

Inference and Representation

Lecture 6

Joan Bruna
Courant Institute, NYU



Lecture 6 Objectives

- Explaining Away and Clustering
 - Review of Gaussian Mixture Models
- The Expectation-Maximization Algorithm
- Markov-Chain Monte-Carlo (MCMC)

Explaining Away

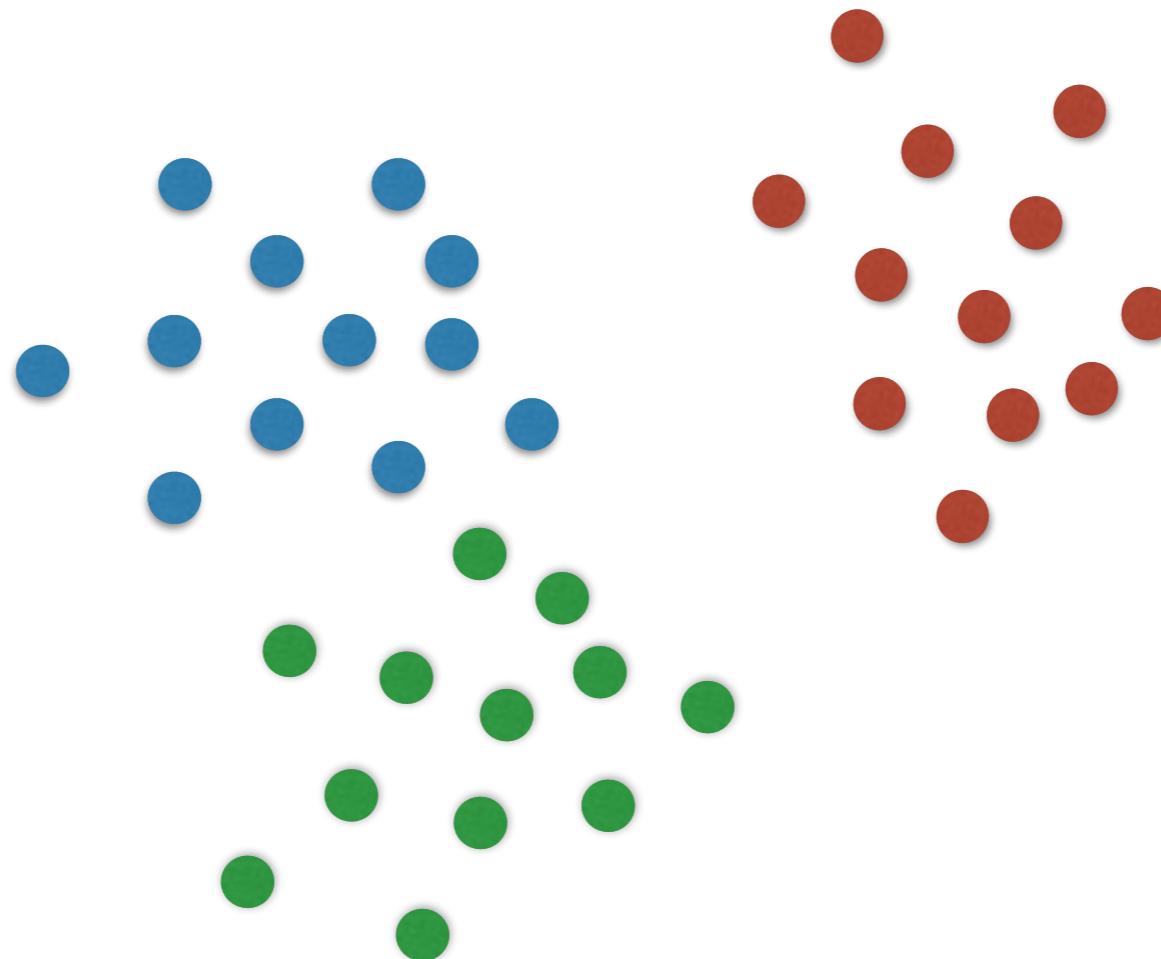
- The models we have seen so far ultimately construct latent variables by applying a linear transformation over the observed variables.
- Most interesting inferential tasks consider “competing” hypothesis.



– This is known as *explaining away*.

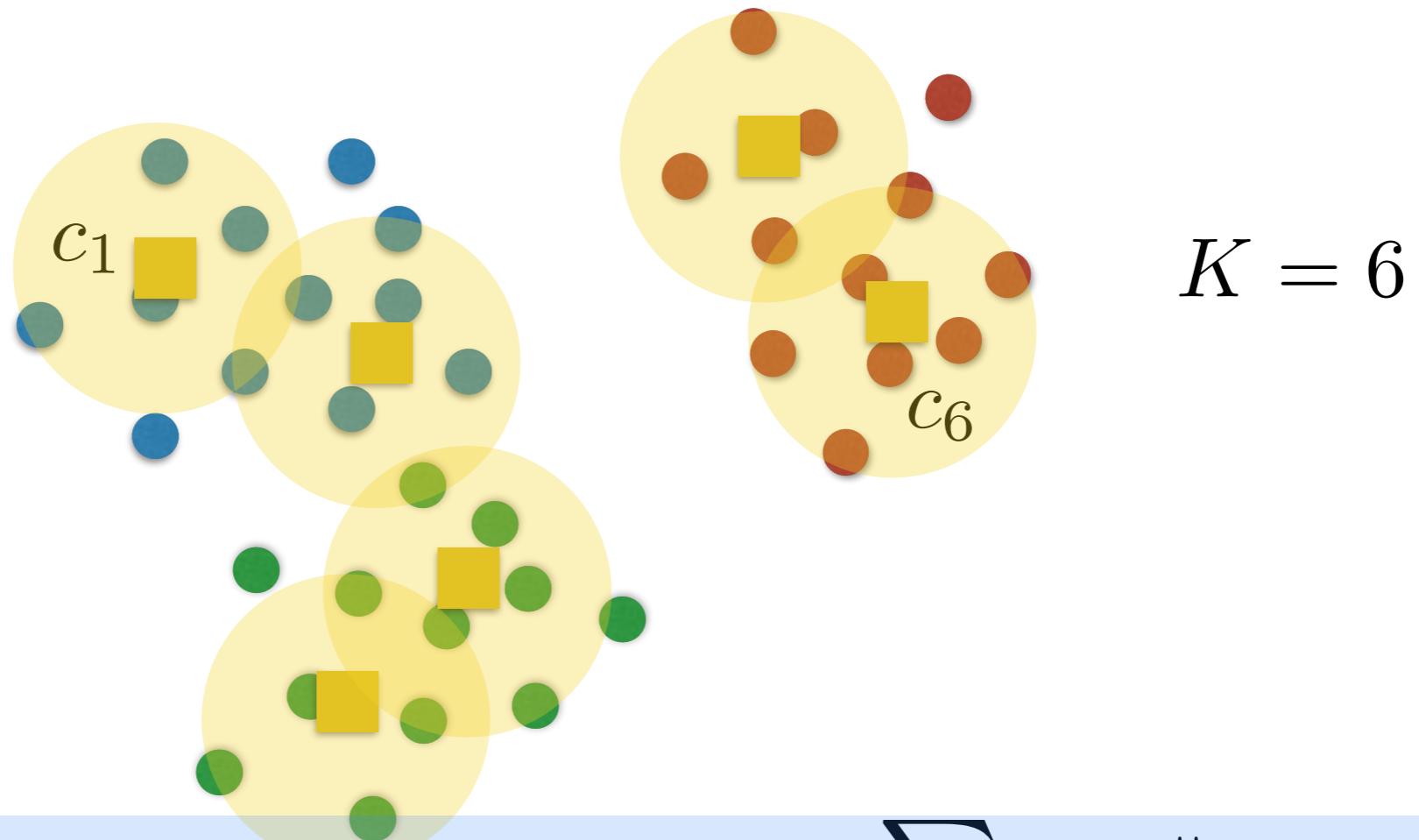
Latent Variables

- The simplest model that captures explaining away is K-means clustering:



Latent Variables

- The simplest model that captures explaining away is K-means clustering:



Given data $X = (x_1, \dots, x_n)$, $\min_{c_1, \dots, c_K} \sum_{i \leq n} \min_j \|x_i - c_j\|^2$

Floyd Algorithm

- For each i , we define r_i a one-hot vector of length K encoding its cluster.
- Cost function is

$$E(c, r) = \sum_i \sum_k r_i(k) \|x_i - c_k\|^2$$

Floyd Algorithm

- For each i , we define r_i a one-hot vector of length K encoding its cluster.

