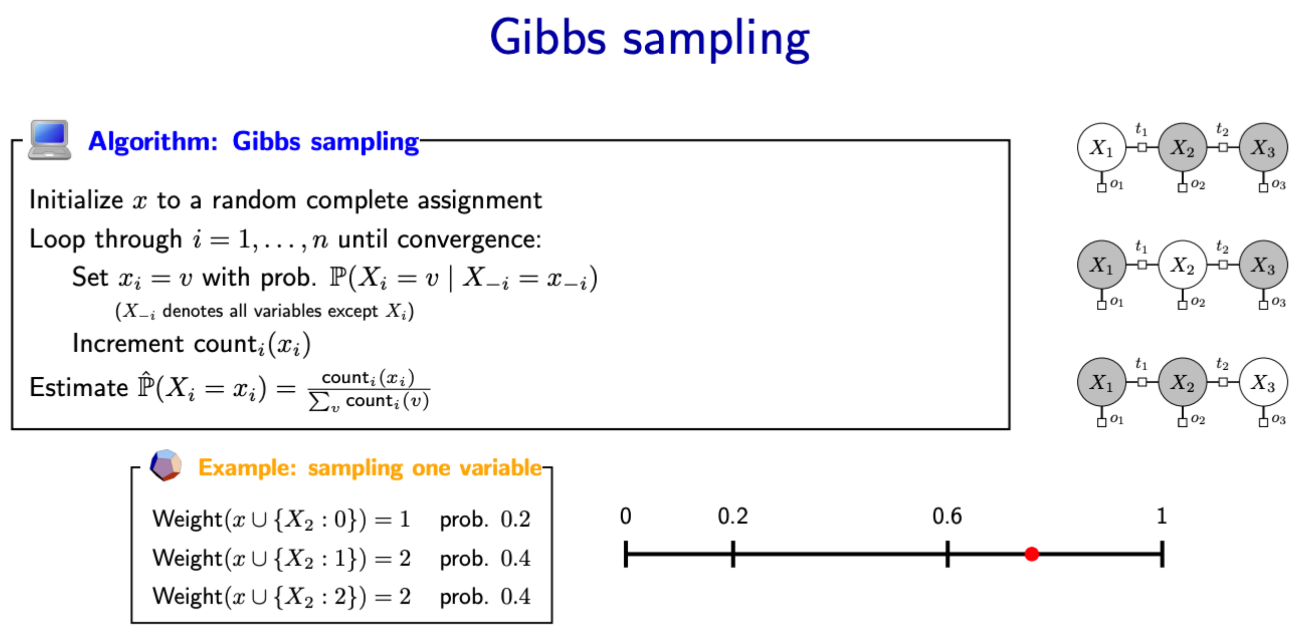
Gibbs Sampling

- 주변 확률을 대략적으로 계산하기 위한 간단한 알고리즘
- Gibbs Sampling은 한 번에 하나의 변수를 변경하는 local search 방식을 따르지만, Iterated Conditional Modes (ICM)과 달리 랜덤한 알고리즘임
- Markov Chain Monte Carlo; MCMC 알고리즘의 예시(M-H algorithm의 special case)
    - 다음에 생성될 표본은 현재 샘플에 영향을 받는다는 점에서 MCMC와 같지만, **나머지 변수는 그대로 두고 한 변수에만 변화를 준다는 점**이 다름



- Gibbs Sampling은 전체 중에서 일부 $v \in Domain_i$는 X_i의 모든 가능한 assignment를 고려하여 진행되며, 나머지 다른 모든 것들이 주어지면 $X_i = v$의 조건부 확률과 동일한 확률로 $X_i = v$를 설정하는 단계를 포함

- 이 단계를 수행하기 위해 확률 법칙을 이용하여 아래와 같이 나타낼 수 있음

- $$\mathbb{P}(X_i = v \mid X_{-i} = x_{-i}) = \frac{\text{Weight}(x \cup \{X_i : v\})}{Z\mathbb{P}(X_{-i} = x_{-i}}$$

