Inference and Representation

DS-GA-1005, CSCI-GA.2569

Joan Bruna
Courant Institute of Mathematical Sciences
Center for Data Science
NYU

Lecture 5 Announcements

- Project Proposal:

- Today:
  - PCA and Factor Analysis
  - Markov Chains

Principal Component Analysis

Reminder: [Real Spectral Theorem]: If $A \in \mathbb{R}^{n \times n}$ is symmetric and positive semidefinite ($A \succeq 0$), then A diagonalizes
in a real orthonormal basis with non-negative eigenvalues.

Principal Component Analysis

Reminder: [Real Spectral Theorem]: If $A \in \mathbb{R}^{n \times n}$ is symmetric and positive semidefinite ($A \succeq 0$), then A diagonalizes
in a real orthonormal basis with non-negative eigenvalues.

Fact: The covariance operator $\Sigma_X = \mathbb{E}((X - \mu_X)(X - \mu_X)^T)$ is **symmetric** and **positive semidefinite**.

Principal Component Analysis

Reminder: [Real Spectral Theorem]: If $A \in \mathbb{R}^{n \times n}$ is symmetric and positive semidefinite ($A \succeq 0$), then A diagonalizes in a real orthonormal basis with non-negative eigenvalues.

Fact: The covariance operator $\Sigma_X = \mathbb{E}((X - \mu_X)(X - \mu_X)^T)$ is **symmetric** and **positive semidefinite**.

Therefore, Σ_X admits a real eigenbasis:

$$\Sigma_X = U \Lambda U^T \ , \ \Lambda = \mathrm{diag}(\lambda_i) \succeq 0 \ .$$

Thus $U^T \Sigma_X U = \Lambda$.

Principal Component Analysis

Reminder: [Real Spectral Theorem]: If $A \in \mathbb{R}^{n \times n}$ is symmetric and positive semidefinite ($A \succeq 0$), then A diagonalizes in a real orthonormal basis with non-negative eigenvalues.

Fact: The covariance operator $\Sigma_X = \mathbb{E}((X - \mu_X)(X - \mu_X)^T)$ is **symmetric** and **positive semidefinite**.

Therefore, Σ_X admits a real eigenbasis:

$$\Sigma_X = U \Lambda U^T \ , \ \Lambda = \mathrm{diag}(\lambda_i) \succeq 0 \ .$$

Thus $U^T \Sigma_X U = \Lambda$.

Moreover, we can write $\Lambda = S \cdot S$, with $s_{i,i} = \sqrt{\lambda_i}$.
If $\min_i \lambda_i > 0$, it results that $\tilde{U} = U S^{-1}$ satisfies $\tilde{U}^T \Sigma_X \tilde{U} = \mathbf{1}$.

Principal Component Analysis

- We have just shown that $\mathbf{Y} = \tilde{U}(\mathbf{X} - \mu_X)$ provides variables that are uncorrelated and jointly Gaussian, thus independent.

Principal Component Analysis

- We have just shown that $\mathbf{Y} = \tilde{U}(\mathbf{X} - \mu_X)$ provides variables that are uncorrelated and jointly Gaussian, thus independent.

- Remarks
  - The decomposition is not unique: Any orthogonal transformation of Y also satisfies the same property.
  - PCA provides linear compression: if $J < \mathbf{rank}(\Sigma_X)$, what is the best linear approximation of X with J independent components?

$$\min_{A \in \mathbb{R}^{L \times J}} \mathbb{E}(\|X - AX\|^2) \ .$$

Principal Component Analysis

- We have just shown that $\mathbf{Y} = \tilde{U}(\mathbf{X} - \mu_X)$ provides variables that are uncorrelated and jointly Gaussian, thus independent.

- Remarks
  - The decomposition is not unique: Any orthogonal transformation of Y also satisfies the same property.
  - PCA provides linear compression: if $J < \mathbf{rank}(\Sigma_X)$, what is the best linear approximation of X with J independent components?

$$\min_{A \in \mathbb{R}^{L \times J}} \mathbb{E}(\|X - AX\|^2) \ .$$

$$A = \{ \text{ eigenvectors of } \Sigma_X \text{ corresponding to } J \text{ largest eigenvalues.} \}$$

(again, A is determined up to an orthogonal transformation)

Estimating the Principal Components

- So far, we have seen how to extract information from the covariance of X.