- Cost function is

$$E(c, r) = \sum_i \sum_k r_i(k) \|x_i - c_k\|^2$$

- Fixing c , we optimize r as

$$r_i \leftarrow \arg \min_k \|x_i - c_k\|$$

Given assignments r , optimize E with respect to c :

$$c_k = \frac{\sum_i r_i(k) x_i}{\sum_i r_i(k)}$$

*mean of all
datapoints falling
in cluster k*

Floyd Algorithm

- This iterative algorithm converges towards a local optimum (each step decreases the cost).
- It is in fact an instance of the Expectation-Maximization algorithm (EM).
- In that case, the discrete latent variables are the cluster assignments.

Gaussian Mixture Models (GMM)

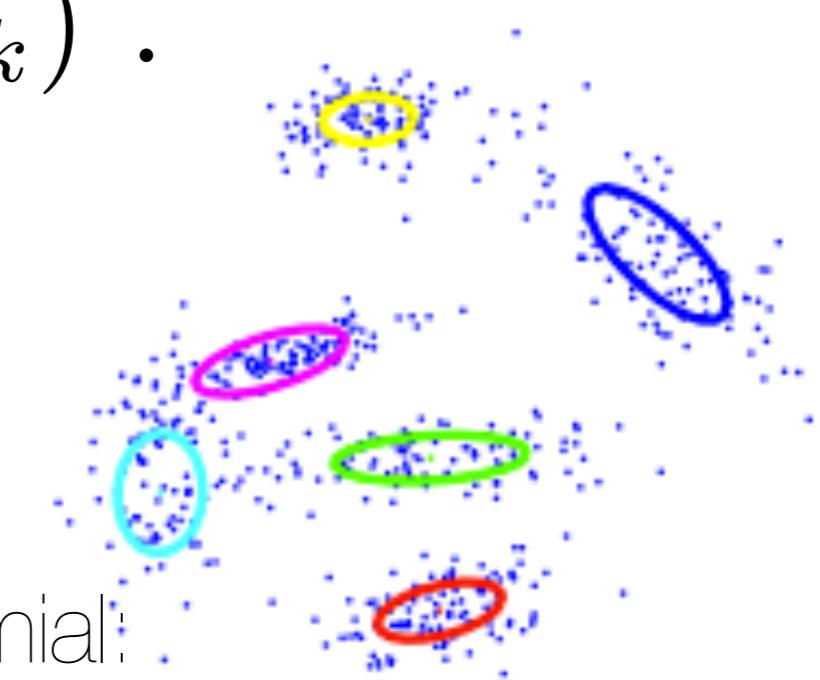
- A generalization of K-Means is given by a Gaussian Mixture:

$$k \sim \text{Mult}(\pi) , \quad x \sim \mathcal{N}(\mu_k, \Sigma_k) .$$

- This is also a discrete latent variable model:

$$z \in \{0, 1\}^K , \quad \sum_k z_k = 1 .$$

- The distribution of the latent variable is multinomial:



(figure from R.Salakhutdinov)

$$p(z_k = 1) = \pi_k , \quad 0 \leq \pi_k \leq 1 , \quad \sum_k \pi_k = 1 .$$

Gaussian Mixture Models (GMM)

- We can write

$$p(z) = \prod_{k=1}^K \pi_k^{z_k} \quad p(x \mid z_k = 1) = \mathcal{N}(x; \mu_k, \Sigma_k)$$

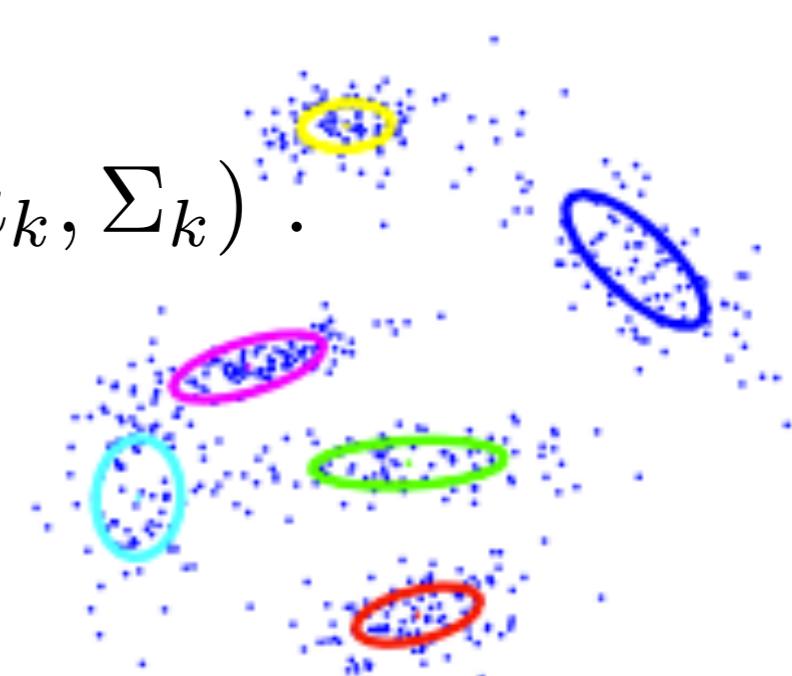
- Thus

$$p(x \mid z) = \prod_{k=1}^K \mathcal{N}(x; \mu_k, \Sigma_k)^{z_k}$$

- Joint and marginal distributions are given by

$$p(x, z) = p(x \mid z)p(z) ,$$

$$p(x) = \sum_z p(x, z) = \sum_{k=1}^K \pi_k \mathcal{N}(x; \mu_k, \Sigma_k) .$$



GMM and Posterior Inference

- What about the conditional $p(z \mid x)$? i.e., given data, which mixture components are “responsible”?

$$\begin{aligned} p(z_k = 1 \mid x) &= \frac{p(z_k = 1, x)}{\sum_{k' \leq K} p(z_{k'} = 1, x)} = \frac{p(z_k = 1)p(x \mid z_k = 1)}{\sum_{k' \leq K} p(z_{k'} = 1)p(x \mid z_{k'} = 1)} \\ &= \frac{\pi_k \mathcal{N}(x; \mu_k, \Sigma_k)}{\sum_{k'} \pi_{k'} \mathcal{N}(x; \mu_{k'}, \Sigma_{k'})} \end{aligned}$$

- The posterior probability that $z_k = 1$ is a weighted average of prior probabilities that depends upon the data.
- Q: How to estimate the parameters $\{\pi, \mu, \Sigma\}$?