- 각 v에 대해 $x \cup \{X_i : v\}$ 의 가중치를 계산한 다음 정규화하여 분포를 얻음

- 그 다음 해당 분포에 따라 v를 샘플링 진행

- 그 과정에서 추적하려는 각 변수 X_i에 대해 $X_i = v$를 본 횟수의 카운터 $count_i(v)$를 유지함

- Gibbs Sampling Example

    - 3개의 확률변수의 결합확률분포 $p(x1, x2, x3)$로부터 1개의 표본을 얻으려고 할 때

- 임의의 표본 $X^0 = (x_1^0, x_2^0, x_3^0)$ 을 선택
- 모든 변수에 대해 변수 하나만을 변경하여 새로운 표본 X^1 추출
    - 현재 주어진 표본 X^0의 두번째, 세번째 변수 x_2^0, x_3^0 고정
    - 첫번째 기존 변수 x_1^0를 대체할 새로운 값 x_1^1을 다음과 같은 확률로 추출 $p(x_1^1|x_2^0, x_3^0)$
    - 첫번째 변수 x_1^1, 세번째 변수 x_3^0 를 고정
    - 두번째 기존 변수 x_2^0를 대체할 새로운 값 x_2^1을 다음과 같은 확률로 새로 추출 $p(x_2^1|x_1^1, x_3^0)$
    - 첫번째 변수 x_1^1, 두번째 변수 x_2^1 를 고정
    - 세번째 기존 변수 x_3^0를 대체할 새로운 값 x_3^1을 다음과 같은 확률로 새로 추출 $p(x_3^1|x_1^1, x_2^1)$
    - 최종적으로 구한 X^1은 다음과 같음
        - $X^1 = (x_1^1, x_2^1, x_3^1)$