Estimating the Principal Components

- So far, we have seen how to extract information from the covariance of X.
- In practice, we will observe $x_1, \ldots, x_N$ iid samples of X
- Empirical Covariance:

$$\hat{\Sigma}_N = \frac{1}{N} \sum_{n<N} (x_n - \hat{\mu})(x_n - \hat{\mu})^T \ . \quad \in \mathbb{R}^{L \times L} \ .$$

Estimating the Principal Components

- So far, we have seen how to extract information from the covariance of X.
- In practice, we will observe $x_1, \ldots, x_N$ iid samples of X
- Empirical Covariance:

$$\widehat{\Sigma}_N = \frac{1}{N} \sum_{n<N} (x_n - \hat{\mu})(x_n - \hat{\mu})^T \ . \quad \in \mathbb{R}^{L \times L} \ .$$

- $\widehat{\Sigma}_N$ is symmetric, positive definite. (why?)

- Estimated Principal Components:

$$\widehat{\Sigma}_N = \widehat{U} \widehat{\Lambda} \widehat{U}^T \ .$$

Estimating the Principal Components

- Q: How good are these estimates?
  - i.e. for a desired accuracy ϵ, how many samples N=N(L) are required?

Estimating the Principal Components

- Q: How good are these estimates?
  - i.e. for a desired accuracy ϵ, how many samples N=N(L) are required?

- **Theorem** [Vershynin]: For distributions with bounded order-q moment, the empirical covariance satisfies

$$\|\widehat{\Sigma}_N - \Sigma\| \lesssim O(\log\log L)^2 \left(\frac{L}{N}\right)^{1/2-2/q} \qquad (q > 4)$$

Estimating the Principal Components

- Q: How good are these estimates?
  - i.e. for a desired accuracy ϵ, how many samples N=N(L) are required?

- **Theorem** [Vershynin]: For distributions with bounded order-q moment, the empirical covariance satisfies

$$\|\widehat{\Sigma}_N - \Sigma\| \lesssim O(\log\log L)^2 \left(\frac{L}{N}\right)^{1/2-2/q} \qquad (q > 4)$$

It results that for a desired approximation $\|\widehat{\Sigma}_N - \Sigma\| \leq \epsilon$ we need $O((\log\log L)^\alpha L) \approx O(L)$ samples.

Estimating the Principal Components

- Q: How good are these estimates?
  - i.e. for a desired accuracy ϵ, how many samples N=N(L) are required?

- **Theorem** [Vershynin]: For distributions with bounded order-q moment, the empirical covariance satisfies $(q > 4)$

$$\|\widehat{\Sigma}_N - \Sigma\| \lesssim O(\log\log L)^2 \left(\frac{L}{N}\right)^{1/2-2/q}.$$

- It results that for a desired approximation $\|\widehat{\Sigma}_N - \Sigma\| \leq \epsilon$ we need $O((\log\log L)^\alpha L) \approx O(L)$ samples. $\left(\frac{1}{\alpha} + \frac{1}{q} = \frac{1}{4}\right)$

- **Very Important Consequence: PCA does not suffer from the curse of dimensionality!**

Estimating Principal Components

- Q: What is the computational complexity of computing PCA?
  - Naïve estimation of covariance: $O(NL^2)$
  - Diagonalizing covariance: $O(L^3)$
  - So, in big data applications we are doomed.

Estimating Principal Components

- Q: What is the computational complexity of computing PCA?
  - Naïve estimation of covariance: $O(NL^2)$
  - Diagonalizing covariance: $O(L^3)$
  - So, in big data applications we are doomed.

- Reminder: Given a data matrix, the principal components are directly given by

The **Singular Value Decomposition** (SVD) of $B \in \mathbb{R}^{n \times p}$ is defined as $B = U \Lambda V^T$, with

$$U \in \mathbb{R}^{n \times n}, V \in \mathbb{R}^{p \times p}, \Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_{\min(n,p)}),$$
$$UU^T = \mathbf{1}, V^T V = \mathbf{1} \ .$$

Computing the first p principal components costs $O(pNL)$.

Estimating Principal Components

- Q: What is the computational complexity of computing PCA?
  - Naïve estimation of covariance: $O(NL^2)$
  - Diagonalizing covariance: $O(L^3)$
  - So, in big data applications we are doomed.

- Reminder: Given a data matrix, the principal components are directly given by

The **Singular Value Decomposition** (SVD) of $B \in \mathbb{R}^{n \times p}$ is defined as $B = U \Lambda V^T$, with

$$U \in \mathbb{R}^{n \times n}, V \in \mathbb{R}^{p \times p}, \Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_{\min(n,p)}),$$
$$UU^T = \mathbf{1}, V^T V = \mathbf{1} \ .$$

Computing the first p principal components costs $O(pNL)$.

- Alternatives?

Interlude: Randomized PCA

- Suppose we suspect that the data matrix $X \in \mathbb{R}^{N \times L}$ has rank $p \ll \min(N, L)$.
- Q: Can we leverage that prior into a faster algorithm?

Interlude: Randomized PCA

- Suppose we suspect that the data matrix $X \in \mathbb{R}^{N \times L}$ has rank $p \ll \min(N, L)$.
- Q: Can we leverage that prior into a faster algorithm?