Maximum Likelihood Estimation

- Given independent samples $X = \{x_1, \dots, x_n\}$, the total log-likelihood is

$$E(\pi, \mu, \Sigma) = \log p(X \mid \pi, \mu, \Sigma) = \sum_{i \leq n} \log \left(\sum_k \pi_k \mathcal{N}(x_i; \mu_k, \Sigma_k) \right)$$

- $$\frac{\partial E}{\partial \mu_k} = \sum_i \frac{\pi_k \mathcal{N}(x_i; \mu_k, \Sigma_k)}{\sum_{k'} \pi_{k'} \mathcal{N}(x_i; \mu_{k'}, \Sigma_{k'})} \Sigma_k^{-1} (x_i - \mu_k) .$$
- $$\mu_k = \frac{1}{N_k} \sum_i p(z_{i,k} = 1 \mid x_i) x_i , \quad N_k = \sum_i p(z_{i,k} = 1 \mid x_i) .$$

Thus the mean μ_k is the weighted average of datapoints, with weights given by the posterior probabilities of belonging to component k .

Maximum Likelihood Estimation

- Similarly

$$\frac{\partial E}{\partial \Sigma_k} = 0 \Rightarrow \Sigma_k = \frac{1}{N_k} \sum_i p(z_{i,k} = 1 \mid x_i) (x_i - \mu_k) (x_i - \mu_k)^T.$$

$$\frac{\partial E}{\partial \pi_k} = 0 \Rightarrow \pi_k = \frac{N_k}{n}.$$

- MLE parameters do not have closed-form solution
 - Parameters depend upon posterior probabilities $p(z_k = 1 \mid x)$, which themselves depend upon parameters.
- Iterative algorithm: Expectation-Maximization (EM):
 - E-step: Update posterior probabilities with parameters fixed.
 - M-step: Update parameters with posterior probabilities fixed.

The EM algorithm

- It is designed to find MLE solutions of latent variable models.
- In general, we have log-likelihoods of the form

$$\log p(X \mid \theta) = \log \left(\sum_Z p(X, Z \mid \theta) \right), \quad \begin{aligned} \theta &= \text{model parameters .} \\ Z &= \text{latent variables} \end{aligned}$$

The EM algorithm

- It is designed to find MLE solutions of latent variable models.
- In general, we have log-likelihoods of the form

$$\log p(X \mid \theta) = \log \left(\sum_Z p(X, Z \mid \theta) \right), \quad \begin{aligned} \theta &= \text{model parameters .} \\ Z &= \text{latent variables} \end{aligned}$$

- Using current parameters θ_{old} , we compute the expected total likelihood of the model (E-step):
- Then we update the parameters to maximize this likelihood:

$$Q(\theta, \theta_{old}) = \mathbb{E}_{Z \sim p(Z \mid X, \theta_{old})} \log p(X, Z \mid \theta)$$

$$\theta_{new} = \arg \max_{\theta} Q(\theta, \theta_{old}) .$$

EM and Variational Bound

- Q: Does this algorithm monotonically improve the likelihood?
- Assume for now that latent variables are discrete.

EM and Variational Bound

- Q: Does this algorithm monotonically improve the likelihood?
- Assume for now that latent variables are discrete.
- For any distribution $q(Z)$ over latent variables, we have

$$\begin{aligned}\log p(X \mid \theta) &= \log \left(\sum_Z p(X, Z \mid \theta) \right) = \log \left(\sum_Z q(Z) \frac{p(X, Z \mid \theta)}{q(Z)} \right) \\ &\geq \sum_Z q(Z) \log \left(\frac{p(X, Z \mid \theta)}{q(Z)} \right) = \mathcal{L}(q, \theta) .\end{aligned}$$

(Jensen's Inequality: $\mathbb{E}(f(X)) \geq f(\mathbb{E}(X))$ if f is convex)

Variational Bound

- We can express the variational lower bound as

$$\begin{aligned}\mathcal{L}(q, \theta) &= \mathbb{E}_{q(Z)} [\log p(X, Z \mid \theta)] - \mathbb{E}_{q(Z)} \log q(Z) \\ &= \mathbb{E}_{q(Z)} [\log p(X, Z \mid \theta)] + H(q) .\end{aligned}$$

$H(q)$: Entropy of $q(Z)$.

- Also, we have

$$\log p(X \mid \theta) = \mathcal{L}(q, \theta) + KL(q(z) \parallel p(z \mid x, \theta)) , \text{ where}$$

$$KL(q \parallel p) = - \sum_z q(z) \log \left(\frac{p(z)}{q(z)} \right)$$

is the Kullback-Leibler divergence.

Variational Bound

- Thus, the divergence $KL(q||p)$ measures how far our variational approximation $q(z)$ is from the true posterior, and directly controls the bound on the log-likelihood.
- Using $\log p(X \mid \theta) = \mathcal{L}(q, \theta) + KL(q(z)||p(z \mid x, \theta))$
- E-step: maximize lower bound $\mathcal{L}(q, \theta)$ with respect to q , holding parameters fixed.
- M-step: maximize lower bound $\mathcal{L}(q, \theta)$ with respect to parameters, holding q fixed.

Correctness of EM

- Suppose current parameter value is $\theta^{(n)}$.
- Consider the variational bound by picking $q(z) = p(z|x, \theta^{(n)})$:

$$\log p(X|\theta) - \log p(X|\theta^{(n)}) \geq \mathcal{L}(p(z|x, \theta^{(n)}), \theta)$$

$$\begin{aligned} &= \sum_z p(z|x, \theta^{(n)}) \log \left(\frac{p(x|z, \theta)p(z|\theta)}{p(z|x, \theta^{(n)})p(x|\theta^{(n)})} \right) \\ &= \Delta(\theta|\theta^{(n)}). \end{aligned}$$

with $\Delta(\theta^{(n)}|\theta^{(n)}) = 0$.

Correctness of EM

- Thus

$$\theta^{(n+1)} = \arg \max_{\theta} \Delta(\theta | \theta^{(n)})$$

$$\theta^{(n+1)} = \arg \max_{\theta} \Delta(\theta | \theta^{(n)})$$

$$= \arg \max_{\theta} \sum_z p(z|x, \theta^{(n)}) \log \{ (p(x|z, \theta) p(z|\theta)) \}$$

$$= \arg \max_{\theta} \sum_z p(z|x, \theta^{(n)}) \log \{ (p(x, z|\theta)) \}$$

$$= \arg \max_{\theta} \mathbb{E}_{z \sim p(z|x, \theta^{(n)})} \log p(X, Z|\theta) .$$

Approximate Inference

- We will see two major approximate inference paradigms:
 - Variational Inference (next lectures)
 - Markov-Chain Monte-Carlo (now)

Approximate Inference

- We will see two major approximate inference paradigms:
 - Variational Inference (next lectures)
 - Markov-Chain Monte-Carlo (now)
- MCMC
 - the Metropolis-Hastings algorithm
 - revisiting Gibbs sampling
 - extensions (Langevin, HMC)
 - *Interlude*: assessing sample quality in MCMC with Stein's Method.

