\nu p_n(x)}, 1\right)\right]
\end{aligned}
$$

- Let S be any (strictly) PSR
- NCE objective by taking expectation w.r.t. $p_{d,n}$

$$
\begin{aligned}
J(\theta) &= \mathbb{E}_{(x,z)\sim p_{d,n}(x,z)}\left[S(p_{\theta,n}(z|x), z)\right] \\
&= \mathbb{E}_{(x,z)\sim p_{d,n}(x,z)}\left[S\left(\frac{(1-z)p_\theta(x)+\nu z p_n(x)}{p_\theta(x)+\nu p_n(x)}, z\right)\right] \\
&= \mathbb{E}_{z\sim p(z), x\sim p_{d,n}(x|z)}\left[S\left(\frac{(1-z)p_\theta(x)+\nu z p_n(x)}{p_\theta(x)+\nu p_n(x)}, z\right)\right] \\
&= \eta\mathbb{E}_{x\sim p_d}\left[S\left(\frac{p_\theta(x)}{p_\theta(x)+\nu p_n(x)}, 0\right)\right] + (1-\eta)\mathbb{E}_{x\sim p_n}\left[S\left(\frac{\nu p_n(x)}{p_\theta(x)+\nu p_n(x)}, 1\right)\right] \\
&\propto \mathbb{E}_{x\sim p_d}\left[S\left(\frac{p_\theta(x)}{p_\theta(x)+\nu p_n(x)}, 0\right)\right] + \nu\mathbb{E}_{x\sim p_n}\left[S\left(\frac{\nu p_n(x)}{p_\theta(x)+\nu p_n(x)}, 1\right)\right]
\end{aligned}
$$

Noise-contrastive estimation

- Let S be any (strictly) PSR
- NCE objective by taking expectation w.r.t. $p_{d,n}$

$$
\begin{aligned}
J(\theta) &= \mathbb{E}_{(x,z)\sim p_{d,n}(x,z)}\left[S(p_{\theta,n}(z|x), z)\right] \\
&= \mathbb{E}_{(x,z)\sim p_{d,n}(x,z)}\left[S\left(\frac{(1-z)p_\theta(x)+\nu z p_n(x)}{p_\theta(x)+\nu p_n(x)}, z\right)\right] \\
&= \mathbb{E}_{z\sim p(z), x\sim p_{d,n}(x|z)}\left[S\left(\frac{(1-z)p_\theta(x)+\nu z p_n(x)}{p_\theta(x)+\nu p_n(x)}, z\right)\right] \\
&= \eta\mathbb{E}_{x\sim p_d}\left[S\left(\frac{p_\theta(x)}{p_\theta(x)+\nu p_n(x)}, 0\right)\right] + (1-\eta)\mathbb{E}_{x\sim p_n}\left[S\left(\frac{\nu p_n(x)}{p_\theta(x)+\nu p_n(x)}, 1\right)\right] \\
&\propto \mathbb{E}_{x\sim p_d}\left[S\left(\frac{p_\theta(x)}{p_\theta(x)+\nu p_n(x)}, 0\right)\right] + \nu\mathbb{E}_{x\sim p_n}\left[S\left(\frac{\nu p_n(x)}{p_\theta(x)+\nu p_n(x)}, 1\right)\right]
\end{aligned}
$$

Noise-contrastive estimation

- Let S be any (strictly) PSR
- NCE objective by taking expectation w.r.t. $p_{d,n}$

$$
\begin{aligned}
J(\theta) &= \mathbb{E}_{(x,z)\sim p_{d,n}(x,z)}\left[S(p_{\theta,n}(z|x), z)\right] \\
&= \mathbb{E}_{(x,z)\sim p_{d,n}(x,z)}\left[S\left(\frac{(1-z)p_\theta(x)+\nu z p_n(x)}{p_\theta(x)+\nu p_n(x)}, z\right)\right] \\
&= \mathbb{E}_{z\sim p(z), x\sim p_{d,n}(x|z)}\left[S\left(\frac{(1-z)p_\theta(x)+\nu z p_n(x)}{p_\theta(x)+\nu p_n(x)}, z\right)\right] \\
&= \eta\mathbb{E}_{x\sim p_d}\left[S\left(\frac{p_\theta(x)}{p_\theta(x)+\nu p_n(x)}, 0\right)\right] + (1-\eta)\mathbb{E}_{x\sim p_n}\left[S\left(\frac{\nu p_n(x)}{p_\theta(x)+\nu p_n(x)}, 1\right)\right] \\
&\propto \mathbb{E}_{x\sim p_d}\left[S\left(\frac{p_\theta(x)}{p_\theta(x)+\nu p_n(x)}, 0\right)\right] + \nu\mathbb{E}_{x\sim p_n}\left[S\left(\frac{\nu p_n(x)}{p_\theta(x)+\nu p_n(x)}, 1\right)\right]
\end{aligned}
$$

Noise-contrastive estimation

- Choose $S(q, z) = \log q(z)$

$$J(\theta) \propto \mathbb{E}_{x \sim p_d}\left[\log \frac{p_\theta(x)}{p_\theta(x) + \nu p_n(x)}\right] + \nu \mathbb{E}_{x \sim p_n}\left[\log \frac{\nu p_n(x)}{p_\theta(x) + \nu p_n(x)}\right]$$

$$\approx \frac{1}{N}\sum_{i=1}^{N} \log \frac{p_\theta(x_i)}{p_\theta(x_i) + \nu p_n(x_i)} + \frac{1}{N}\sum_{i=1}^{\nu N} \log \frac{\nu p_n(x_i')}{p_\theta(x_i') + \nu p_n(x_i')}$$

with $x_i \sim p_d$ and $x_i' \sim p_n$

- This is noise-contrastive estimation
- Learning distribution parameters by binary classification
  - Classifier distinguishes between real data and noise samples
  - *Superficially* similar to GANs
- In practice p_n should be as close to p_d as possible
  - Theory only requires that p_d and p_n overlap

$$p_d(x) > 0 \implies p_n(x) > 0$$

Noise-contrastive estimation

- Choose $S(q, z) = \log q(z)$

$$J(\theta) \propto \mathbb{E}_{x\sim p_d}\left[\log \frac{p_\theta(x)}{p_\theta(x) + \nu p_n(x)}\right] + \nu \mathbb{E}_{x\sim p_n}\left[\log \frac{\nu p_n(x)}{p_\theta(x) + \nu p_n(x)}\right]$$
$$\approx \frac{1}{N}\sum_{i=1}^{N} \log \frac{p_\theta(x_i)}{p_\theta(x_i) + \nu p_n(x_i)} + \frac{1}{N}\sum_{i=1}^{\nu N} \log \frac{\nu p_n(x'_i)}{p_\theta(x'_i) + \nu p_n(x'_i)}$$

  with $x_i \sim p_d$ and $x'_i \sim p_n$
- This is noise-contrastive estimation
- Learning distribution parameters by binary classification
  - Classifier distinguishes between real data and noise samples
  - *Superficially* similar to GANs
- In practice p_n should be as close to p_d as possible
  - Theory only requires that p_d and p_n overlap

$$p_d(x) > 0 \implies p_n(x) > 0$$

Noise-contrastive estimation

- Choose $S(q,z) = \log q(z)$

$$J(\theta) \propto \mathbb{E}_{x\sim p_d}\left[\log \frac{p_\theta(x)}{p_\theta(x)+\nu p_n(x)}\right] + \nu \mathbb{E}_{x\sim p_n}\left[\log \frac{\nu p_n(x)}{p_\theta(x)+\nu p_n(x)}\right]$$
$$\approx \frac{1}{N}\sum_{i=1}^{N} \log \frac{p_\theta(x_i)}{p_\theta(x_i)+\nu p_n(x_i)} + \frac{1}{N}\sum_{i=1}^{\nu N} \log \frac{\nu p_n(x'_i)}{p_\theta(x'_i)+\nu p_n(x'_i)}$$

  with $x_i \sim p_d$ and $x'_i \sim p_n$
- This is noise-contrastive estimation
- Learning distribution parameters by binary classification
  - Classifier distinguishes between real data and noise samples
  - *Superficially* similar to GANs
- In practice p_n should be as close to p_d as possible
  - Theory only requires that p_d and p_n overlap

$$p_d(x) > 0 \implies p_n(x) > 0$$

Noise-contrastive estimation

- What about unnormalized models?

$$p_\theta(x;\theta) = \frac{1}{Z(\theta)} p_\theta^0(x;\theta)$$

$Z(\theta) = \int p_\theta^0(x;\theta)\,dx$ is hard to compute

- Solution: add Z to the set of parameters!
  - In practice add $c = \log Z$
  - Numerical stability
- Strictly PSR makes sure that

$$c \overset{N\to\infty}{\to} \log \int p_\theta^0(x;\theta^*)\,dx \qquad \text{a.s.}$$

Discussion

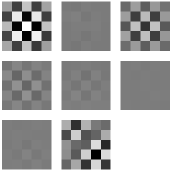
What are pros and cons of NCE?