- **Idea**: Break the estimation into two steps:
  1. Compute an approximate basis for the range of X:

$$X \approx QQ^T X \ , \ \text{with } Q \in \mathbb{R}^{N \times n} \ , \ n \ll N, \ Q^T Q = \mathbf{1} \ .$$

  2. Form $B = Q^T X \in \mathbb{R}^{n \times L}$ and compute its SVD: $B = \tilde{U} \Lambda V^T$.
  3. Set $U = Q\tilde{U}$.

- How to solve stage 1?

Interlude: Randomized PCA

- Suppose we suspect that the data matrix $X \in \mathbb{R}^{N \times L}$ has rank $p \ll \min(N, L)$.
- Q: Can we leverage that prior into a faster algorithm?

- **Idea**: Break the estimation into two steps:
  1. Compute an approximate basis for the range of X:

$$X \approx QQ^T X \ , \ \text{with } Q \in \mathbb{R}^{N \times n} \ , \ n \ll N, \ Q^T Q = \mathbf{1} \ .$$

  2. Form $B = Q^T X \in \mathbb{R}^{n \times L}$ and compute its SVD: $B = \tilde{U} \Lambda V^T$.
  3. Set $U = Q\tilde{U}$.

- How to solve stage 1? **Randomize!!**

Interlude: Randomized PCA

- If we target a rank-k approximation, we simply draw a random matrix $\Omega \in \mathbb{R}^{L \times (k+p)}$, form the matrix $Y = X\Omega \in \mathbb{R}^{N \times (k+p)}$, and perform a SVD of Y: $Y = V \Lambda Q^T$.

Interlude: Randomized PCA

- If we target a rank-k approximation, we simply draw a random matrix $\Omega \in \mathbb{R}^{L \times (k+p)}$, form the matrix $Y = X\Omega \in \mathbb{R}^{N \times (k+p)}$, and perform a SVD of Y: $Y = V \Lambda Q^T$.

- Strong guarantees from concentration of measure:

Theorem: [Halko, Martinsson, Tropp] Given data matrix $X \in \mathbb{R}^{N \times L}$ and $\Omega \in \mathbb{R}^{L \times (k+p)}$ drawn from iid standard Gaussian, the resulting Q satisfies

$$\|X - QQ^T X\| \leq \left(1 + C\sqrt{(k+p)\min(N,L)}\right) \lambda_{k+1} \ ,$$

whp, where λ_{k+1} is the $k+1$-th singular value of X.

$(1 - cp^{-p})$

Interlude: Randomized PCA

- If we target a rank-k approximation, we simply draw a random matrix $\Omega \in \mathbb{R}^{L\times(k+p)}$, form the matrix $Y = X\Omega \in \mathbb{R}^{N\times(k+p)}$, and perform a SVD of Y: $Y = V\Lambda Q^T$.

- Strong guarantees from concentration of measure:

Theorem: [Halko, Martinsson, Tropp] Given data matrix $X \in \mathbb{R}^{N\times L}$ and $\Omega \in \mathbb{R}^{L\times(k+p)}$ drawn from iid standard Gaussian, the resulting Q satisfies

$$\|X - QQ^T X\| \leq \left(1 + C\sqrt{(k+p)\min(N,L)}\right)\lambda_{k+1} \ ,$$

whp, where λ_{k+1} is the $k+1$-th singular value of X.

$(1 - cp^{-p})$

- Resulting computational gains:

from $O(NLk)$ to $O(NL\log(k))$ for k ppal components.

Factor Analysis

- We have seen that PCA can be interpreted as a linear latent model:

$$X_i = \sum_j \alpha_{i,j} Y_j + \mu_i \ , \ (i = 1, \ldots L) \ ,$$

with Y_j uncorrelated, unit variance.

Factor Analysis

- We have seen that PCA can be interpreted as a linear latent model:

$$X_i = \sum_j \alpha_{i,j} Y_j + \mu_i \ , \ (i = 1, \dots L) \ ,$$

with Y_j uncorrelated, unit variance.

- Lack of unicity: given an $L \times L$ orthogonal matrix R, we also have

$$X = \mu + AR^T RY = \mu + \tilde{A}\tilde{Y} \ ,$$

where $\tilde{Y}_j$ are also uncorrelated and unit variance.

Factor Analysis

- We have seen that PCA can be interpreted as a linear latent model:

$$X_i = \sum_j \alpha_{i,j} Y_j + \mu_i \ , \ (i = 1, \dots L) \ ,$$

with Y_j uncorrelated, unit variance.

- Lack of unicity: given an $L \times L$ orthogonal matrix R, we also have

$$X = \mu + AR^T RY = \mu + \tilde{A}\tilde{Y} \ ,$$

where $\tilde{Y}_j$ are also uncorrelated and unit variance.

