GDSC 1기 세미나

Posterior (사후확률)

MAP/MLE 개념 소개

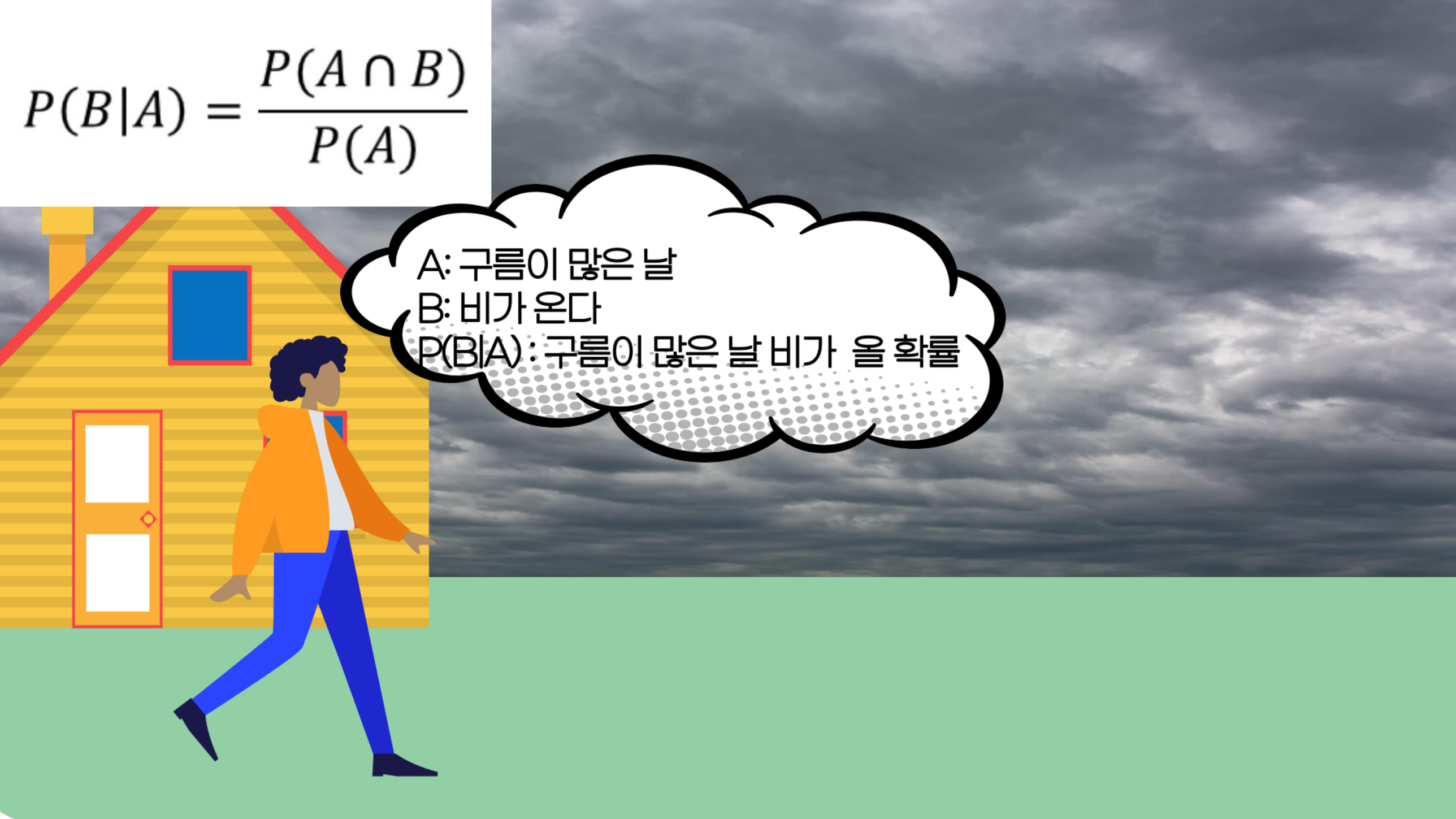
AI/ML
이선호

조건부확률

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

A인 상황일 때 B가 발생할 확률

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



Bayes' theorem 영국의 통계학자 베이즈 정리

Thomas Bayes
(1701 - 1761)