Approximate Inference via Sampling

- Suppose $p(x) = \frac{\tilde{p}(x)}{Z}$ and we only know $\tilde{p}(x)$.
- Q: How to infer properties of $p(x)$ without knowing Z ?
 - e.g, marginal distributions

$$\bar{p}(x_1) = \int p(x) dx_2 \dots dx_n .$$

– or more generally moments of the form

$$\mathbb{E}(\phi(X)) = \int \phi(x)p(x)dx , \quad X \sim p(x)$$

Approximate Inference via Sampling

- Suppose $p(x) = \frac{\tilde{p}(x)}{Z}$ and we only know $\tilde{p}(x)$.
- Q: How to infer properties of $p(x)$ without knowing Z ?
 - e.g, marginal distributions

$$\bar{p}(x_1) = \int p(x) dx_2 \dots dx_n .$$

– or more generally moments of the form

$$\mathbb{E}(\phi(X)) = \int \phi(x)p(x)dx , \quad X \sim p(x)$$

- **Monte-Carlo** approximation replaces integrals with sums over iid samples:

$$\mathbb{E}(\phi(X)) \approx \frac{1}{N} \sum_{i \leq N} \phi(X_i) , \quad X_i \sim p .$$

Approximate Inference via Sampling

- Suppose $p(x) = \frac{\tilde{p}(x)}{Z}$ and we only know $\tilde{p}(x)$.
- Q: How to infer properties of $p(x)$ without knowing Z ?
 - e.g, marginal distributions

$$\bar{p}(x_1) = \int p(x) dx_2 \dots dx_n .$$

– or more generally moments of the form

$$\mathbb{E}(\phi(X)) = \int \phi(x)p(x)dx , \quad X \sim p(x)$$

- **Monte-Carlo** approximation replaces integrals with sums over iid samples:

$$\mathbb{E}(\phi(X)) \approx \frac{1}{N} \sum_{i \leq N} \phi(X_i) , \quad X_i \sim p .$$

- **Key property:** This is also (mostly) true even when samples are not independent!

Importance Sampling

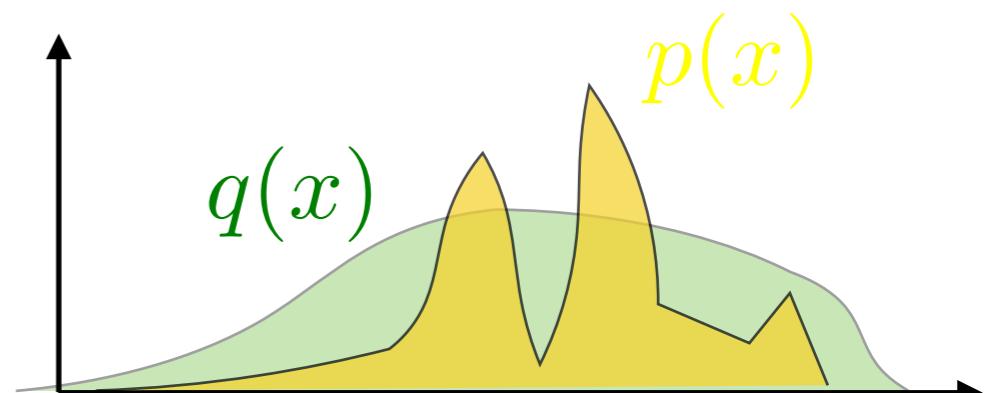
- Suppose first we know $p(x)$ for all x , but we don't know how to sample, nor how to integrate under $p(x)$.
- Crazy idea: Can we approximate $\mathbb{E}(f(X)) = \int p(x)f(x)dx$ with samples from *another* distribution with density $q(x)$?

Importance Sampling

- Suppose first we know $p(x)$ for all x , but we don't know how to sample, nor how to integrate under $p(x)$.
- Crazy idea: Can we approximate $\mathbb{E}(f(X)) = \int p(x)f(x)dx$ with samples from *another* distribution with density $q(x)$?

- Assume that

$$\text{Supp}(p) \subseteq \text{Supp}(q) .$$



- Then

$$\int f(x)p(x)dx = \int f(x)\frac{p(x)}{q(x)}q(x)dx = \mathbb{E}_q(f(x)w(x)) , \text{ where}$$

$$w(x) = \frac{p(x)}{q(x)} .$$

Importance Sampling

- So we can estimate the integral as

$$\widehat{\mathbb{E}(f(x))} = \frac{1}{N} \sum_{i \leq N} f(x^{(i)}) w(x^{(i)}) , \quad x^{(i)} \sim q .$$

Importance Sampling

- So we can estimate the integral as

$$\widehat{\mathbb{E}(f(x))} = \frac{1}{N} \sum_{i \leq N} f(x^{(i)}) w(x^{(i)}) , \quad x^{(i)} \sim q .$$

- Q: How good is this estimator?

Unbiased: $\mathbb{E}_q \left\{ \widehat{\mathbb{E}_p(f(x))} \right\} = \mathbb{E}_p\{f(x)\}.$

$$\text{var}_q \{f(x)w(x)\} = \mathbb{E}_q\{f^2(x)w^2(x)\} - \mathbb{E}_p\{f(x)\}^2 .$$

Importance Sampling

- So we can estimate the integral as

$$\widehat{\mathbb{E}(f(x))} = \frac{1}{N} \sum_{i \leq N} f(x^{(i)}) w(x^{(i)}) , \quad x^{(i)} \sim q .$$

- Q: How good is this estimator?

Unbiased: $\mathbb{E}_q \left\{ \widehat{\mathbb{E}_p(f(x))} \right\} = \mathbb{E}_p\{f(x)\}.$

$$\text{var}_q \{f(x)w(x)\} = \mathbb{E}_q\{f^2(x)w^2(x)\} - \mathbb{E}_p\{f(x)\}^2 .$$

- We can consider the proposal $q(x)$ that minimizes the variance:

$$\mathbb{E}_q\{f^2(x)w^2(x)\} \geq (\mathbb{E}_q|f(x)|w(x)|)^2 = \left(\int |f(x)|p(x)dx \right)^2 ,$$

which is attained if $q(x) = \frac{|f(x)|p(x)}{\int |f(x)|p(x)dx}$

Importance Sampling

- Remarks:
 - In general, it is not possible to sample from optimal $q(x)$.
 - Importance sampling can yield lower variance than original Monte-Carlo estimates.
- Q: Is it possible to do importance sampling if $p(x)$ is only known up to a multiplicative constant?

We have $\mathbb{E}_p\{f(x)\} = \frac{\int f(x)w(x)q(x)dx}{\int w(x)q(x)dx}$, with $w(x) = \frac{p(x)}{q(x)}$.

So now we can estimate both numerator and denominator:

$$\widehat{\mathbb{E}_p\{f(x)\}} = \frac{\sum_{i \leq N} f(x^{(i)})w(x^{(i)})}{\sum_{i \leq N} w(x^{(i)})} = \sum_{i \leq N} f(x^{(i)})\tilde{w}(x^{(i)}) ,$$
$$\tilde{w}(x^{(i)}) = \frac{w(x^{(i)})}{\sum_{j \leq N} w(x^{(j)})} : \text{normalized importance weights.}$$

Limitations of Importance Sampling

- The efficiency of importance sampling relies on how well we can approximate the *typical* set of $p(x)$:

$$\Lambda(p) = \{x \in \Omega ; \log p(x) \approx \mathbb{E}\{\log p(X)\}\} .$$

- As the dimensionality of Ω increases, capturing the typical set becomes intractable.
- Therefore, we need a more powerful, adaptive tool to capture the typical samples of $p(x)$, that can extend to high-dimensional densities.

Markov-Chain Monte-Carlo (MCMC)

- Recall the challenge: produce nearly independent samples of $p(x)$ without knowing the partition function Z in $p(x) = \frac{\tilde{p}(x)}{Z}$
- As it turns out, we can design a Markov Chain whose stationary distribution is $p(x)$ without knowing Z
- Q: How does that solve our problem?
 1. Run the Markov chain from an arbitrary initial point x_0
 2. After enough iterations, the resulting point will be a sample of $p(x)$
- So:
 - How to design the chain?
 - How many steps are needed?