Caveats

- Curse of dimensionality
  - In high dimensions you need *many* noise samples to carve out p_θ
- Interpolation regime
  - Finite sets $\{x_i\}$ and $\{x'_i\}$ and overparametrized p_θ
  - NCE loss can approach zero, θ unbounded
- Be careful to model p_θ right!
  - You want to model properties of p_d not of p_n
  - It is easy to have a "good" solution if p_θ detects features in (simple) noise

$$\log p_\theta(x) = \sum_k \log\left(1 + e^{w_k^T x}\right) \qquad \text{okay}$$

$$\log p_\theta(x) = -\sum_k \log\left(1 + e^{w_k^T x}\right) \qquad \text{bad}$$



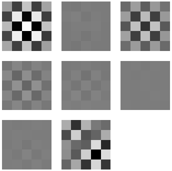
Noise-contrastive estimation

Caveats

- Curse of dimensionality
  - In high dimensions you need *many* noise samples to carve out p_θ
- Interpolation regime
  - Finite sets $\{x_i\}$ and $\{x_i'\}$ and overparametrized p_θ
  - NCE loss can approach zero, θ unbounded
- Be careful to model p_θ right!
  - You want to model properties of p_d not of p_n
  - It is easy to have a "good" solution if p_θ detects features in (simple) noise

$$\log p_\theta(x) = \sum_k \log\left(1 + e^{w_k^T x}\right) \qquad \text{okay}$$

$$\log p_\theta(x) = -\sum_k \log\left(1 + e^{w_k^T x}\right) \qquad \text{bad}$$



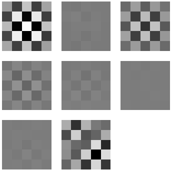
Noise-contrastive estimation

Caveats

- Curse of dimensionality
  - In high dimensions you need *many* noise samples to carve out p_θ
- Interpolation regime
  - Finite sets $\{x_i\}$ and $\{x_i'\}$ and overparametrized p_θ
  - NCE loss can approach zero, θ unbounded
- Be careful to model p_θ right!
  - You want to model properties of p_d not of p_n
  - It is easy to have a "good" solution if p_θ detects features in (simple) noise

$$\log p_\theta(x) = \sum\nolimits_k \log\left(1 + e^{w_k^T x}\right) \qquad \text{okay}$$

$$\log p_\theta(x) = -\sum\nolimits_k \log\left(1 + e^{w_k^T x}\right) \qquad \text{bad}$$



Conditional NCE

Overview

Conditional NCE (cNCE) is a variant of NCE that

- perturbs the real training data instead of using a separate noise distribution
- therefore distinguishes between real data and nearby noisy samples
- is not able to estimate the partition function

- Ceylan & Gutmann, "Conditional Noise-Contrastive Estimation of Unnormalised Models"

Conditional NCE

- As with NCE, we convert the problem by matching suitable posteriors
- Let $p_n(x|x')$ be a noise distribution conditioned (dependend) on x', e.g.

$$p_n(x|x') = \mathcal{N}(x; x', \sigma^2 \mathrm{I})$$

- For given $z \in \{0, 1\}$ model the conditional probabilities

$$p_d(x, x'|z) = \begin{cases} p_d(x)p_n(x'|x) & \text{if } z = 0 \\ p_d(x')p_n(x|x') & \text{if } z = 1 \end{cases}$$

$$p_\theta(x, x'|z) = \begin{cases} p_\theta(x)p_n(x'|x) & \text{if } z = 0 \\ p_\theta(x')p_n(x|x') & \text{if } z = 1 \end{cases}$$

- $z \in \{0, 1\}$ determines whether the 1st or the 2nd element in a pair is the clean data
  - Clean data comes from p_d or p_θ

Conditional NCE

- As with NCE, we convert the problem by matching suitable posteriors
- Let $p_n(x|x')$ be a noise distribution conditioned (dependend) on x', e.g.

$$p_n(x|x') = \mathcal{N}(x; x', \sigma^2 \mathrm{I})$$

- For given $z \in \{0, 1\}$ model the conditional probabilities

$$p_d(x, x'|z) = \begin{cases} p_d(x)p_n(x'|x) & \text{if } z = 0 \\ p_d(x')p_n(x|x') & \text{if } z = 1 \end{cases}$$

$$p_\theta(x, x'|z) = \begin{cases} p_\theta(x)p_n(x'|x) & \text{if } z = 0 \\ p_\theta(x')p_n(x|x') & \text{if } z = 1 \end{cases}$$

- $z \in \{0, 1\}$ determines whether the 1st or the 2nd element in a pair is the clean data
    - Clean data comes from p_d or p_θ

Conditional NCE

- Recall

$$p_\theta(x,x'|z) = \begin{cases} p_\theta(x)p_n(x'|x) & \text{if } z=0 \\ p_\theta(x')p_n(x|x') & \text{if } z=1 \end{cases}$$

- W.l.o.g. we can assume $p(z=0) = p(z=1) = 1/2$
- The posterior $p_\theta(z|x,x')$ is given by

$$p_\theta(z=0|x,x') = \frac{p_\theta(x,x'|z=0)p(z=0)}{p_\theta(x,x'|z=0)p(z=0) + p_\theta(x,x'|z=1)p(z=1)}$$
$$= \frac{p_\theta(x)p_n(x'|x)}{p_\theta(x)p_n(x'|x) + p_\theta(x')p_n(x|x')}$$
$$p_\theta(z=1|x,x') = \frac{p_\theta(x')p_n(x|x')}{p_\theta(x)p_n(x'|x) + p_\theta(x')p_n(x|x')}$$

- In this formulation we cannot estimate the partition function as it cancels
  - The posterior is invariant to scaling of p_θ

Conditional NCE

- Recall

$$p_\theta(x, x'|z) = \begin{cases} p_\theta(x) p_n(x'|x) & \text{if } z = 0 \\ p_\theta(x') p_n(x|x') & \text{if } z = 1 \end{cases}$$

- W.l.o.g. we can assume $p(z = 0) = p(z = 1) = 1/2$
- The posterior $p_\theta(z|x, x')$ is given by

$$\begin{aligned} p_\theta(z = 0|x, x') &= \frac{p_\theta(x, x'|z = 0) p(z = 0)}{p_\theta(x, x'|z = 0) p(z = 0) + p_\theta(x, x'|z = 1) p(z = 1)} \\ &= \frac{p_\theta(x) p_n(x'|x)}{p_\theta(x) p_n(x'|x) + p_\theta(x') p_n(x|x')} \\ p_\theta(z = 1|x, x') &= \frac{p_\theta(x') p_n(x|x')}{p_\theta(x) p_n(x'|x) + p_\theta(x') p_n(x|x')} \end{aligned}$$

- In this formulation we cannot estimate the partition function as it cancels
  - The posterior is invariant to scaling of p_θ

Conditional NCE

- Recall

$$p_\theta(x, x'|z) = \begin{cases} p_\theta(x)p_n(x'|x) & \text{if } z = 0 \\ p_\theta(x')p_n(x|x') & \text{if } z = 1 \end{cases}$$