- Also, an underlying assumption is that data has *low-rank,* i.e. covariance directly reveals dependencies in data.

Factor Analysis

- An alternative to PCA is *Factor Analysis*. Suppose a generative model of the form

$$X_i = \sum_{j \leq J} \alpha_{i,j} Y_j + \mu_i \ + \epsilon_i \ , \ (i = 1, \ldots L) \ ,$$

with $J < L$ and ϵ_i uncorrelated, zero-mean.

Factor Analysis

- An alternative to PCA is *Factor Analysis*. Suppose a generative model of the form

$$X_i = \sum_{j \leq J} \alpha_{i,j} Y_j + \mu_i \ + \epsilon_i \ , \ (i = 1, \ldots L) \ ,$$

with $J < L$ and ϵ_i uncorrelated, zero-mean.

- Interpretation:
  - Latent variables Y_j are common factors of variability.
  - Latent variables ϵ_i explain the remaining individual variability, uncorrelated from the rest.
- Example:
  - Factor analysis on the topics of your final project.

Factor Analysis

- Gaussian joint likelihood model:

$$X \sim \mathcal{N}(\mu, AA^T + \mathrm{diag}(\beta))$$

  - with $\beta_i = \mathbf{Var}(\epsilon_i)$.
  - Parameter Estimation? The covariance is a sufficient statistic:

$$\Sigma_X = AA^T + \mathrm{diag}(\beta) \ .$$

$$\uparrow$$

$$\text{low rank}$$

  - SVD is still useful, but does not automatically yield the solution.
  - We will soon see an alternative estimation algorithm (EM).

Independent Component Analysis

- We have seen that asking latent variables only to be decorrelated leads to lack of unicity (any orthogonal transformation is also decorrelated).

Independent Component Analysis

- We have seen that asking latent variables only to be decorrelated leads to lack of unicity (any orthogonal transformation is also decorrelated).
- What happens if we tighten our requirements, e.g. asking that

$$Y_i \text{ and } Y_j \text{ independent}$$

Independent Component Analysis

- We have seen that asking latent variables only to be decorrelated leads to lack of unicity (any orthogonal transformation is also decorrelated).
- What happens if we tighten our requirements, e.g. asking that

$$Y_i \text{ and } Y_j \text{ independent}$$

- We have seen that in the Gaussian case, this does not alleviate the problem. (why?)

Independent Component Analysis

- We have seen that asking latent variables only to be decorrelated leads to lack of unicity (any orthogonal transformation is also decorrelated).
- What happens if we tighten our requirements, e.g. asking that

$$Y_i \text{ and } Y_j \text{ independent}$$

- We have seen that in the Gaussian case, this does not alleviate the problem. (why?)

- But, as it turns out, it is the exception. The model becomes uniquely identifiable if

$$Y_i \text{ and } Y_j \text{ independent and non-Gaussian.}$$

Independent Component Analysis

- Without loss of generality, we can assume X has trivial covariance: $\Sigma_X = \mathbf{1}$.
- Since $\Sigma_Y = \mathbf{1}$ as well, it results that $\mathbf{1} = A\Sigma_Y A^T = AA^T$

Independent Component Analysis

- Without loss of generality, we can assume X has trivial covariance: $\Sigma_X = \mathbf{1}$.
- Since $\Sigma_Y = \mathbf{1}$ as well, it results that $\mathbf{1} = A\Sigma_Y A^T = AA^T$

- so we are reduced to finding an orthogonal transformation such that $Y = A^T X$ becomes independent and non-Gaussian.

Independent Component Analysis

- Without loss of generality, we can assume X has trivial covariance: $\Sigma_X = \mathbf{1}$.
- Since $\Sigma_Y = \mathbf{1}$ as well, it results that $\mathbf{1} = A\Sigma_Y A^T = AA^T$

- so we are reduced to finding an orthogonal transformation such that $Y = A^T X$ becomes independent and non-Gaussian.

- It is a form of "inverse" Central Limit Theorem method.

- Q: How to measure/estimate statistical independence?

Entropy

- For a continuous random variable X with density p(x), the *differential entropy* is

$$H(X) = -\int p(x) \log p(x) dx = -\mathbb{E}(\log p(X)) \ .$$

  – it extends the notion of uncertainty/information carried by X.

Entropy

- For a continuous random variable X with density p(x), the *differential entropy* is

$$H(X) = -\int p(x) \log p(x) dx = -\mathbb{E}(\log p(X)) \ .$$

  – it extends the notion of uncertainty/information carried by X.