Markov Chains flash review

- We consider:

A state-space Ω (discrete or continuous)

A *stochastic* kernel P that updates the densities at each time step:

$$q^{(t+1)}(u) = \int_{\Omega} P(u, v) q^{(t)}(v) dv .$$
$$P(u, v) \geq 0 , \quad \int P(u, v) dv = 1 , \forall u \in \Omega .$$

- In a discrete state space, the kernel can be viewed as a matrix, and

$P(k, l)$ = Probability to transition from state l to k .

Markov-Chains Flash Review

Definition: A Markov Chain is irreducible if $\Pr(\text{visit state } u \text{ from state } v) > 0$ for all $u, v \in \Omega$.

Definition: A Markov Chain is aperiodic if there exists $n > 0$ such that $\Pr(X_{n'} = i \mid X_0 = i) > 0$ for all $n' \geq n$.

Definition: A stationary distribution π of a Markov Chain P is such that $\pi = P\pi$.

Theorem: [Perron-Frobenius] An irreducible aperiodic Markov Chain P admits a unique stationary distribution.

Remark: The stationary distribution(s) correspond to eigenvectors of P of leading eigenvalue ($= 1$).

Metropolis-Hastings Algorithm

Definition: A Markov Chain P is reversible with respect to π if

$$\forall u, v \in \Omega, \pi_u P(u, v) = \pi_v P(v, u).$$

Fact: If P is reversible wrt π then π is a stationary for P .

$$\int \pi(u) P(v, u) du = \int \pi(v) P(u, v) du = \pi(v) \int P(u, v) du = \pi(v).$$

- How to build a Markov Chain such that it is reversible wrt $p(x)$? without involving the partition function?

Metropolis-Hastings

- We start with a *proposal* Markov chain K such that
 - $\forall i \in \Omega, K_{i,i} > 0$, and
 - $G = (\Omega, E(K))$ is connected, where i and j are connected if $K_{i,j}K_{j,i} > 0$.(we can travel from any state to any other state in finite time using K).
- Then we adapt it to $p(x)$ as follows:

$$R(x_i, x_j) = \min \left\{ 1, \frac{\tilde{p}(x_j)K(x_j, x_i)}{\tilde{p}(x_i)K(x_i, x_j)} \right\},$$

$$P(x_i, x_j) = K(x_i, x_j)R(x_i, x_j) + \delta(x_i - x_j)r(x_i), \text{ with}$$

$$\begin{aligned} r(x_i) &= \int_{\Omega} K(y, x_i)(1 - R(x_i, y))dy . \\ &= \text{Probability to reject the update.} \end{aligned}$$

Metropolis-Hastings

Fact: The kernel P is irreducible, aperiodic and satisfies the detailed balance condition wrt p .
Therefore p is the unique stationary distribution of P .

Assume wlog $\tilde{p}(x_j)K(x_j, x_i) \geq \tilde{p}(x_i)K(x_i, x_j)$.

Then $R(x_i, x_j) = 1$ and $R(x_j, x_i) = \frac{p(x_i)K(x_i, x_j)}{p(x_j)K(x_j, x_i)}$.

$$\begin{aligned} p(x_i)P(x_i, x_j) &= p(x_i)K(x_i, x_j) = p(x_i)K(x_i, x_j) \frac{p(x_j)K(x_j, x_i)}{p(x_j)K(x_j, x_i)} \\ &= \frac{p(x_i)K(x_i, x_j)}{p(x_j)K(x_j, x_i)} p(x_j)K(x_j, x_i) = R(x_j, x_i)K(x_j, x_i)p(x_j) \\ &= p(x_j)P(x_j, x_i) \end{aligned}$$

Irreducible by assumption on K .
(ex: Why is it aperiodic?)

Metropolis-Hastings

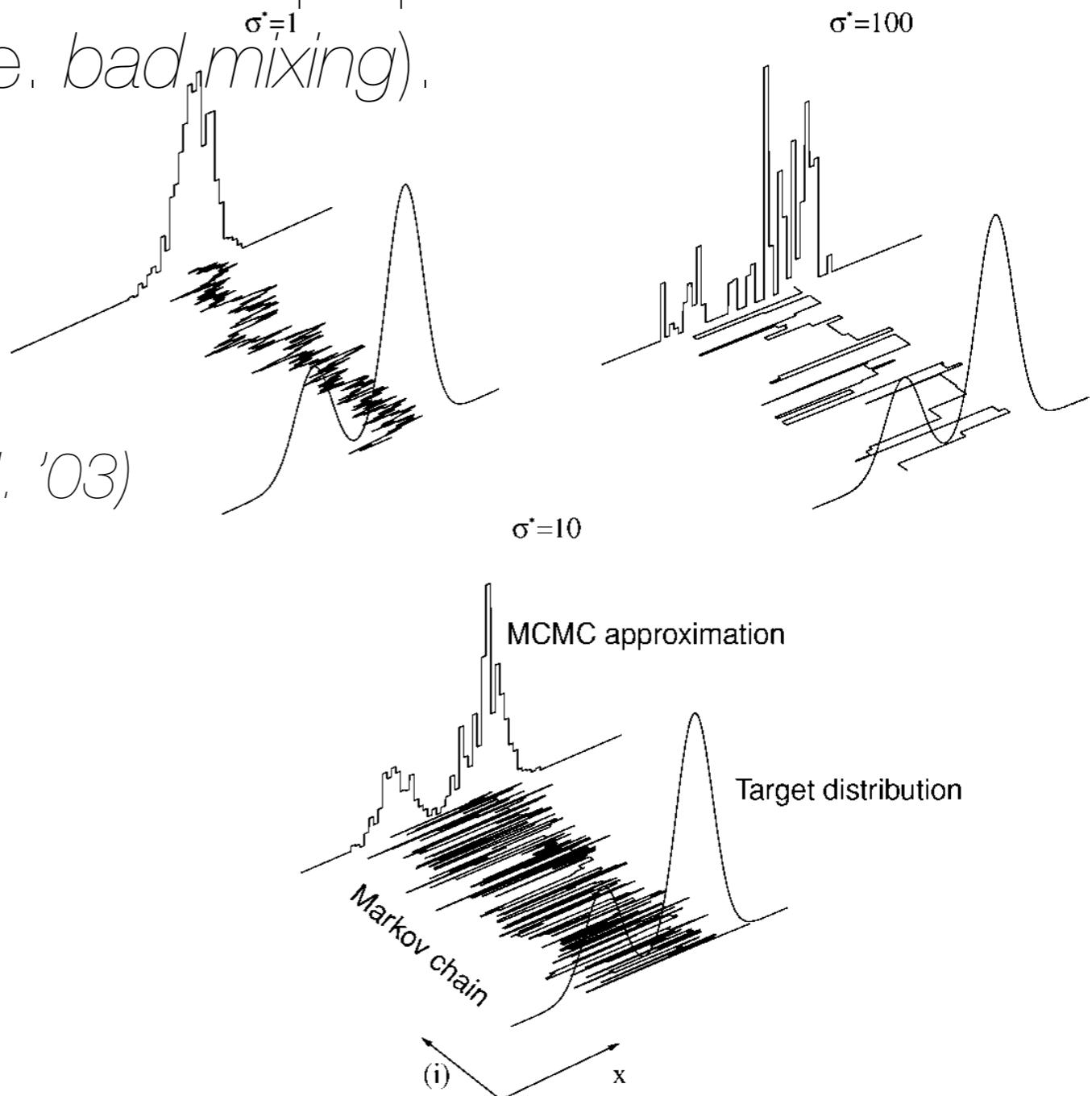
- Practical Implementation:
 1. Given current state x_i , propose next step by sampling $x_s \sim K(\cdot, x_i)$
 2. Compute $R = \min \left(1, \frac{p(x_s)K(x_s, x_i)}{p(x_i)K(x_i, x_s)} \right)$.
 3. Accept new state x_s with probability R and go to step 1.