- W.l.o.g. we can assume $p(z = 0) = p(z = 1) = 1/2$
- The posterior $p_\theta(z|x, x')$ is given by

$$\begin{aligned} p_\theta(z = 0|x, x') &= \frac{p_\theta(x, x'|z = 0)p(z = 0)}{p_\theta(x, x'|z = 0)p(z = 0) + p_\theta(x, x'|z = 1)p(z = 1)} \\ &= \frac{p_\theta(x)p_n(x'|x)}{p_\theta(x)p_n(x'|x) + p_\theta(x')p_n(x|x')} \\ p_\theta(z = 1|x, x') &= \frac{p_\theta(x')p_n(x|x')}{p_\theta(x)p_n(x'|x) + p_\theta(x')p_n(x|x')} \end{aligned}$$

- In this formulation we cannot estimate the partition function as it cancels
  - The posterior is invariant to scaling of p_θ

Conditional NCE

- Let S be a PSR

$$J(\theta) = \mathbb{E}_{(x,x',z)\sim p_d(x,x',z)}\left[S\left(p_\theta(z|x,x'), z\right)\right]$$
$$= \tfrac{1}{2}\mathbb{E}_{x\sim p_d(x),x'\sim p_n(x'|x)}\left[S\left(\frac{p_\theta(x)p_n(x'|x)}{p_\theta(x)p_n(x'|x)+p_\theta(x')p_n(x|x')}, 0\right)\right]$$
$$+ \tfrac{1}{2}\mathbb{E}_{x'\sim p_d(x),x\sim p_n(x|x')}\left[S\left(\frac{p_\theta(x')p_n(x|x')}{p_\theta(x)p_n(x'|x)+p_\theta(x')p_n(x|x')}, 1\right)\right]$$

- Assume $p_n(x|x') = p_n(x'|x)$

$$J(\theta) = \tfrac{1}{2}\mathbb{E}_{x\sim p_d(x),x'\sim p_n(x'|x)}\left[S\left(\frac{p_\theta(x)}{p_\theta(x)+p_\theta(x')}, 0\right)\right]$$
$$+ \tfrac{1}{2}\mathbb{E}_{x'\sim p_d(x),x\sim p_n(x|x')}\left[S\left(\frac{p_\theta(x')}{p_\theta(x)+p_\theta(x')}, 1\right)\right]$$

Conditional NCE

- Let S be a PSR

$$\begin{aligned} J(\theta) &= \mathbb{E}_{(x,x',z)\sim p_d(x,x',z)}\left[S\left(p_\theta(z|x,x'),z\right)\right] \\ &= \tfrac{1}{2}\mathbb{E}_{x\sim p_d(x),x'\sim p_n(x'|x)}\left[S\left(\frac{p_\theta(x)p_n(x'|x)}{p_\theta(x)p_n(x'|x)+p_\theta(x')p_n(x|x')},0\right)\right] \\ &+ \tfrac{1}{2}\mathbb{E}_{x'\sim p_d(x),x\sim p_n(x|x')}\left[S\left(\frac{p_\theta(x')p_n(x|x')}{p_\theta(x)p_n(x'|x)+p_\theta(x')p_n(x|x')},1\right)\right] \end{aligned}$$

- Assume $p_n(x|x') = p_n(x'|x)$

$$\begin{aligned} J(\theta) &= \tfrac{1}{2}\mathbb{E}_{x\sim p_d(x),x'\sim p_n(x'|x)}\left[S\left(\frac{p_\theta(x)}{p_\theta(x)+p_\theta(x')},0\right)\right] \\ &+ \tfrac{1}{2}\mathbb{E}_{x'\sim p_d(x),x\sim p_n(x|x')}\left[S\left(\frac{p_\theta(x')}{p_\theta(x)+p_\theta(x')},1\right)\right] \end{aligned}$$

Conditional NCE

- Let S be a PSR

$$\begin{aligned}
J(\theta) &= \mathbb{E}_{(x,x',z)\sim p_d(x,x',z)}\left[S\left(p_\theta(z|x,x'),z\right)\right] \\
&= \tfrac{1}{2}\mathbb{E}_{x\sim p_d(x),x'\sim p_n(x'|x)}\left[S\left(\frac{p_\theta(x)p_n(x'|x)}{p_\theta(x)p_n(x'|x)+p_\theta(x')p_n(x|x')},0\right)\right] \\
&+ \tfrac{1}{2}\mathbb{E}_{x'\sim p_d(x),x\sim p_n(x|x')}\left[S\left(\frac{p_\theta(x')p_n(x|x')}{p_\theta(x)p_n(x'|x)+p_\theta(x')p_n(x|x')},1\right)\right]
\end{aligned}$$

- Assume $p_n(x|x') = p_n(x'|x)$

$$\begin{aligned}
J(\theta) = \tfrac{1}{2}\mathbb{E}_{x\sim p_d(x),x'\sim p_n(x'|x)}&\left[S\left(\frac{p_\theta(x)}{p_\theta(x)+p_\theta(x')},0\right)\right] \\
+ \tfrac{1}{2}\mathbb{E}_{x'\sim p_d(x),x\sim p_n(x|x')}&\left[S\left(\frac{p_\theta(x')}{p_\theta(x)+p_\theta(x')},1\right)\right]
\end{aligned}$$

- Let S be the logarithmic PSR

$$
\begin{aligned}
J(\theta) &= \tfrac{1}{2}\mathbb{E}_{x\sim p_d(x),x'\sim p_n(x'|x)}\left[\log\left(\frac{p_\theta(x)}{p_\theta(x)+p_\theta(x')}\right)\right] \\
&\quad + \tfrac{1}{2}\mathbb{E}_{x'\sim p_d(x),x\sim p_n(x|x')}\left[\log\left(\frac{p_\theta(x')}{p_\theta(x)+p_\theta(x')}\right)\right]
\end{aligned}
$$

- The two terms are the same (renaming x and x')

$$
\begin{aligned}
J(\theta) &= \mathbb{E}_{x\sim p_d(x),x'\sim p_n(x'|x)}\left[\log\left(\frac{p_\theta(x)}{p_\theta(x)+p_\theta(x')}\right)\right] \\
&= \mathbb{E}_{x\sim p_d(x),x'\sim p_n(x'|x)}\left[\log p_\theta(x) - \log\left(p_\theta(x)+p_\theta(x')\right)\right]
\end{aligned}
$$

- Only conceptual swapping of the elements (x, x')
- Relies on symmetry of S: $S(q(0),0) = S(q(1),1)$
  - One can design asymmetric PSRs

Conditional NCE

- Let S be the logarithmic PSR

$$J(\theta) = \tfrac{1}{2}\mathbb{E}_{x\sim p_d(x), x'\sim p_n(x'|x)}\left[\log\left(\frac{p_\theta(x)}{p_\theta(x)+p_\theta(x')}\right)\right] + \tfrac{1}{2}\mathbb{E}_{x'\sim p_d(x), x\sim p_n(x|x')}\left[\log\left(\frac{p_\theta(x')}{p_\theta(x)+p_\theta(x')}\right)\right]$$

- The two terms are the same (renaming x and x')

$$\begin{aligned} J(\theta) &= \mathbb{E}_{x\sim p_d(x), x'\sim p_n(x'|x)}\left[\log\left(\frac{p_\theta(x)}{p_\theta(x)+p_\theta(x')}\right)\right] \\ &= \mathbb{E}_{x\sim p_d(x), x'\sim p_n(x'|x)}\left[\log p_\theta(x) - \log\left(p_\theta(x)+p_\theta(x')\right)\right] \end{aligned}$$

- Only conceptual swapping of the elements (x, x')
- Relies on symmetry of S: $S(q(0), 0) = S(q(1), 1)$
  - One can design asymmetric PSRs

Conditional NCE

- Let S be the logarithmic PSR

$$J(\theta) = \tfrac{1}{2}\mathbb{E}_{x\sim p_d(x),x'\sim p_n(x'|x)}\left[\log\left(\frac{p_\theta(x)}{p_\theta(x)+p_\theta(x')}\right)\right] + \tfrac{1}{2}\mathbb{E}_{x'\sim p_d(x),x\sim p_n(x|x')}\left[\log\left(\frac{p_\theta(x')}{p_\theta(x)+p_\theta(x')}\right)\right]$$