- Entropy measures independence through the *mutual information*:

Given $X_1, \ldots, X_n$, the mutual information is

$$I(Y) = \sum_{i \leq n} H(Y_i) - H(Y) \ \geq 0 \ .$$

Entropy

- For a continuous random variable X with density p(x), the *differential entropy* is

$$H(X) = -\int p(x) \log p(x) dx = -\mathbb{E}(\log p(X)) \ .$$

  – it extends the notion of uncertainty/information carried by X.

- Entropy measures independence through the *mutual information*:

Given $X_1, \ldots, X_n$, the mutual information is

$$I(Y) = \sum_{i \leq n} H(Y_i) - H(Y) \ \geq 0 \ .$$

- **Fact:** $I(Y) = 0$ iff Y_i and Y_j are mutually independent.
- **Fact:** If A is unitary and $Y = A^T X$, then $H(Y) = H(X)$.

Independent Component Analysis

- So ICA attempts to solve the following problem:

$$\arg \min_{A^T A = \mathbf{1}} \sum_{i \leq L} H(\langle A_i, X \rangle) - H(X) = \arg \min_{A^T A = \mathbf{1}} \sum_{i \leq L} H(\langle A_i, X \rangle) .$$

- Ex from ESLL:

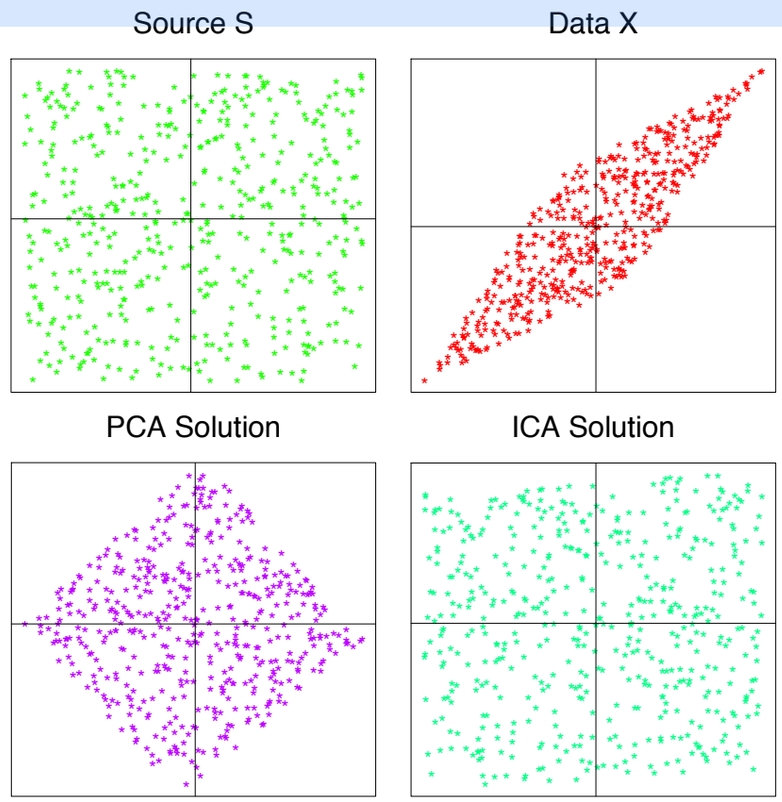


FIGURE 14.38. *Mixtures of independent uniform random variables. The upper left panel shows* 500 *realizations from the two independent uniform sources, the upper right panel their mixed versions. The lower two panels show the PCA and ICA solutions, respectively.*

- Challenge: computing entropy requires estimating the density: exposed to curse of dimensionality!