- Two important particular cases:
 - $K(x_j, x_i) = q(x_j)$ is independent of the current state.
Acceptance probability becomes
$$R(x_i, x_j) = \min \left(1, \frac{p(x_j)q(x_i)}{p(x_i)q(x_j)} \right) = \min \left(1, \frac{w(x_j)}{w(x_i)} \right).$$
 - Similar to importance sampling, except that here samples are correlated (why?)
 - $K(x_j, x_i) = K(x_i, x_j)$ is symmetric.
Acceptance probability becomes $R(x_i, x_j) = \min \left(1, \frac{p(x_j)}{p(x_i)} \right)$.
 - We always accept the sample as long as it “climbs” wrt the true density.

Metropolis-Hastings

- This algorithm seems to be *magic*: it only asks us to cover the target density with our proposal distribution!
- There is no “free lunch”: a poor choice of proposal distribution results in bad approximations (i.e. *bad mixing*).

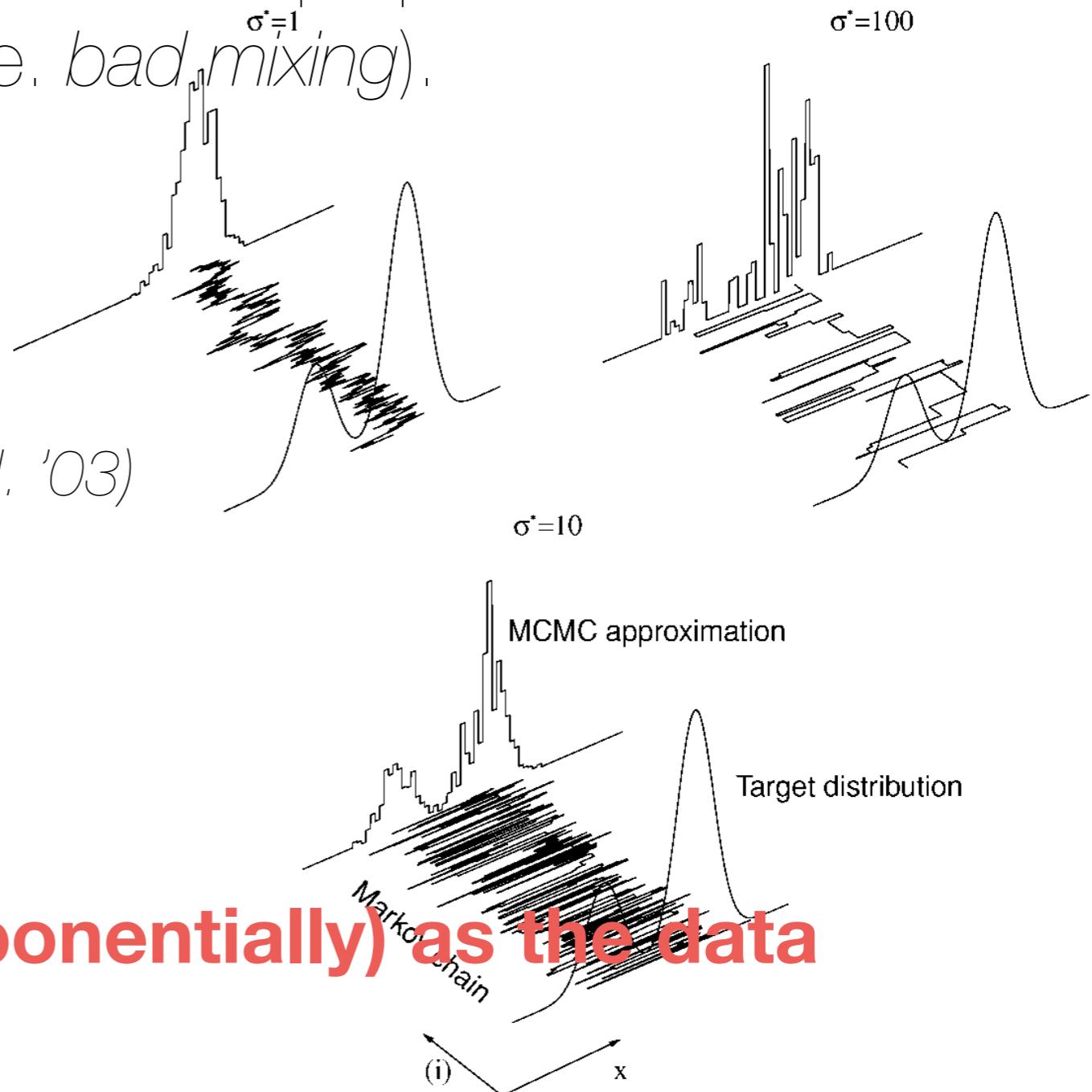
(figure from Andrieu et al. '03)



Metropolis-Hastings

- This algorithm seems to be *magic*: it only asks us to cover the target density with our proposal distribution!
- There is no “free lunch”: a poor choice of proposal distribution results in bad approximations (i.e. *bad mixing*).

(figure from Andrieu et al. '03)



- **This danger increases (exponentially) as the data dimensionality increases.**

Revisiting Gibbs Sampling

- Recall the Gibbs sampling we saw in Lecture 4.
 1. Select $k \in \{1, \dots, n\}$ from a uniform distribution.
 2. Set $x_j^{(n+1)} = x_j^{(n)}$ for $j \neq k$.
 3. Sample $x_k^{(n+1)}$ from $p(x_k \mid x_{-k}^{(n)})$.
- Q: Is this a “correct” algorithm? i.e, can we guarantee that samples from this algorithm come from $p(\mathbf{x})$?
 - Consider $K(y, x) = \begin{cases} p(y_k \mid x_{-k}) & \text{if } y_{-k} = x_{-k} \\ 0 & \text{otherwise.} \end{cases}$
- If $p(\cdot \mid x_{-k}) > 0$ then the resulting graph for K is connected
⇒ its resulting Markov Chain is irreducible and aperiodic.
- Does it satisfy detailed balance wrt $p(\mathbf{x})$?

Revisiting Gibbs Sampling

Lemma: The kernel K satisfies detailed balance wrt $p(x)$.

Revisiting Gibbs Sampling

Lemma: The kernel K satisfies detailed balance wrt $p(x)$.

We need to show that $p(x)K(x, x') = p(x')K(x', x)$.

Suppose that $x \neq x'$ and they differ in exactly one position k .

$$\begin{aligned} p(x)K(x, x') &= \frac{1}{n}p(x)p(x'_k \mid x_{-k}) \\ &= \frac{1}{n}p(x_k \mid x'_{-k})p(x'_{-k})p(x'_k \mid x'_{-k}) \\ &= \frac{1}{n}p(x_k \mid x'_{-k})p(x') \\ &= p(x')K(x', x) . \end{aligned}$$

Therefore, here $K = P$ (we accept the sample with probability 1).

Mixing Time in MCMC

- We have seen that the correctness of MCMC relies on a Markov chain having the appropriate stationary distribution.