- The two terms are the same (renaming x and x')

$$\begin{aligned} J(\theta) &= \mathbb{E}_{x\sim p_d(x),x'\sim p_n(x'|x)}\left[\log\left(\frac{p_\theta(x)}{p_\theta(x)+p_\theta(x')}\right)\right] \\ &= \mathbb{E}_{x\sim p_d(x),x'\sim p_n(x'|x)}\left[\log p_\theta(x) - \log\left(p_\theta(x)+p_\theta(x')\right)\right] \end{aligned}$$

- Only conceptual swapping of the elements (x, x')
- Relies on symmetry of S: $S(q(0),0) = S(q(1),1)$
  - One can design asymmetric PSRs

Conditional NCE

- We are given N data samples $\{x_i\}$ and generate $x'_i \sim p_n(x_i)$
- The sample version of the cNCE objective is

$$J(\theta) = \frac{1}{N} \sum_i \left[\log p_\theta(x_i) - \log \left(p_\theta(x_i) + p_\theta(x'_i) \right) \right]$$

- When is J large?
  - If $p_\theta(x) \gg p_\theta(x')$ where x is real (clean) data and x' is a perturbed sample
- Conditional NCE aims for p_θ ($\log p_\theta$) to be a local maximum at real samples x_i
  - It shares this property with score matching discussed next

Conditional NCE

- We are given N data samples $\{x_i\}$ and generate $x_i' \sim p_n(x_i)$
- The sample version of the cNCE objective is

$$J(\theta) = \frac{1}{N} \sum_i \left[\log p_\theta(x_i) - \log \left(p_\theta(x_i) + p_\theta(x_i') \right) \right]$$

- When is J large?
  - If $p_\theta(x) \gg p_\theta(x')$ where x is real (clean) data and x' is a perturbed sample
- Conditional NCE aims for p_θ ($\log p_\theta$) to be a local maximum at real samples x_i
  - It shares this property with score matching discussed next

Outline

Score matching

Overview

Score matching

- fits model parameters such that training samples are preferably local maxima of $\log p_\theta$
- works with unnormalized models, but does not estimate Z
- is a non-trivial instance of a local PSR
- works only for continuous data

- A. Hyvärinen, "Estimation of Non-Normalized Statistical Models by Score Matching"

Score matching

- Origin of the name
  - $s(\theta) = \nabla_\theta f(\theta)$ is called the "score" in statistics
  - Score used here: $\nabla_x \log p(x)$
- Choice of strictly convex F

$$F(q) = \frac{1}{2}\mathbb{E}_{x\sim q}\left[\,\|\nabla_x \log q(x)\|^2\,\right] = \frac{1}{2}\int q(x)\,\|\nabla_x \log q(x)\|^2\,dx$$
$$= \frac{1}{2}\int \frac{\|\nabla_x q(x)\|^2}{q(x)}\,dx$$

- Why is F convex?
- What is $S(q,x)$?

Score matching

- Origin of the name
  - $s(\theta) = \nabla_\theta f(\theta)$ is called the “score” in statistics
  - Score used here: $\nabla_x \log p(x)$
- Choice of strictly convex F

$$
\begin{aligned}
F(q) &= \frac{1}{2}\mathbb{E}_{x\sim q}\left[\, \|\nabla_x \log q(x)\|^2 \,\right] = \frac{1}{2}\int q(x)\, \|\nabla_x \log q(x)\|^2 \, dx \\
&= \frac{1}{2}\int \frac{\|\nabla_x q(x)\|^2}{q(x)}\, dx
\end{aligned}
$$

- Why is F convex?
- What is $S(q, x)$?

Score matching

- Why is F convex?

$$F(q) = \frac{1}{2} \int \frac{\|\nabla_x q(x)\|^2}{q(x)} \, dx$$

- ∇_x is linear operator
- We have quadratic-over-linear terms

$$\frac{\|Ax\|^2}{b^T x}$$

- Convex when $b^T x > 0$!
- Extension to $\{x : b^T x \geq 0\}$ possible
  - Yields constraint $Ax = 0$ whenever $b^T x = 0$

Score matching

- We recall

$$S(q,x) = F(q) + \frac{\partial F(q)}{\partial q(x)} - \int \frac{\partial F(q)}{\partial q(x')} q(x')\, dx'$$

- Main problem: what is

$$\frac{\partial F(q)}{\partial q(x)} = \frac{1}{2} \frac{\partial}{\partial q(x)} \left(\int q(x') \left\| \nabla_x \log q(x') \right\|^2 \, dx' \right)$$

- Solution 1: calculus of variations
- Solution 2: we guess S and recover $F(q) = S(q,q)$

Score matching

- Use infinite dimensional Hilbert spaces: A is a linear operator

$$\langle f, g\rangle = \int f(x)g(x)\,dx \qquad \langle Af, g\rangle = \langle f, A^T g\rangle$$

- $A = \nabla$: integration by parts (related to divergence thm and Green's identities)

$$\int_{-\infty}^{\infty} f' g\,dx = f(x)g(x)\big|_{-\infty}^{\infty} - \int_{-\infty}^{\infty} fg'\,dx$$

Assume $f(x) \to 0$ and $g(x) \to 0$ for $\|x\| \to \infty$

$$\int_{-\infty}^{\infty} f' g\,dx = -\int_{-\infty}^{\infty} fg'\,dx$$

- Higher dimensions (sum over dimensions)

$$\int \operatorname{div}(f) g\,dx = \langle \operatorname{div}(f), g\rangle = -\int f^T \nabla g\,dx = -\langle f, \nabla g\rangle \implies \nabla^T = -\operatorname{div}$$

- Δ is the Laplace operator

$$\Delta f(x) = \operatorname{div} \nabla_x f(x) = -\nabla_x^T \nabla_x f(x) = \sum_{j=1}^{d} \frac{\partial^2 f(x)}{\partial x_j^2}$$

Score matching

- Use infinite dimensional Hilbert spaces: A is a linear operator

$$\langle f, g \rangle = \int f(x) g(x)\, dx \qquad \langle Af, g \rangle = \langle f, A^T g \rangle$$

- $A = \nabla$: integration by parts (related to divergence thm and Green's identities)

$$\int_{-\infty}^{\infty} f' g\, dx = f(x)g(x)\Big|_{-\infty}^{\infty} - \int_{-\infty}^{\infty} f g'\, dx$$

Assume $f(x) \to 0$ and $g(x) \to 0$ for $\|x\| \to \infty$

$$\int_{-\infty}^{\infty} f' g\, dx = -\int_{-\infty}^{\infty} f g'\, dx$$

- Higher dimensions (sum over dimensions)

$$\int \mathrm{div}(f) g\, dx = \langle \mathrm{div}(f), g \rangle = -\int f^T \nabla g\, dx = -\langle f, \nabla g \rangle \implies \nabla^T = -\mathrm{div}$$

- Δ is the Laplace operator

$$\Delta f(x) = \mathrm{div} \nabla_x f(x) = -\nabla_x^T \nabla_x f(x) = \sum_{j=1}^{d} \frac{\partial^2 f(x)}{\partial x_j^2}$$

Score matching

- Use infinite dimensional Hilbert spaces: A is a linear operator

$$\langle f, g\rangle = \int f(x)g(x)\,dx \qquad\qquad \langle Af, g\rangle = \langle f, A^T g\rangle$$

- $A = \nabla$: integration by parts (related to divergence thm and Green's identities)

$$\int_{-\infty}^{\infty} f' g\,dx = f(x)g(x)\big|_{-\infty}^{\infty} - \int_{-\infty}^{\infty} fg'\,dx$$

  Assume $f(x) \to 0$ and $g(x) \to 0$ for $\|x\| \to \infty$

$$\int_{-\infty}^{\infty} f' g\,dx = -\int_{-\infty}^{\infty} fg'\,dx$$

- Higher dimensions (sum over dimensions)

$$\int \mathrm{div}(f) g\,dx = \langle \mathrm{div}(f), g\rangle = -\int f^T \nabla g\,dx = -\langle f, \nabla g\rangle \implies \nabla^T = -\mathrm{div}$$