- Gibbs Sampling 예시

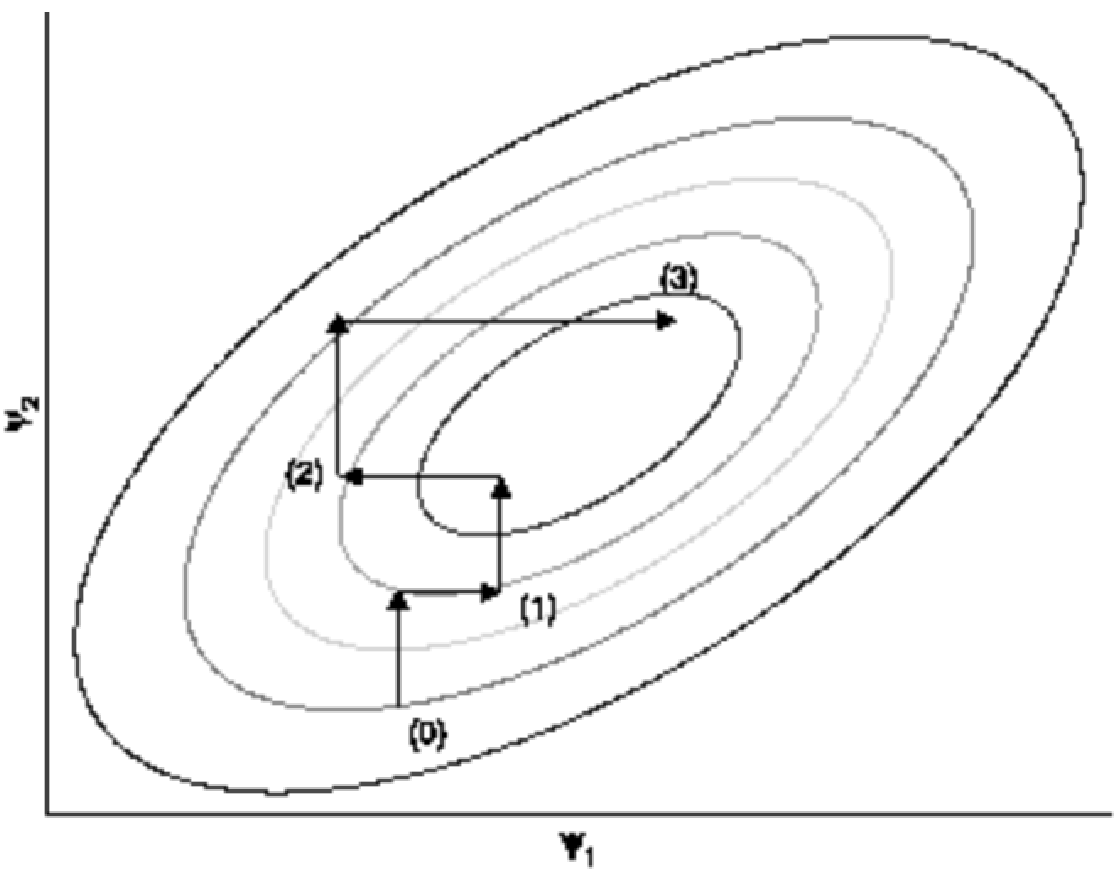
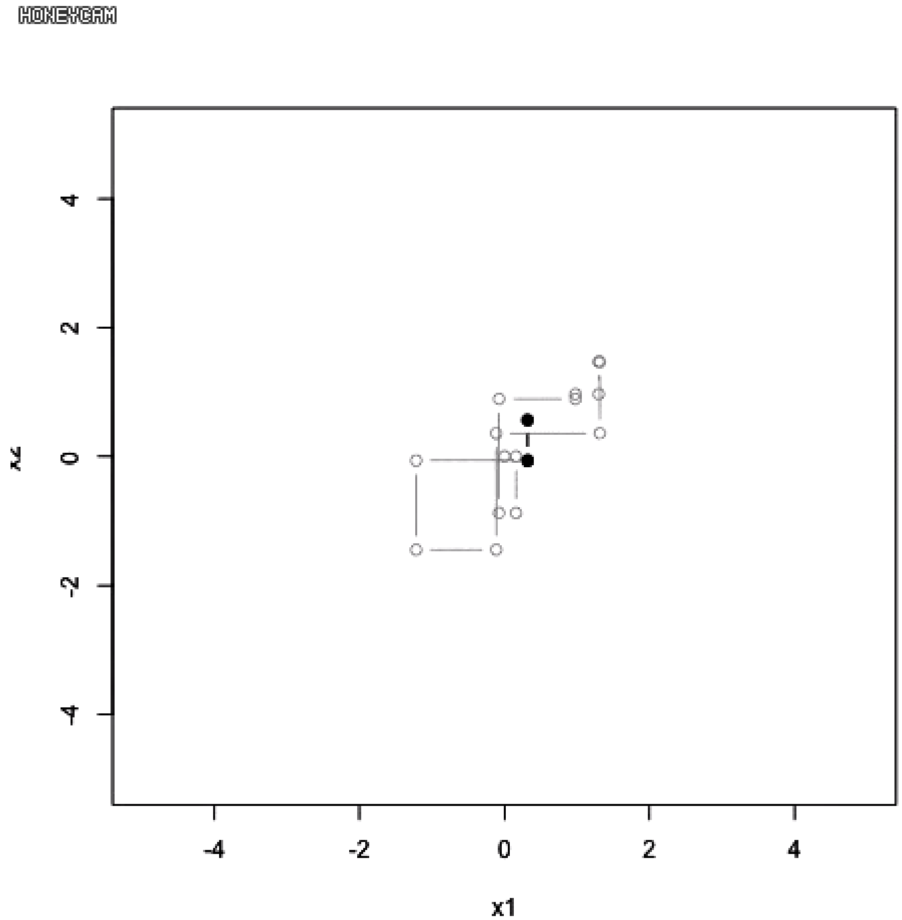


Figure 3.4: Gibbs sampling algorithm in two dimensions starting from an initial point and then completing three iterations

-



-

- Gibbs Sampling의 변형들
  - Block Gibbs Sampling : 그룹으로 묶어 뽑는 기법
  - Collapsed Gibbs Sampling : 불필요한 일부 변수를 샘플링에서 생략하는 기법

Gibbs Sampling을 이용한 Image Denoise

- Image Denoising이란
  - 이미지의 임의의 노이즈가 주어지거나 특정 측정 값이 누락된 노이즈가 있는 이미지가 주어졌을 때 denoise를 통해 원본 이미지를 복구하는 것
- Geman, Stuart, and Donald Geman. "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images." *IEEE Transactions on pattern analysis and machine intelligence* 6 (1984): 721-741.
  - S Geman 저술 · 1984 · 26083회 인용