Mixing Time in MCMC

- We have seen that the correctness of MCMC relies on a Markov chain having the appropriate stationary distribution.
- We have also seen that this stationary distribution is associated with the leading eigenvector of its kernel.
i.e, given initial guess μ , $\mu P^t \rightarrow \pi$ as $t \rightarrow \infty$.

Mixing Time in MCMC

- We have seen that the correctness of MCMC relies on a Markov chain having the appropriate stationary distribution.
- We have also seen that this stationary distribution is associated with the leading eigenvector of its kernel.
 - i.e, given initial guess μ , $\mu P^t \rightarrow \pi$ as $t \rightarrow \infty$.
- How fast does the chain converge to its leading eigenvector?
 - i.e. how small $\|\mu P^t - \pi\|$ is as $t \rightarrow \infty$?

Mixing Time in MCMC

- We have seen that the correctness of MCMC relies on a Markov chain having the appropriate stationary distribution.
- We have also seen that this stationary distribution is associated with the leading eigenvector of its kernel.
 - i.e, given initial guess μ , $\mu P^t \rightarrow \pi$ as $t \rightarrow \infty$.
- How fast does the chain converge to its leading eigenvector?
 - i.e. how small $\|\mu P^t - \pi\|$ is as $t \rightarrow \infty$?
- The convergence is dictated by the second largest eigenvalue:

$$\|\mu P^t - \pi\|_{TV} \leq \lambda_2^t \sqrt{\frac{1}{\min_j \pi_j}} .$$

Mixing Time in MCMC

- We have seen that the correctness of MCMC relies on a Markov chain having the appropriate stationary distribution.
- We have also seen that this stationary distribution is associated with the leading eigenvector of its kernel.

i.e, given initial guess μ , $\mu P^t \rightarrow \pi$ as $t \rightarrow \infty$.

- How fast does the chain converge to its leading eigenvector?
i.e. how small $\|\mu P^t - \pi\|$ is as $t \rightarrow \infty$?
- The convergence is dictated by the second largest eigenvalue:

$$\|\mu P^t - \pi\|_{TV} \leq \lambda_2^t \sqrt{\frac{1}{\min_j \pi_j}} .$$

– Second largest eigenvalues can be bounded using the Cheeger bound:

$$\lambda_2 \leq 1 - \frac{\Phi^2}{2} , \text{ where}$$

Φ is the conductance of P :

$$\min_{\Omega' \subset \Omega} \frac{\sum_{i \in \Omega', j \notin \Omega'} \pi_i P(i, j)}{\pi(\Omega')(1 - \pi(\Omega'))} .$$

Interlude: the Stein Method

- More generally, how can we evaluate whether a sample $\{x_1, \dots, x_n\}$ is legitimate for a certain $p(x)$, possibly known up to normalization?

Interlude: the Stein Method

- More generally, how can we evaluate whether a sample $\{x_1, \dots, x_n\}$ is legitimate for a certain $p(x)$, possibly known up to normalization?
- The *Stein method* (70') is a powerful tool to control the distance between two probability distributions using a metric of the form

$$d(P, Q) = \sup_{h \in \mathcal{H}} |\mathbb{E}_{X \sim P} h(X) - \mathbb{E}_{Y \sim Q} h(Y)| ,$$

for a given family of functions $\mathcal{H} = \{h : \Omega \rightarrow \mathbb{R}\}$.

Interlude: the Stein Method

- More generally, how can we evaluate whether a sample $\{x_1, \dots, x_n\}$ is legitimate for a certain $p(x)$, possibly known up to normalization?
- The *Stein method* (70') is a powerful tool to control the distance between two probability distributions using a metric of the form

$$d(P, Q) = \sup_{h \in \mathcal{H}} |\mathbb{E}_{X \sim P} h(X) - \mathbb{E}_{Y \sim Q} h(Y)| ,$$

for a given family of functions $\mathcal{H} = \{h : \Omega \rightarrow \mathbb{R}\}$.

- If we know Q , we know how its moments $E_{Y \sim f(Q)}$ behave for different f .

Stein's method reverses this: can we characterize the distribution by finding functional relationships between moments?

Interlude: the Stein Method

Suppose that Q is fixed, known distribution.

We look for an operator \mathcal{A} acting on functions $f : \Omega \rightarrow \mathbb{R}$ such that

$$\mathbb{E}_Y\{(\mathcal{A}f)Y\} = 0 \text{ for all } f \in \mathcal{F} \Leftrightarrow Y \text{ has distribution } Q.$$

Interlude: the Stein Method

Suppose that Q is fixed, known distribution.

We look for an operator \mathcal{A} acting on functions $f : \Omega \rightarrow \mathbb{R}$ such that

$$\mathbb{E}_Y\{(\mathcal{A}f)Y\} = 0 \text{ for all } f \in \mathcal{F} \Leftrightarrow Y \text{ has distribution } Q.$$

Ex: If Q is a standard normal $\mathcal{N}(0, 1)$, we have

$$\mathbb{E}\{f'(Y) - Yf(Y)\} = 0 \text{ for all } f \in C^1 \Leftrightarrow Y \sim \mathcal{N}(0, 1) .$$

(Stein's lemma)

So we can consider $(\mathcal{A}f)(x) = f'(x) - xf(x)$ in that case.

Interlude: the Stein Method

Suppose that Q is fixed, known distribution.

We look for an operator \mathcal{A} acting on functions $f : \Omega \rightarrow \mathbb{R}$ such that

$$\mathbb{E}_Y\{(\mathcal{A}f)Y\} = 0 \text{ for all } f \in \mathcal{F} \Leftrightarrow Y \text{ has distribution } Q.$$

Ex: If Q is a standard normal $\mathcal{N}(0, 1)$, we have

$$\mathbb{E}\{f'(Y) - Yf(Y)\} = 0 \text{ for all } f \in C^1 \Leftrightarrow Y \sim \mathcal{N}(0, 1) .$$

(Stein's lemma)

So we can consider $(\mathcal{A}f)(x) = f'(x) - xf(x)$ in that case.

Thus, if $P = Q$, then $\mathbb{E}_{X \sim P}\{(\mathcal{A}f)(X)\} = 0$ for all $f \in C^1$.

Hope: if $P \approx Q$, then $\mathbb{E}_{X \sim P}\{(\mathcal{A}f)(X)\} \approx 0$ for all $f \in C^1$.

Interlude: the Stein Method

Given a test function h , in some cases we can find $f = f_h$ such that

$$(\mathcal{A}f)(x) = h(x) - \mathbb{E}_{Y \sim Q}(h(Y)) .$$

Interlude: the Stein Method

Given a test function h , in some cases we can find $f = f_h$ such that

$$(\mathcal{A}f)(x) = h(x) - \mathbb{E}_{Y \sim Q}(h(Y)) .$$

It results that

$$\mathbb{E}_{W \sim P}\{(\mathcal{A}f)W\} = \mathbb{E}_{W \sim P}\{h(W)\} - \mathbb{E}_{Y \sim P}\{h(Y)\} .$$

So we can control $\text{dist}(P, Q)$ by bounding the functional form on the lhs.

When is bounding lhs easier than bounding rhs?

Interlude: the Stein Method

Given a test function h , in some cases we can find $f = f_h$ such that

$$(\mathcal{A}f)(x) = h(x) - \mathbb{E}_{Y \sim Q}(h(Y)) .$$

It results that

$$\mathbb{E}_{W \sim P}\{(\mathcal{A}f)W\} = \mathbb{E}_{W \sim P}\{h(W)\} - \mathbb{E}_{Y \sim P}\{h(Y)\} .$$

So we can control $\text{dist}(P, Q)$ by bounding the functional form on the lhs.