- Δ is the Laplace operator

$$\Delta f(x) = \mathrm{div}\nabla_x f(x) = -\nabla_x^T \nabla_x f(x) = \sum_{j=1}^{d} \frac{\partial^2 f(x)}{\partial x_j^2}$$

Score matching

- Our guess for S

$$S(q,x) = -\frac{1}{2}\left\|\nabla_x \log q(x)\right\|^2 - \Delta \log q(x)$$

- We calculate

$$\begin{aligned}
S(q,q) = \mathbb{E}_{x\sim q}\left[S(q,x)\right] &= \int q(x) S(q,x)\,dx \\
&= -\int q(x)\left(\frac{1}{2}\left\|\nabla_x \log q(x)\right\|^2 + \Delta \log q(x)\right)dx \\
&= -\int \left(\frac{1}{2}\frac{\left\|\nabla_x q(x)\right\|^2}{q(x)} - q(x)\nabla_x^T \nabla_x \log q(x)\right)dx \\
&= -\frac{1}{2}\left\langle \frac{\nabla_x q}{q}, \nabla_x q\right\rangle + \left\langle q, \nabla_x^T \nabla_x \log q\right\rangle \\
&= -\frac{1}{2}\left\langle \frac{\nabla_x q}{q}, \nabla_x q\right\rangle + \left\langle \nabla_x q, \frac{\nabla_x q}{q}\right\rangle = \frac{1}{2}\left\langle \nabla_x q, \frac{\nabla_x q}{q}\right\rangle = \frac{1}{2}\int \frac{\left\|\nabla_x q(x)\right\|^2}{q(x)}dx
\end{aligned}$$

- Our guess for S

$$S(q,x) = -\frac{1}{2}\|\nabla_x \log q(x)\|^2 - \Delta \log q(x)$$

- We calculate

$$\begin{aligned}
S(q,q) &= \mathbb{E}_{x\sim q}\left[S(q,x)\right] = \int q(x) S(q,x)\, dx \\
&= -\int q(x) \left(\frac{1}{2}\|\nabla_x \log q(x)\|^2 + \Delta \log q(x)\right) dx \\
&= -\int \left(\frac{1}{2}\frac{\|\nabla_x q(x)\|^2}{q(x)} - q(x)\nabla_x^T \nabla_x \log q(x)\right) dx \\
&= -\frac{1}{2}\left\langle \frac{\nabla_x q}{q}, \nabla_x q\right\rangle + \langle q, \nabla_x^T \nabla_x \log q\rangle \\
&= -\frac{1}{2}\left\langle \frac{\nabla_x q}{q}, \nabla_x q\right\rangle + \left\langle \nabla_x q, \frac{\nabla_x q}{q}\right\rangle = \frac{1}{2}\left\langle \nabla_x q, \frac{\nabla_x q}{q}\right\rangle = \frac{1}{2}\int \frac{\|\nabla_x q(x)\|^2}{q(x)} dx
\end{aligned}$$

Score matching

- Our guess for S

$$S(q,x) = -\frac{1}{2}\left\|\nabla_x \log q(x)\right\|^2 - \Delta \log q(x)$$

- We calculate

$$\begin{aligned}
S(q,q) &= \mathbb{E}_{x\sim q}\left[S(q,x)\right] = \int q(x) S(q,x)\, dx \\
&= -\int q(x) \left(\frac{1}{2}\left\|\nabla_x \log q(x)\right\|^2 + \Delta \log q(x)\right) dx \\
&= -\int \left(\frac{1}{2}\frac{\left\|\nabla_x q(x)\right\|^2}{q(x)} - q(x)\nabla_x^T \nabla_x \log q(x)\right) dx \\
&= -\frac{1}{2}\left\langle \frac{\nabla_x q}{q}, \nabla_x q \right\rangle + \left\langle q, \nabla_x^T \nabla_x \log q \right\rangle \\
&= -\frac{1}{2}\left\langle \frac{\nabla_x q}{q}, \nabla_x q \right\rangle + \left\langle \nabla_x q, \frac{\nabla_x q}{q} \right\rangle = \frac{1}{2}\left\langle \nabla_x q, \frac{\nabla_x q}{q} \right\rangle = \frac{1}{2}\int \frac{\left\|\nabla_x q(x)\right\|^2}{q(x)} dx
\end{aligned}$$

Score matching

- Our guess for S

$$S(q,x) = -\frac{1}{2}\left\|\nabla_x \log q(x)\right\|^2 - \Delta \log q(x)$$

- We calculate

$$\begin{aligned}
S(q,q) &= \mathbb{E}_{x\sim q}\left[S(q,x)\right] = \int q(x) S(q,x)\, dx \\
&= -\int q(x) \left(\frac{1}{2}\left\|\nabla_x \log q(x)\right\|^2 + \Delta \log q(x)\right) dx \\
&= -\int \left(\frac{1}{2}\frac{\left\|\nabla_x q(x)\right\|^2}{q(x)} - q(x)\nabla_x^T \nabla_x \log q(x)\right) dx \\
&= -\frac{1}{2}\left\langle \frac{\nabla_x q}{q}, \nabla_x q\right\rangle + \left\langle q, \nabla_x^T \nabla_x \log q\right\rangle \\
&= -\frac{1}{2}\left\langle \frac{\nabla_x q}{q}, \nabla_x q\right\rangle + \left\langle \nabla_x q, \frac{\nabla_x q}{q}\right\rangle = \frac{1}{2}\left\langle \nabla_x q, \frac{\nabla_x q}{q}\right\rangle = \frac{1}{2}\int \frac{\left\|\nabla_x q(x)\right\|^2}{q(x)} dx
\end{aligned}$$

- Our guess for S

$$S(q,x) = -\frac{1}{2}\left\|\nabla_x \log q(x)\right\|^2 - \Delta \log q(x)$$

- We calculate

$$\begin{aligned}
S(q,q) &= \mathbb{E}_{x\sim q}\left[S(q,x)\right] = \int q(x) S(q,x)\, dx \\
&= -\int q(x)\left(\frac{1}{2}\left\|\nabla_x \log q(x)\right\|^2 + \Delta \log q(x)\right) dx \\
&= -\int \left(\frac{1}{2}\frac{\left\|\nabla_x q(x)\right\|^2}{q(x)} - q(x)\nabla_x^T \nabla_x \log q(x)\right) dx \\
&= -\frac{1}{2}\left\langle \frac{\nabla_x q}{q}, \nabla_x q\right\rangle + \left\langle q, \nabla_x^T \nabla_x \log q\right\rangle \\
&= -\frac{1}{2}\left\langle \frac{\nabla_x q}{q}, \nabla_x q\right\rangle + \left\langle \nabla_x q, \frac{\nabla_x q}{q}\right\rangle = \frac{1}{2}\left\langle \nabla_x q, \frac{\nabla_x q}{q}\right\rangle = \frac{1}{2}\int \frac{\left\|\nabla_x q(x)\right\|^2}{q(x)} dx
\end{aligned}$$

Score matching

- Recall PSR

$$S(q,x) = -\frac{1}{2}\left\|\nabla_x \log q(x)\right\|^2 - \Delta \log q(x)$$

- Depends only on $q(x)$ not $q(x')$ for any $x' \neq x$
  - and 1st and 2nd derivatives of q at x
  - A local PSR
- $S(q,x) = S(q/c, x)$ for any $c > 0$
  - We cannot directly estimate the partition function of q
  - We can compare likelihood ratio of two points $q(x)/q(x')$

Score matching

- For an unknown data distribution p_d the goal is to minimize

$$J_{SM}(\theta) = \mathbb{E}_{x\sim p_d}\left[-S(p_\theta, x)\right] = \mathbb{E}_{x\sim p_d}\left[\frac{1}{2}\left\|\nabla_x \log p_\theta(x)\right\|^2 + \Delta \log p_\theta(x)\right]$$