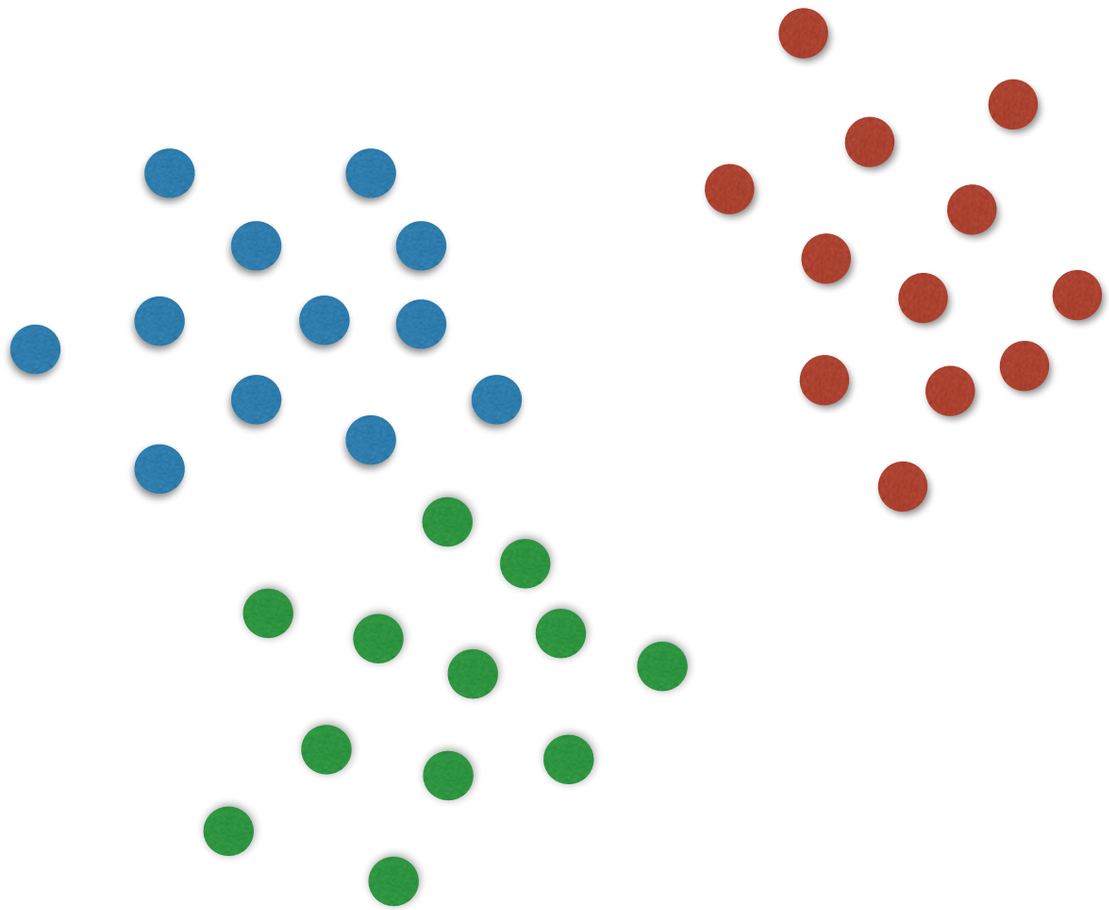
Explaining Away

- 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*.