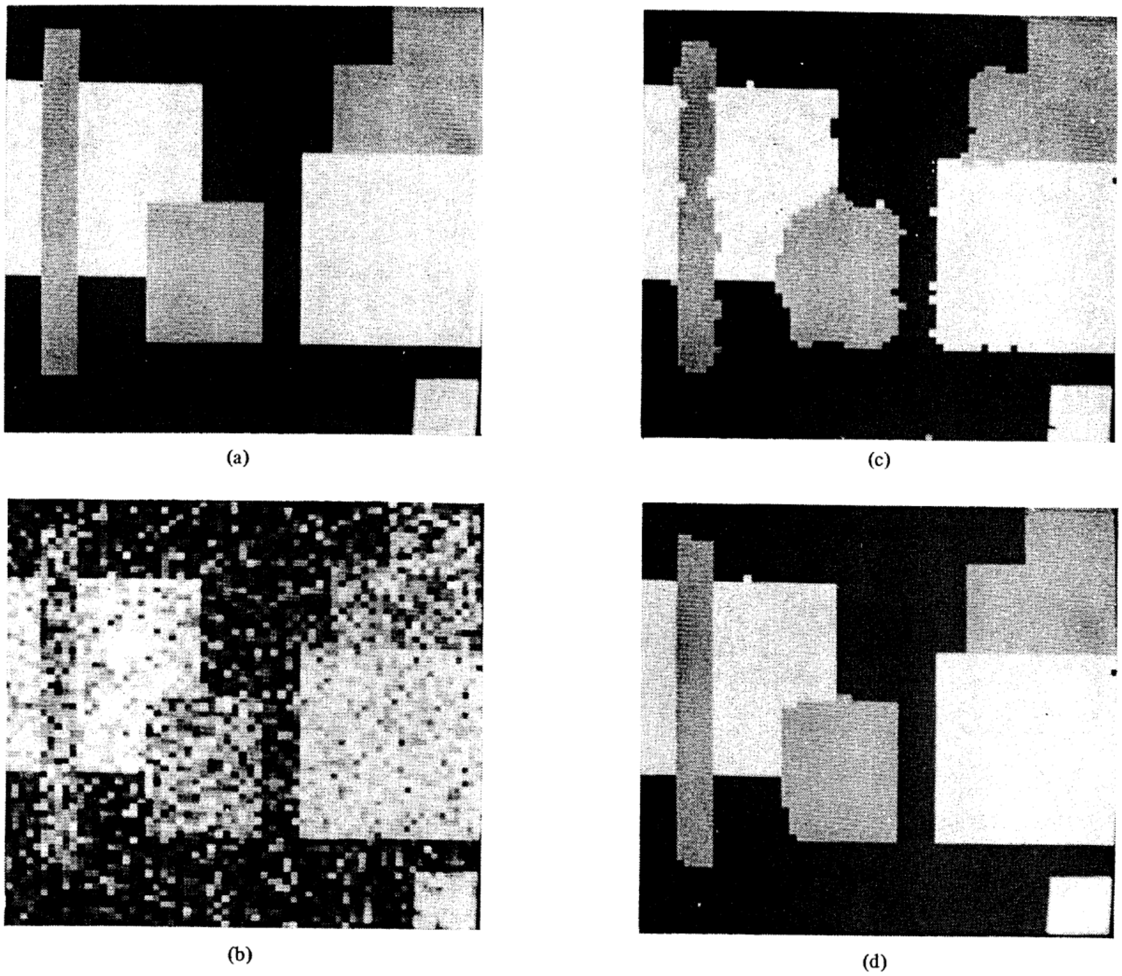


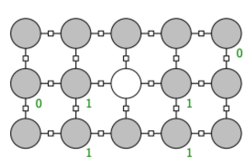
Fig. 4. (a) Original image: "Hand-drawn." (b) Degraded image: Additive noise. (c) Restoration: Without line process; 1000 iterations. (d) Restoration: Including line process; 1000 iterations.