- Given N training samples $\{x_i\}$ its sample version is

$$\frac{1}{N}\sum_i \left(\frac{1}{2}\left\|\nabla_x \log p_\theta(x_i)\right\|^2 + \Delta \log p_\theta(x_i)\right) \to \min_q$$

  - 1st term: x_i aims to be a critical point of $\log p_\theta$
  - 2nd term: x_i should be a local maximum of $\log p_\theta$

Score matching

- In the original paper, Hyvärinen started from the following objective

$$
\begin{aligned}
J(q) &= \frac{1}{2}\mathbb{E}_{x\sim p_d}\left[\|\nabla_x \log q(x) - \nabla_x \log p_d\|^2\right] \\
&\doteq \frac{1}{2}\mathbb{E}_{x\sim p_d}\left[\|\nabla \log q\|^2\right] - \mathbb{E}_{x\sim p_d}\left[\langle \nabla \log q(x), \nabla \log p_d(x)\rangle\right] \\
&= \frac{1}{2}\mathbb{E}_{x\sim p_d}\left[\|\nabla \log q\|^2\right] - \int p_d(x)\,\langle \nabla \log q(x), \nabla \log p_d(x)\rangle\, dx \\
&= \frac{1}{2}\mathbb{E}_{x\sim p_d}\left[\|\nabla \log q\|^2\right] - \int \langle \nabla \log q(x), \nabla p_d(x)\rangle\, dx \\
&= \frac{1}{2}\mathbb{E}_{x\sim p_d}\left[\|\nabla \log q\|^2\right] - \langle \nabla \log q, \nabla p_d\rangle \\
&= \frac{1}{2}\mathbb{E}_{x\sim p_d}\left[\|\nabla \log q\|^2\right] - \left\langle \nabla^T \nabla \log q, p_d\right\rangle \\
&= \mathbb{E}_{x\sim p_d}\left[\tfrac{1}{2}\|\nabla \log q\|^2 + \Delta \log q(x)\right]
\end{aligned}
$$

- Matching of scores (in the L^2-sense)

Score matching

- In the original paper, Hyvärinen started from the following objective

$$\begin{aligned}
J(q) &= \frac{1}{2}\mathbb{E}_{x\sim p_d}\left[\|\nabla_x \log q(x) - \nabla_x \log p_d\|^2\right] \\
&\doteq \frac{1}{2}\mathbb{E}_{x\sim p_d}\left[\|\nabla \log q\|^2\right] - \mathbb{E}_{x\sim p_d}\left[\langle \nabla \log q(x), \nabla \log p_d(x)\rangle\right] \\
&= \frac{1}{2}\mathbb{E}_{x\sim p_d}\left[\|\nabla \log q\|^2\right] - \int p_d(x)\, \langle \nabla \log q(x), \nabla \log p_d(x)\rangle\, dx \\
&= \frac{1}{2}\mathbb{E}_{x\sim p_d}\left[\|\nabla \log q\|^2\right] - \int \langle \nabla \log q(x), \nabla p_d(x)\rangle\, dx \\
&= \frac{1}{2}\mathbb{E}_{x\sim p_d}\left[\|\nabla \log q\|^2\right] - \langle \nabla \log q, \nabla p_d\rangle \\
&= \frac{1}{2}\mathbb{E}_{x\sim p_d}\left[\|\nabla \log q\|^2\right] - \left\langle \nabla^T \nabla \log q, p_d\right\rangle \\
&= \mathbb{E}_{x\sim p_d}\left[\tfrac{1}{2}\|\nabla \log q\|^2 + \Delta \log q(x)\right]
\end{aligned}$$

- Matching of scores (in the L^2-sense)

Score matching

- In the original paper, Hyvärinen started from the following objective

$$
\begin{aligned}
J(q) &= \frac{1}{2}\mathbb{E}_{x\sim p_d}\left[\left\|\nabla_x \log q(x) - \nabla_x \log p_d\right\|^2\right] \\
&\doteq \frac{1}{2}\mathbb{E}_{x\sim p_d}\left[\left\|\nabla \log q\right\|^2\right] - \mathbb{E}_{x\sim p_d}\left[\langle \nabla \log q(x), \nabla \log p_d(x)\rangle\right] \\
&= \frac{1}{2}\mathbb{E}_{x\sim p_d}\left[\left\|\nabla \log q\right\|^2\right] - \int p_d(x)\left\langle \nabla \log q(x), \nabla \log p_d(x)\right\rangle\, dx \\
&= \frac{1}{2}\mathbb{E}_{x\sim p_d}\left[\left\|\nabla \log q\right\|^2\right] - \int \left\langle \nabla \log q(x), \nabla p_d(x)\right\rangle\, dx \\
&= \frac{1}{2}\mathbb{E}_{x\sim p_d}\left[\left\|\nabla \log q\right\|^2\right] - \left\langle \nabla \log q, \nabla p_d\right\rangle \\
&= \frac{1}{2}\mathbb{E}_{x\sim p_d}\left[\left\|\nabla \log q\right\|^2\right] - \left\langle \nabla^T \nabla \log q, p_d\right\rangle \\
&= \mathbb{E}_{x\sim p_d}\left[\tfrac{1}{2}\left\|\nabla \log q\right\|^2 + \Delta \log q(x)\right]
\end{aligned}
$$

- Matching of scores (in the L^2-sense)

Score matching

- In the original paper, Hyvärinen started from the following objective

$$
\begin{aligned}
J(q) &= \frac{1}{2}\mathbb{E}_{x\sim p_d}\left[\|\nabla_x \log q(x) - \nabla_x \log p_d\|^2\right] \\
&\doteq \frac{1}{2}\mathbb{E}_{x\sim p_d}\left[\|\nabla \log q\|^2\right] - \mathbb{E}_{x\sim p_d}\left[\langle \nabla \log q(x), \nabla \log p_d(x)\rangle\right] \\
&= \frac{1}{2}\mathbb{E}_{x\sim p_d}\left[\|\nabla \log q\|^2\right] - \int p_d(x)\,\langle \nabla \log q(x), \nabla \log p_d(x)\rangle\ dx \\
&= \frac{1}{2}\mathbb{E}_{x\sim p_d}\left[\|\nabla \log q\|^2\right] - \int \langle \nabla \log q(x), \nabla p_d(x)\rangle\ dx \\
&= \frac{1}{2}\mathbb{E}_{x\sim p_d}\left[\|\nabla \log q\|^2\right] - \langle \nabla \log q, \nabla p_d\rangle \\
&= \frac{1}{2}\mathbb{E}_{x\sim p_d}\left[\|\nabla \log q\|^2\right] - \langle \nabla^T \nabla \log q, p_d\rangle \\
&= \mathbb{E}_{x\sim p_d}\left[\tfrac{1}{2}\|\nabla \log q\|^2 + \Delta \log q(x)\right]
\end{aligned}
$$

- Matching of scores (in the L^2-sense)

Score matching

- In the original paper, Hyvärinen started from the following objective

$$
\begin{aligned}
J(q) &= \frac{1}{2}\mathbb{E}_{x\sim p_d}\left[\|\nabla_x \log q(x) - \nabla_x \log p_d\|^2\right] \\
&\doteq \frac{1}{2}\mathbb{E}_{x\sim p_d}\left[\|\nabla \log q\|^2\right] - \mathbb{E}_{x\sim p_d}\left[\langle \nabla \log q(x), \nabla \log p_d(x)\rangle\right] \\
&= \frac{1}{2}\mathbb{E}_{x\sim p_d}\left[\|\nabla \log q\|^2\right] - \int p_d(x)\left\langle \nabla \log q(x), \nabla \log p_d(x)\right\rangle\, dx \\
&= \frac{1}{2}\mathbb{E}_{x\sim p_d}\left[\|\nabla \log q\|^2\right] - \int \left\langle \nabla \log q(x), \nabla p_d(x)\right\rangle\, dx \\
&= \frac{1}{2}\mathbb{E}_{x\sim p_d}\left[\|\nabla \log q\|^2\right] - \left\langle \nabla \log q, \nabla p_d\right\rangle \\
&= \frac{1}{2}\mathbb{E}_{x\sim p_d}\left[\|\nabla \log q\|^2\right] - \left\langle \nabla^T \nabla \log q, p_d\right\rangle \\
&= \mathbb{E}_{x\sim p_d}\left[\tfrac{1}{2}\|\nabla \log q\|^2 + \Delta \log q(x)\right]
\end{aligned}
$$