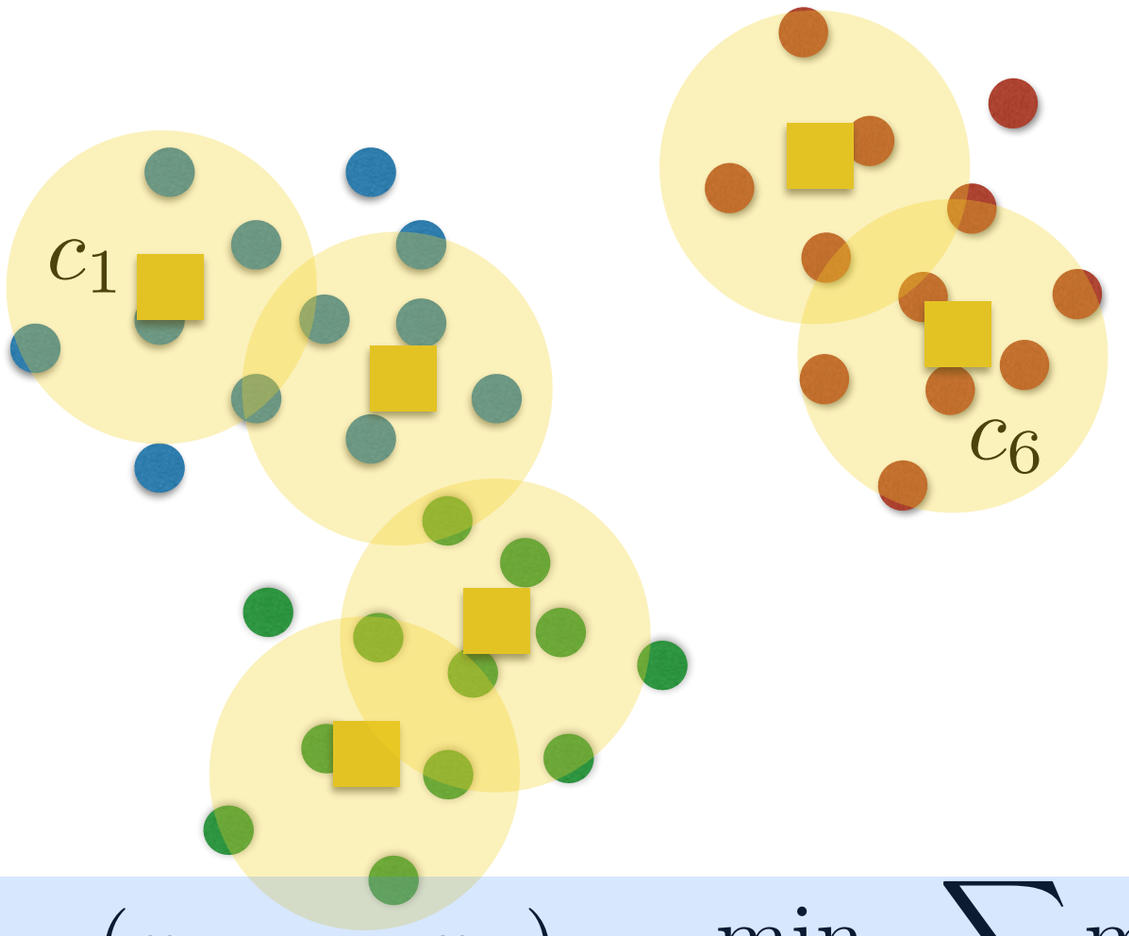
Latent Variables

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



Latent Variables

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



$K = 6$

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

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)

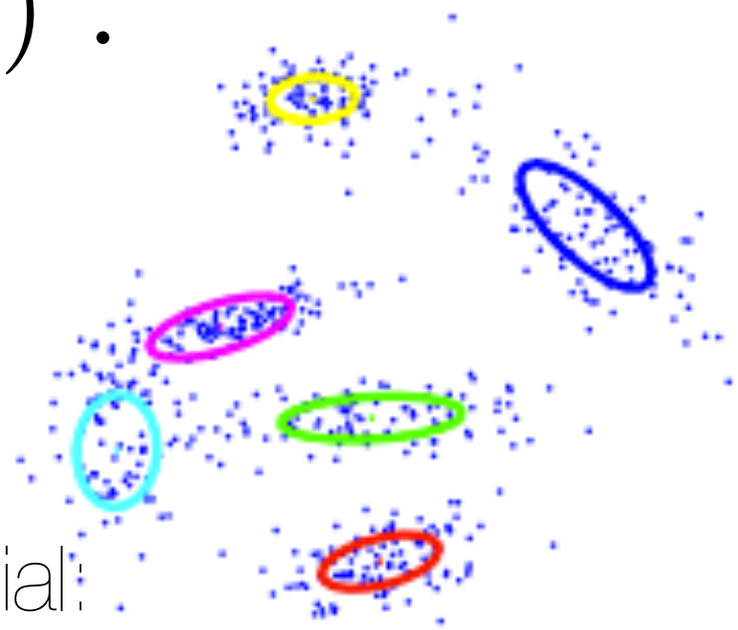
- A generalization of K-Means is given by a Gaussian Mixture:

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

- This is also a discrete latent variable model:

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

- The distribution of the latent variable is multinomial:



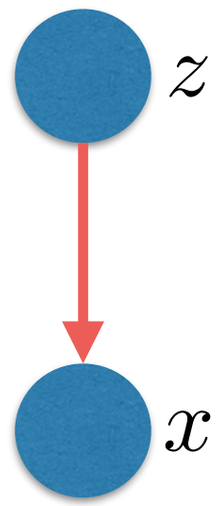
(figure from R.Salakhutdinov)

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

Gaussian Mixture Models (GMM)

- We can write

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



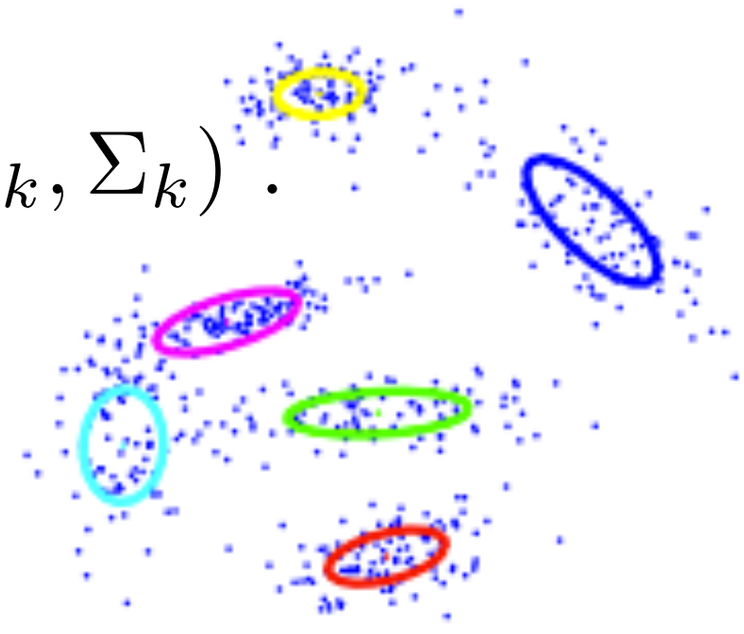
- Thus

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

- 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 | x) ? Ie, given data, which mixture components are "responsible" ?