-

- Example

Application: image denoising

Example: image denoising



- $X_i \in \{0,1\}$ is pixel value in location i
- Subset of pixels are observed

$o_i(x_i) = [x_i = \text{observed value at } i]$

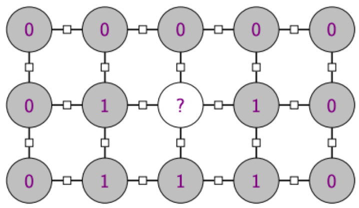
- Neighboring pixels more likely to be same than different

$t_{ij}(x_i, x_j) = [x_i = x_j] + 1$

-

- 각 위치에서 픽셀 값은 0 또는 1 값을 가짐
- 픽셀 격자에서, 일부 픽셀 결과는 알 수 있고, 나머지 픽셀을 채우고자 할 때 gibbs sampling을 사용할 수 있음
- 알 수 없는 픽셀은 각 픽셀 i에 대한 변수 X_i로 표현

Gibbs sampling for image denoising



$$t_{ij}(x_i, x_j) = [x_i = x_j] + 1$$

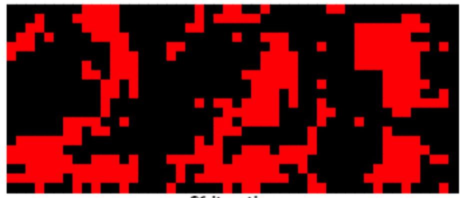
Scan through image and update each pixel given rest:

v	weight	$\mathbb{P}(X_i = v \mid X_{-i} = x_{-i})$
0	$2 \cdot 1 \cdot 1 \cdot 1$	0.2
1	$1 \cdot 2 \cdot 2 \cdot 2$	0.8

-
  - 가운데 "?" 픽셀이 모르는 픽셀이라고 가정 한다면,
    - 각 픽셀에서 값이 일치하면 2를 반환
    - 일치하지 않으면 1을 반환
  - 그 다음 가중치를 정규화하여 분포를 형성한 다음 v 를 샘플링
  - 예제에서는 "?" 픽셀은 1일 확률이 높음
- Results

Image denoising demo

Click image to start Gibbs sampling.



21 iterations

```
// Press ctrl-enter to save settings
setImage(0)            // What true image to use (either 0 or 1)
missingFrac = 0.6      // Fraction of missing pixels
coherenceFactor = 2    // Transition factors: prefer equal neighbors
initRandom = false     // Initialize sampler randomly
icm = false            // Use ICM, not Gibbs sampling
showMarginals = false  // Show marginals (averages), not samples
sampleTime = 100       // Milliseconds to wait between samples
```

Image denoising demo

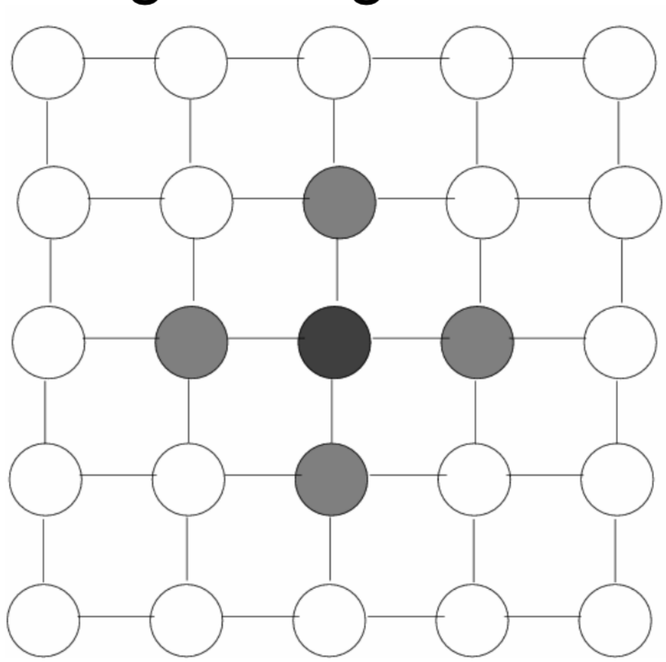
Click image to start Gibbs sampling.