- Matching of scores (in the L^2-sense)

Score matching

- In the original paper, Hyvärinen started from the following objective

$$\begin{aligned}
J(q) &= \frac{1}{2}\mathbb{E}_{x\sim p_d}\left[\left\|\nabla_x \log q(x) - \nabla_x \log p_d\right\|^2\right] \\
&\doteq \frac{1}{2}\mathbb{E}_{x\sim p_d}\left[\left\|\nabla \log q\right\|^2\right] - \mathbb{E}_{x\sim p_d}\left[\langle \nabla \log q(x), \nabla \log p_d(x)\rangle\right] \\
&= \frac{1}{2}\mathbb{E}_{x\sim p_d}\left[\left\|\nabla \log q\right\|^2\right] - \int p_d(x)\,\langle \nabla \log q(x), \nabla \log p_d(x)\rangle\, dx \\
&= \frac{1}{2}\mathbb{E}_{x\sim p_d}\left[\left\|\nabla \log q\right\|^2\right] - \int \langle \nabla \log q(x), \nabla p_d(x)\rangle\, dx \\
&= \frac{1}{2}\mathbb{E}_{x\sim p_d}\left[\left\|\nabla \log q\right\|^2\right] - \langle \nabla \log q, \nabla p_d\rangle \\
&= \frac{1}{2}\mathbb{E}_{x\sim p_d}\left[\left\|\nabla \log q\right\|^2\right] - \left\langle \nabla^T \nabla \log q, p_d\right\rangle \\
&= \mathbb{E}_{x\sim p_d}\left[\tfrac{1}{2}\left\|\nabla \log q\right\|^2 + \Delta \log q(x)\right]
\end{aligned}$$

- Matching of scores (in the L^2-sense)

Score matching

- In the original paper, Hyvärinen started from the following objective

$$
\begin{aligned}
J(q) &= \frac{1}{2}\mathbb{E}_{x\sim p_d}\left[\left\|\nabla_x \log q(x) - \nabla_x \log p_d\right\|^2\right] \\
&\doteq \frac{1}{2}\mathbb{E}_{x\sim p_d}\left[\left\|\nabla \log q\right\|^2\right] - \mathbb{E}_{x\sim p_d}\left[\langle \nabla \log q(x), \nabla \log p_d(x)\rangle\right] \\
&= \frac{1}{2}\mathbb{E}_{x\sim p_d}\left[\left\|\nabla \log q\right\|^2\right] - \int p_d(x)\, \langle \nabla \log q(x), \nabla \log p_d(x)\rangle \; dx \\
&= \frac{1}{2}\mathbb{E}_{x\sim p_d}\left[\left\|\nabla \log q\right\|^2\right] - \int \langle \nabla \log q(x), \nabla p_d(x)\rangle \; dx \\
&= \frac{1}{2}\mathbb{E}_{x\sim p_d}\left[\left\|\nabla \log q\right\|^2\right] - \langle \nabla \log q, \nabla p_d\rangle \\
&= \frac{1}{2}\mathbb{E}_{x\sim p_d}\left[\left\|\nabla \log q\right\|^2\right] - \left\langle \nabla^T \nabla \log q, p_d\right\rangle \\
&= \mathbb{E}_{x\sim p_d}\left[\tfrac{1}{2}\left\|\nabla \log q\right\|^2 + \Delta \log q(x)\right]
\end{aligned}
$$

- Matching of scores (in the L^2-sense)

Score matching

- Gaussian example ($\Lambda = \Sigma^{-1} \succ 0$)

$$\log p_\theta(x) = -\frac{1}{2}(x-\mu)^T \Lambda (x-\mu) \qquad \nabla_x \log p_\theta(x) = -\Lambda(x-\mu)$$

$$\Delta \log p_\theta(x) = -\operatorname{trace}(\Lambda) = -\sum\nolimits_j \Lambda_{jj}$$

- Given N samples $\{x_i\}$

$$\begin{aligned} J(\mu, \Lambda) &= \frac{1}{2N} \sum\nolimits_i \|\Lambda(x_i - \mu)\|^2 - \sum\nolimits_j \Lambda_{jj} \\ \frac{\partial J}{\partial \mu} &= \frac{1}{N} \sum\nolimits_i \Lambda^2(\mu - x_i) \stackrel{!}{=} 0 \\ \frac{\partial J}{\partial \Lambda} &= \frac{1}{N} \sum\nolimits_i \Lambda(\mu - x_i)(\mu - x_i)^T - \mathrm{I} \stackrel{!}{=} 0 \end{aligned}$$

- Yields exactly the MLE

$$\mu = \frac{1}{N} \sum\nolimits_i x_i \qquad \Lambda^{-1} = \Sigma = \frac{1}{N} \sum\nolimits_i (\mu - x_i)(\mu - x_i)^T$$

Score matching

- Gaussian example ($\Lambda = \Sigma^{-1} \succ 0$)

$$\log p_\theta(x) = -\frac{1}{2}(x-\mu)^T \Lambda (x-\mu) \qquad \nabla_x \log p_\theta(x) = -\Lambda(x-\mu)$$

$$\Delta \log p_\theta(x) = -\operatorname{trace}(\Lambda) = -\sum\nolimits_j \Lambda_{jj}$$

- Given N samples $\{x_i\}$

$$J(\mu, \Lambda) = \frac{1}{2N}\sum\nolimits_i \|\Lambda(x_i - \mu)\|^2 - \sum\nolimits_j \Lambda_{jj}$$

$$\frac{\partial J}{\partial \mu} = \frac{1}{N}\sum\nolimits_i \Lambda^2(\mu - x_i) \stackrel{!}{=} 0$$

$$\frac{\partial J}{\partial \Lambda} = \frac{1}{N}\sum\nolimits_i \Lambda(\mu - x_i)(\mu - x_i)^T - \mathrm{I} \stackrel{!}{=} 0$$

- Yields exactly the MLE

$$\mu = \frac{1}{N}\sum\nolimits_i x_i \qquad \Lambda^{-1} = \Sigma = \frac{1}{N}\sum\nolimits_i (\mu - x_i)(\mu - x_i)^T$$

Score matching

- Gaussian example ($\Lambda = \Sigma^{-1} \succ 0$)

$$\log p_\theta(x) = -\frac{1}{2}(x-\mu)^T \Lambda (x-\mu) \qquad \nabla_x \log p_\theta(x) = -\Lambda(x-\mu)$$

$$\Delta \log p_\theta(x) = -\operatorname{trace}(\Lambda) = -\sum\nolimits_j \Lambda_{jj}$$

- Given N samples $\{x_i\}$

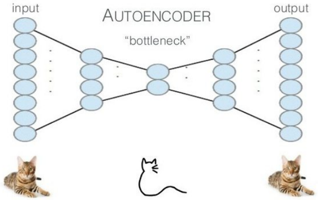
$$\begin{aligned} J(\mu, \Lambda) &= \frac{1}{2N} \sum\nolimits_i \|\Lambda(x_i - \mu)\|^2 - \sum\nolimits_j \Lambda_{jj} \\ \frac{\partial J}{\partial \mu} &= \frac{1}{N} \sum\nolimits_i \Lambda^2(\mu - x_i) \stackrel{!}{=} 0 \\ \frac{\partial J}{\partial \Lambda} &= \frac{1}{N} \sum\nolimits_i \Lambda(\mu - x_i)(\mu - x_i)^T - \mathrm{I} \stackrel{!}{=} 0 \end{aligned}$$

- Yields exactly the MLE

$$\mu = \frac{1}{N} \sum\nolimits_i x_i \qquad \Lambda^{-1} = \Sigma = \frac{1}{N} \sum\nolimits_i (\mu - x_i)(\mu - x_i)^T$$

A connection with auto-encoders

- "An auto-encoder reconstructs the input, which is going through a bottleneck layer"
- Unsupervised feature learning approach



Formal justification? (Probabilistic) interpretation?