$$p(z_k = 1 \mid x) = \frac{p(z_k = 1, x)}{\sum_{k' \leq K} p(z_{k'} = 1, x)} = \frac{p(z_k = 1)p(x \mid z_k = 1)}{\sum_{k' \leq K} p(z_{k'} = 1)p(x \mid z_{k'} = 1)}$$

$$= \frac{\pi_k \mathcal{N}(x; \mu_k, \Sigma_k)}{\sum_{k'} \pi_{k'} \mathcal{N}(x; \mu_{k'}, \Sigma_{k'})}$$

- 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 $X = \{x_1, \ldots, x_n\}$, the total log-likelihood is

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

Maximum Likelihood Estimation

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

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

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

- $$\mu_k = \frac{1}{N_k} \sum_i p(z_{i,k} = 1 \mid x_i) x_i \ , \ N_k = \sum_i p(z_{i,k} = 1 \mid x_i) \ .$$

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

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

The EM algorithm

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

$$\log p(X \mid \theta) = \log \left(\sum_{Z} p(X, Z \mid \theta) \right) \ , \ \theta = \text{model parameters} \ .$$
$$Z = \text{latent variables}$$

- 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:

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

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

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

$$\log p(X \mid \theta) = \log \left(\sum_Z p(X, Z \mid \theta) \right) = \log \left(\sum_Z q(Z) \frac{p(X, Z \mid \theta)}{q(Z)} \right)$$

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

(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

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

$$H(q) \text{: Entropy of } q(Z).$$

- Also, we have

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

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

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 $q(z) = p(z|x, \theta^{(n)})$:

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

$$= \sum_z p(z|x, \theta^{(n)}) \log \left(\frac{p(x|z, \theta)p(z|\theta)}{p(z|x, \theta^{(n)})p(x|\theta^{(n)}} \right)$$

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

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

Correctness of EM

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

$$\begin{aligned}
\theta^{(n+1)} &= \arg\max_{\theta} \Delta(\theta|\theta^{(n)}) \\
&= \arg\max_{\theta} \sum_{z} p(z|x,\theta^{(n)}) \log\{(p(x|z,\theta)p(z|\theta)\} \\
&= \arg\max_{\theta} \sum_{z} p(z|x,\theta^{(n)}) \log\{(p(x,z|\theta)\}\} \\
&= \arg\max_{\theta} \mathbb{E}_{z\sim p(z|x,\theta^{(n)})} \log p(X,Z|\theta) \ .
\end{aligned}$$

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

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

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

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

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

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

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

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

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

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

  - **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 $\mathbb{E}(f(X)) = \int p(x)f(x)dx$

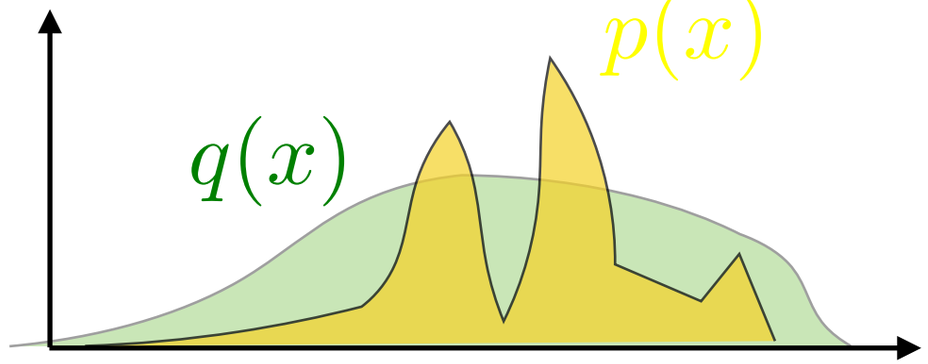
with samples from *another* distribution with density $q(x)$?

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

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

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

Importance Sampling

- So we can estimate the integral as

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

Importance Sampling

- So we can estimate the integral as

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

- Q: How good is this estimator?

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

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

Importance Sampling

- So we can estimate the integral as

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

- Q: How good is this estimator?

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

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

- 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.

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?

$$\left(p(x) = \frac{\tilde{p}(x)}{Z} \right)$$

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

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?

$$\left(p(x) = \frac{\tilde{p}(x)}{Z} \right)$$

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

So now we can estimate both numerator and denominator:

$$\widehat{\mathbb{E}_p\{f(x)\}} = \frac{\sum_{i \leq N} f(x^{(i)})w(x^{(i)})}{\sum_{i \leq N} w(x^{(i)})} = \sum_{i \leq N} f(x^{(i)})\tilde{w}(x^{(i)}) \ ,$$

$\tilde{w}(x^{(i)}) = \dfrac{w(x^{(i)})}{\sum_{j \leq N} w(x^{(j)})}$: 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.