1808 iterations

```
// Press ctrl-enter to save settings
setImage(0)            // What true image to use (either 0 or 1)
missingFrac = 0.6      // Fraction of missing pixels
coherenceFactor = 2    // Transition factors: prefer equal neighbors
initRandom = false     // Initialize sampler randomly
icm = false            // Use ICM, not Gibbs sampling
showMarginals = false  // Show marginals (averages), not samples
sampleTime = 100       // Milliseconds to wait between samples
```

- Example 2

- 2D Grid on {-1,+1} variables
- Neighboring variables are correlated



  -
  - 노이즈가 없는 픽셀을 $x_i \in \{-1, 1\}$ 로 설정
  - 노이즈가 관찰된 픽셀은 $y_i \in \mathbb{R}$라고 가정
  - MAP(Maximum A Posterior) 추정을 이용하여 각 픽셀에 대해 가장 가능성이 높은 single most probable guess를 찾는 것
    $x_i^* = \underset{x_i}{argmax}\ p(x_1|y_{1:D}, \theta)$

- 이를 위해 가장 간단한 모델은 픽셀이 독립적이고 균일한 prior를 갖는다고 가정하는 것
  - 즉, +1과 -1이 동일할 수 있음
- likelihood 에 대해 Gaussian Noise model을 가정하면 아래와 같음
  - $p(y_i|x_i) = N(y_i|x_i, \sigma)$
- σ가 매우 작다면, $x_i = -1$ 이면 $y_i \approx -1$이고, $x_i = +1$이면 $y_i \approx +1$이므로 각 값을 개별적으로 임계값으로 지정하여 y에서 x를 쉽게 추정할 수 있음
- 그러나 noise level이 높다면 some prior를 사용해야함
- 예를 들어 simple prior는 근처에 있는 픽셀은 같은 상태라고 가정하는 것
  - 즉, x_i가 "on" 상태이면 이웃 픽셀도 "on"일 가능성이 높음, "off" 일 때도 마찬가지.
  - 이것을 smoothness proir라고 하며, 확률 모델은 아래와 같음

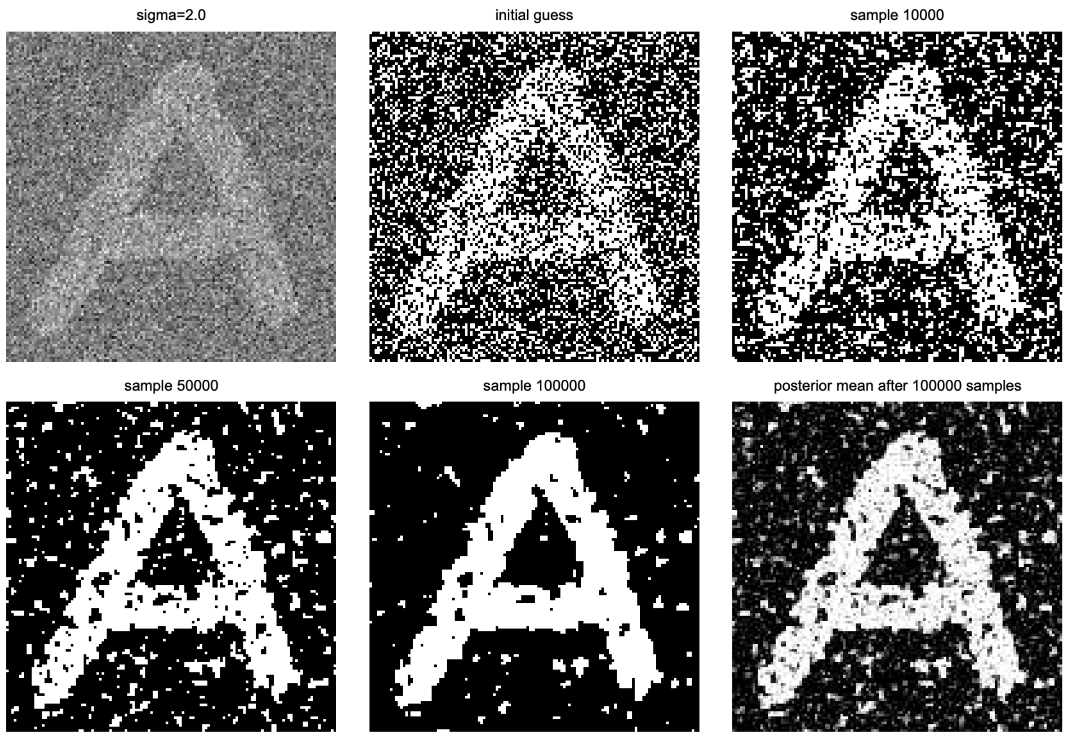
$$
\begin{aligned}
p(x,y) &= p(x)p(y|x) \\
&= \left[\frac{1}{Z}\prod_{<ij>}\psi_{ij}(x_i,x_j)\right]\left[\prod_i p(y_i|x_i)\right]
\end{aligned}
$$