- Score matching
  - Vincent, "A Connection Between Score Matching and Denoising Autoencoders"
  - Swersky et al, "On autoencoders and score matching for energy based models"
  - Kamyshanska & Memisevic, "The potential energy of an autoencoder"
- Variational Bayes
  - Discussed later

Score matching

- Let S be a non-negative function (e.g. $S(z) = \log(1 + \exp(z)))$

$$\log p(x) = -\frac{1}{2}\|x\|^2 + \sum_k S(w_k^T x) \qquad x \in \mathbb{R}^d$$

- Now

$$\nabla_x \log p(x) = \sum_k w_k s(w_k^T x) - x \qquad s(z) = S'(z)$$

$$\Delta \log p(x) = \sum_k \operatorname{trace}\left(w_k w_k^T s'(w_k^T x)\right) - d = \sum_k \|w_k\|^2 s'(w_k^T x) - d$$

- Insert into score matching objective

$$J(W) = \mathbb{E}_{x \sim p_d}\left[\left\|x - \sum_k w_k s(w_k^T x)\right\|^2 + \sum_k \|w_k\|^2 s'(w_k^T x)\right] - d$$

$$= \mathbb{E}_{x \sim p_d}\Big[\underbrace{\left\|x - W\, s(W^T x)\right\|^2}_{\text{reconstruction error / auto-encoder loss}} + \underbrace{\sum_k \|w_k\|^2 s'(w_k^T x)}_{\text{regularization}}\Big] - d$$

- $W\, s(W^T x)$ is a 1-layer AE with s as its non-linear activation function

Score matching

- Let S be a non-negative function (e.g. $S(z) = \log(1 + \exp(z))$)

$$\log p(x) = -\frac{1}{2} \|x\|^2 + \sum_k S(w_k^T x) \qquad\qquad x \in \mathbb{R}^d$$

- Now

$$\nabla_x \log p(x) = \sum_k w_k s(w_k^T x) - x \qquad\qquad s(z) = S'(z)$$

$$\Delta \log p(x) = \sum_k \operatorname{trace}\left(w_k w_k^T s'(w_k^T x)\right) - d = \sum_k \|w_k\|^2 s'(w_k^T x) - d$$

- Insert into score matching objective

$$J(W) = \mathbb{E}_{x \sim p_d}\left[\left\| x - \sum_k w_k s(w_k^T x) \right\|^2 + \sum_k \|w_k\|^2 s'(w_k^T x) \right] - d$$

$$= \mathbb{E}_{x \sim p_d}\left[\underbrace{\left\| x - W\, s(W^T x) \right\|^2}_{\text{reconstruction error / auto-encoder loss}} + \underbrace{\sum_k \|w_k\|^2 s'(w_k^T x)}_{\text{regularization}} \right] - d$$

- $W\, s(W^T x)$ is a 1-layer AE with s as its non-linear activation function

Score matching

- Let S be a non-negative function (e.g. $S(z) = \log(1 + \exp(z))$)

$$\log p(x) = -\frac{1}{2}\|x\|^2 + \sum\nolimits_k S(w_k^T x) \qquad\qquad x \in \mathbb{R}^d$$

- Now

$$\nabla_x \log p(x) = \sum\nolimits_k w_k s(w_k^T x) - x \qquad\qquad s(z) = S'(z)$$

$$\Delta \log p(x) = \sum\nolimits_k \operatorname{trace}\left(w_k w_k^T s'(w_k^T x)\right) - d = \sum\nolimits_k \|w_k\|^2 s'(w_k^T x) - d$$

- Insert into score matching objective

$$J(W) = \mathbb{E}_{x\sim p_d}\left[\left\|x - \sum\nolimits_k w_k s(w_k^T x)\right\|^2 + \sum\nolimits_k \|w_k\|^2 s'(w_k^T x)\right] - d$$

$$= \mathbb{E}_{x\sim p_d}\Bigg[\underbrace{\left\|x - W\,s(W^T x)\right\|^2}_{\text{reconstruction error / auto-encoder loss}} + \underbrace{\sum\nolimits_k \|w_k\|^2 s'(w_k^T x)}_{\text{regularization}}\Bigg] - d$$

- $W\,s(W^T x)$ is a 1-layer AE with s as its non-linear activation function

Outline

Example: fitting Gaussians

- Multi-variate Gaussian distribution in D dimensions

$$p(x; \mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^D |\Sigma|}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)}$$

- When is maximum likelihood estimation easy?
  - Σ is unconstrained
  - Σ is block-diagonal (why?)
- Precision matrix $\Lambda := \Sigma^{-1}$
- What if components are known to be conditionally independent?
  - If x_i and x_j are conditionally independent, then $\Lambda_{ij} = \Lambda_{ji} = 0$
  - Enforce non-zero pattern on Λ
- Constraints on $\Sigma = \Lambda^{-1}$?
  - In general $NZ(\Sigma) \neq NZ(\Lambda)$
  - Exception: block-diagonal Σ

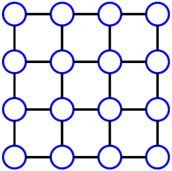
Example: fitting Gaussians

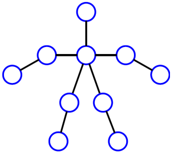
- Multi-variate Gaussian distribution in D dimensions

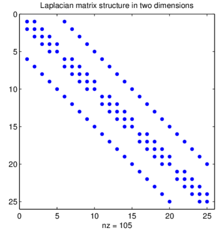
$$p(x;\mu,\Sigma) = \frac{1}{\sqrt{(2\pi)^D|\Sigma|}} e^{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)}$$

Discussion

- When are components of a Gaussian random vector conditionally independent?
- Are there other benefits in high dimensions?







Exercise 1: fitting Gaussians to MNIST patches

Exercise

MNIST dataset: `http://yann.lecun.com/exdb/mnist/`

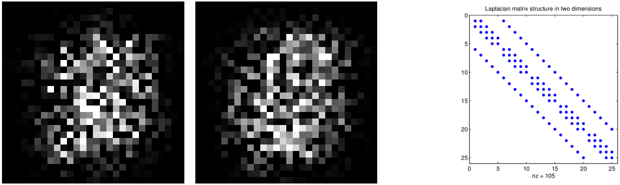
- Convert 8-bit data to $[0, 1]$ range and add per-pixel Gaussian noise $\varepsilon \sim \mathcal{N}(0, 1/100)$
- Model images patches as Gaussian

$$p_\theta(x) = \frac{1}{Z} e^{-\frac{1}{2}(x-\mu)^T \Lambda (x-\mu)}$$

- Subtract the empirical mean, leading to

$$p_\theta(x) = \frac{1}{Z} e^{-\frac{1}{2} x^T \Lambda x}$$

- Λ has 2D Laplacian structure: 4-connected neighboring pixels are correlated



Left: samples from a Gaussian with diagonal Λ. Right: 2D Laplacian NZ structure

Exercise 1: fitting Gaussians to MNIST patches

Exercise

MNIST dataset: `http://yann.lecun.com/exdb/mnist/`

- Estimate Λ via
  - NCE (explain your choice of $p_n(x')$)
  - cNCE (explain your choice of $p_n(x'|x)$) or score matching (coin flip)
- Use SGD (or RMSProp or ADAM) for gradient-based optimization
- Visualize samples from p_θ (with $\mathsf{A} = \Lambda^{-1/2} = \Sigma^{1/2}$)

$$x' \leftarrow \mu + \mathsf{A}\varepsilon \qquad \varepsilon \sim \mathcal{N}(\mathbf{0}, \mathsf{I})$$

- NCE: how close is the estimate of $\log Z$ to $\frac{D}{2}\log(2\pi) + \frac{1}{2}\log|\Sigma|$? $\qquad (D = 28^2)$
- Bonus exercise: rerun with 8-connected neighborhood assumption