When is bounding lhs easier than bounding rhs?

Two very important special cases:

- Gaussian Case: $(\mathcal{A}f)(x) = f'(x) - xf(x)$, with
$$f(x) = e^{x^2/2} \int_{-\infty}^x [h(s) - \mathbb{E}(h(Y))]e^{-s^2/2} ds.$$
- Markov case: $(\mathcal{A}f)(x) = \langle f(x), \nabla \log p(x) \rangle + (\text{div } f)(x)$.

Crucial point: The Stein operator does not depend upon normalizing constant Z !

Evaluating sample quality

- Gorham and Mackey (NIPS'15) exploit the Stein operator to evaluate samples out of any MCMC algorithm.
 - Algorithm solves a linear program that bounds

$$\sup_{f \in \mathcal{H}} \sum_{i \leq n} (\langle f(x_i), \nabla \log p(x_i) \rangle + \operatorname{div} f(x_i)) .$$

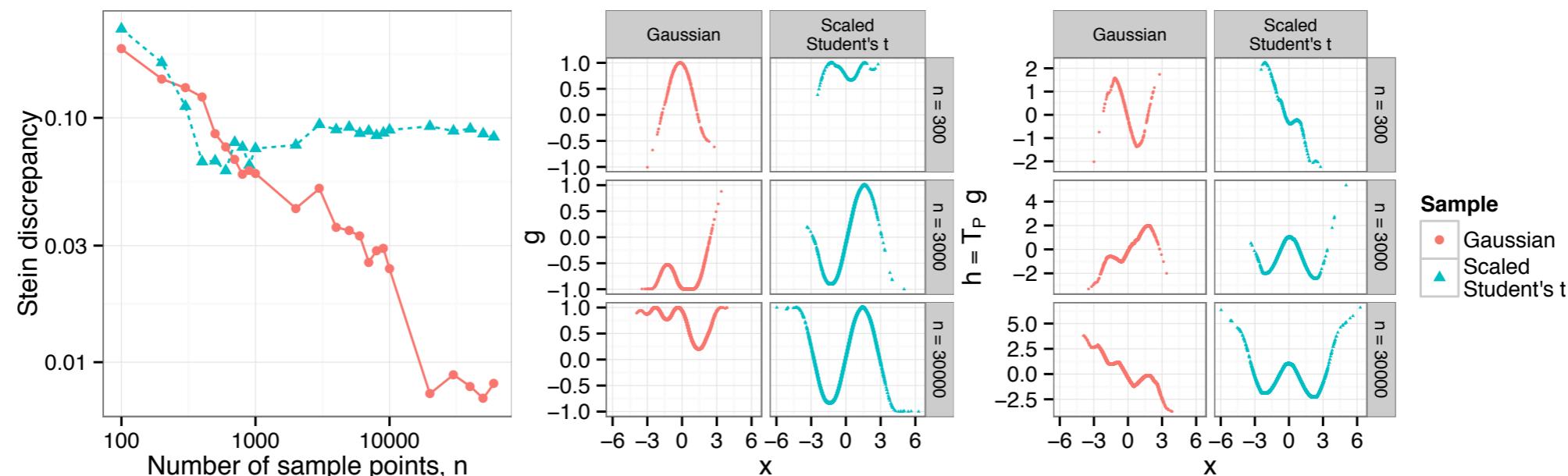


Figure 1: Left: Complete graph Stein discrepancy for a $\mathcal{N}(0, 1)$ target. Middle / right: Optimal Stein functions g and discriminating test functions $h = \mathcal{T}_P g$ recovered by the Stein program.

Evaluating sample quality

- Gorham and Mackey (NIPS'15) exploit the Stein operator to evaluate samples out of any MCMC algorithm.
 - Algorithm solves a linear program that bounds

$$\sup_{f \in \mathcal{H}} \sum_{i \leq n} (\langle f(x_i), \nabla \log p(x_i) \rangle + \operatorname{div} f(x_i)) .$$

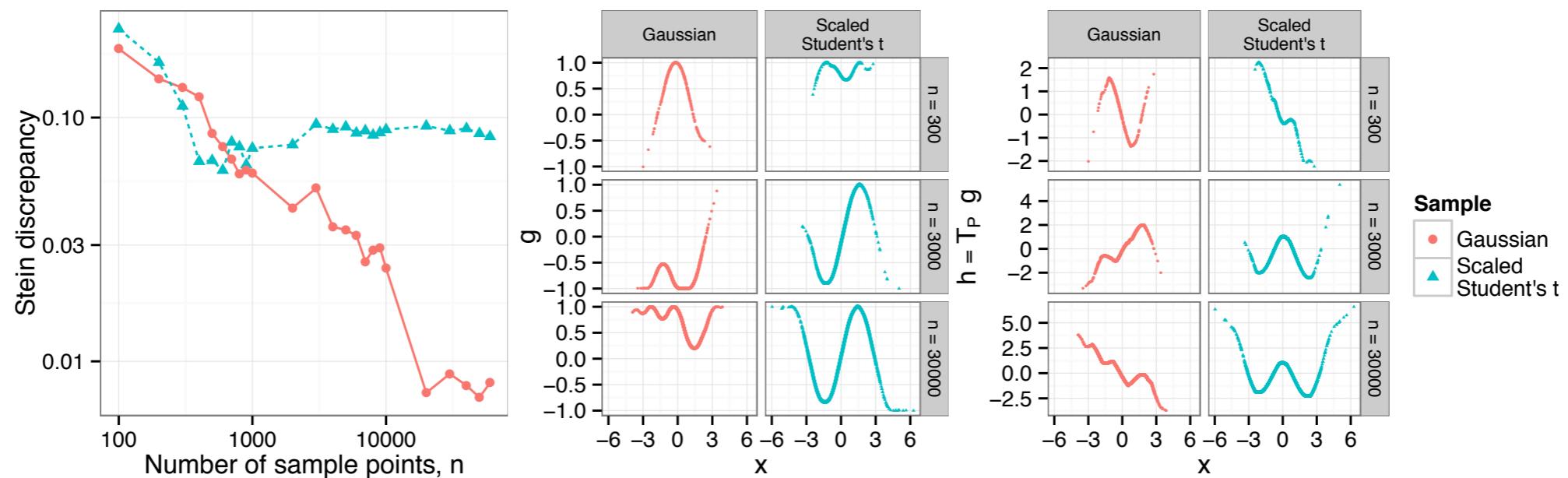


Figure 1: Left: Complete graph Stein discrepancy for a $\mathcal{N}(0, 1)$ target. Middle / right: Optimal Stein functions g and discriminating test functions $h = T_P g$ recovered by the Stein program.

- No assumption on the origin of the samples.
- Similar to Maximum-Mean Discrepancy (MMD) but Stein's method operates under weaker assumptions and is more efficient.

MCMC Extensions

- Hamiltonian Monte-Carlo (HMC) [Duane et al.'87][Neal'11]

Embeds a Gibbs distribution of the form $p(x) = \frac{e^{-T^{-1}U(x)}}{Z}$ into

$$p(x, y) = \frac{e^{-T^{-1}(U(x)+K(y))}}{Z}, \quad \begin{array}{l} x: \text{position} \\ y: \text{momentum} \end{array}$$

$U(x)$: potential energy $K(y)$: kinetic energy

$K(y) \propto \|y\|^2 \Rightarrow p(y|x)$ is Gaussian .

- Langevin Dynamics.