-
    - $\varphi_i(x_i) = p(y_i|x_i)$: node i 를 위한 local evidence potential
    - φ_{i_j} : edge $i - j$ 를 위한 edge potential
    - $\prod_{<i,j>}$: 그래프의 모든 edge에 대한 곱(product)
- edge potential 은 다음과 같음

$$
\psi_{ij}(x_i,x_j) = \exp[J_{ij}x_ix_j] = \begin{pmatrix} e^{J_{ij}} & e^{-J_{ij}} \\ e^{-J_{ij}} & e^{J_{ij}} \end{pmatrix}
$$

    -
        - J는 smoothness prior의 strength(강도)이다.
        - $i - j$ 가 이웃 관계가 아닐 때는 $J_{ij} = 0$ 이고, 이웃 관계일 때는 $J_{ij} = J > 0$ 이다.
        - 이는 $x_i = x_j$ 일 때 더 높은 확률 나타낸다.
            - 이 때 $x_i = x_j$ 이면, $x_ix_j = 1$ 이고, $x_i \neq x_j$ 이면 $x_ix_j = -1$ 이다.

- 이러한 모델을 사용하여 inference 하게 되면 $p(x|y,\theta)$를 추정 할 수 있다.
    - 여기서 $\theta = (J, \sigma)$ 는 prior 및 likelihood의 매개변수이다.
- 아래 그림은 Gibbs Sampling을 사용하여 $p(x|y,\theta)$에서 샘플을 추출한 결과



- *Figure 9:* Example of image denoising using Gibbs sampling. We use an Ising prior with $J = 1$ and a Gaussian noise model with $\sigma = 2$.

Further ...

- Gibbs Sampling을 이용하여 노이즈를 제거한 다음, EM 알고리즘으로 Segmentation을 수행하는 예제
    - https://github.com/chenhuaizhen/Image_denoising_segmentation

Gibbs sampling

Algorithm

Algorithm: Gibbs sampler for the Ising model, n_{iter} iterations

$Get\ x\ with\ \boldsymbol{n}\ values(-1\ or\ 1).$

$for\ t = 1\ to\ n_{iter}\ do$

$\quad for\ j = 1\ to\ n\ do$

$\quad\quad i\ =\ NextSite(j)$

$$\quad\quad p\ =\ \frac{\Psi_i(x_i)\prod_{s\in nbr(i)}\Psi_{si}(x_i,x_s)}{\sum_{x_i}\Psi_i(x_i)\prod_{s\in nbr(i)}\Psi_{si}(x_i,x_s)}$$

$\quad\quad U \sim Uniform(0,1)$

$\quad\quad if\ U < p\ then$

$\quad\quad\quad x_i\ =\ 1$

$\quad\quad else$

$\quad\quad\quad x_i\ =\ -1$

$\quad\quad end\ if$

$\quad end\ for$

$end\ for$

-

Expectation-Maximization

Algorithm

Algorithm: EM for Image Segmentation

$Get\ x\ with\ \boldsymbol{n}\ values$

$Use\ KMean\ to\ find\ a\ suitable\ initialization$

$Get\ mean\ \boldsymbol{\mu_k}, covariance\ \boldsymbol{\Sigma_k}, mixing\ coefficient\ \boldsymbol{\pi_k}, log\ likelihood\ \boldsymbol{l_k}$

$while(True)$

$\#\ Expection\ step$

$$\gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(x_n|\mu_k,\Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_n|\mu_j,\Sigma_j)}$$

$\#\ Maximization\ step$

$$N_k = \sum_{n=1}^{N} \gamma(z_{nk})$$

$$\mu_k^{new} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) x_n$$

$$\Sigma_k^{new} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk})(x_n - \mu_k^{new})(x_n - \mu_k^{new})^T$$

$$\pi_k^{new} = \frac{N_k}{N}$$

$\#\ Evaluate\ step$

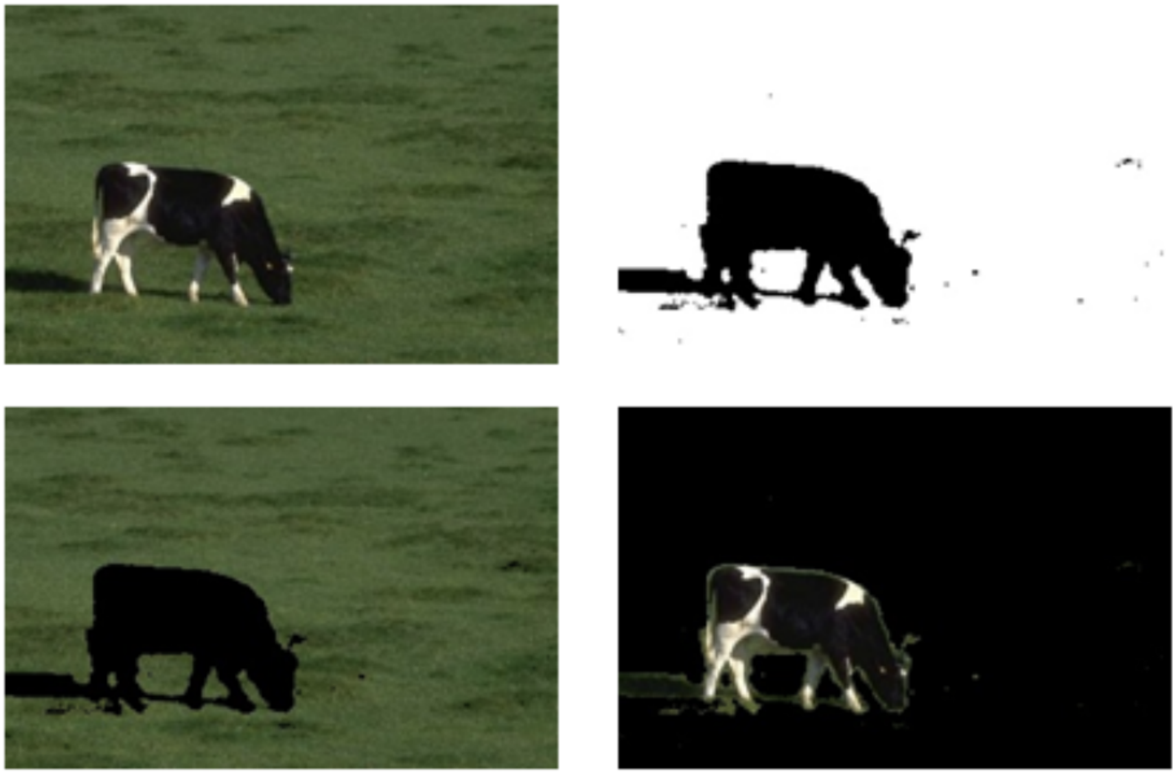
$$l_k^{new} = \sum_{n=1}^{N} \ln \{\sum_{k=1}^{K} \pi_k \mathcal{N}(x_n|\mu_k,\Sigma_k)\}$$

$if\ l\ is\ convergence\ then$

$break$

$end\ while$

-

- 

- https://stanford-cs221.github.io/autumn2021/#coursework
    - 

    - https://stanford-cs221.github.io/autumn2021/modules/

References

- https://ratsgo.github.io/statistics/2017/05/31/gibbs/
- http://sfb649.wiwi.hu-berlin.de/fedc_homepage/xplore/ebooks/html/csa/node28.html
- https://www.researchgate.net/publication/226826838_Undirected_Graphical_Models
- https://www.dam.brown.edu/people/geman/Homepage/Image%20processing,%20image%20analysis,%20Markov%20random%20fields,%20and%20MCMC/stochastic%20relaxation.pdf
- http://stanford.edu/class/ee367/Winter2018/yue_ee367_win18_report.pdf
- https://towardsdatascience.com/image-denoising-with-gibbs-sampling-mcmc-concepts-and-code-implementation-11d42a90e153
- https://stanford-cs221.github.io/autumn2021/modules/module.html#include=markov-networks/gibbs-sampling.js&mode=print1pp
- https://stanford-cs221.github.io/autumn2021-extra/modules/markov-networks/gibbs-sampling.pdf