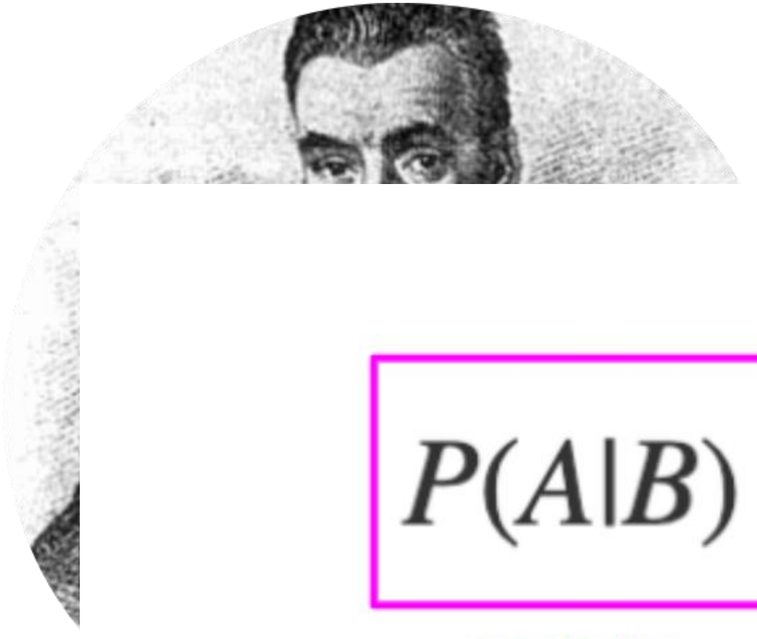
Bayes' Rule

The posterior can be decomposed according to Bayes's Rule

$$\underbrace{P(A|B)}_{\text{posterior}} = \underbrace{P(A)}_{\text{prior}} \times \frac{\overbrace{P(B|A)}^{\text{likelihood}}}{\underbrace{P(B)}_{\text{marginal}}}$$

Bayes' theorem

영국의 통계학자
베이즈 정리



Thomas Bayes
(1701 - 1761)

Bayes' Rule

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

posterior — prior — likelihood — marginal


오늘 점심 뭐 먹을까?


?
오늘 점심 뭐 먹을까?

Posterior

A : 음식
B : 오늘의 컨디션(features)

=> P(food | condition)

Data

먹은 음식	기온	기분	식사 시간	노동량	요일
떡볶이	15도	안좋음	14:00	5	수
국밥	-10도	평범	13:00	8	월
돈까스	17도	좋음	12:00	4	화
파스타	22도	좋음	12:00	3	월
칼국수	-15도	평범	13:00	7	금
. . .	. . .	. . .	. . .	. . .	. . .

Data

P(today's condition | food)
=> P({14, 3, 12, 5 ... } | 떡볶이)
=> P({14, 3, 12, 5 ... } | 국밥)

.
.
.

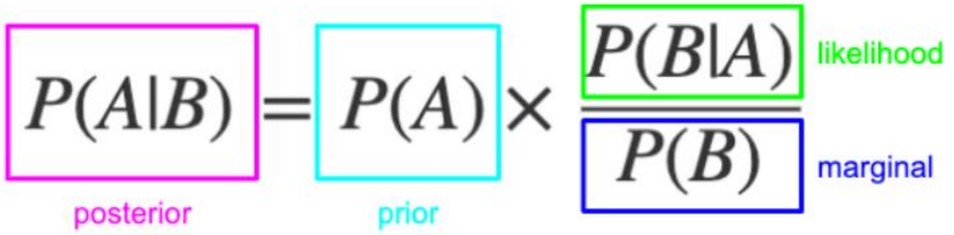
먹은 음식	기온	기분	식사 시간	노동량
떡볶이	15도	안좋음	14:00	5
국밥	-10도	평범	13:00	8
돈까스	17도	좋음	12:00	4
파스타	22도	좋음	12:00	3
칼국수	-15도	평범	13:00	7
. . .	. . .	. . .	. . .	. . .


오늘 점심 뭐 먹을까?

떡볶이 먹으러 갈까?
떡볶이 먹은 날,
오늘 같았잖아~

그래?? 음...

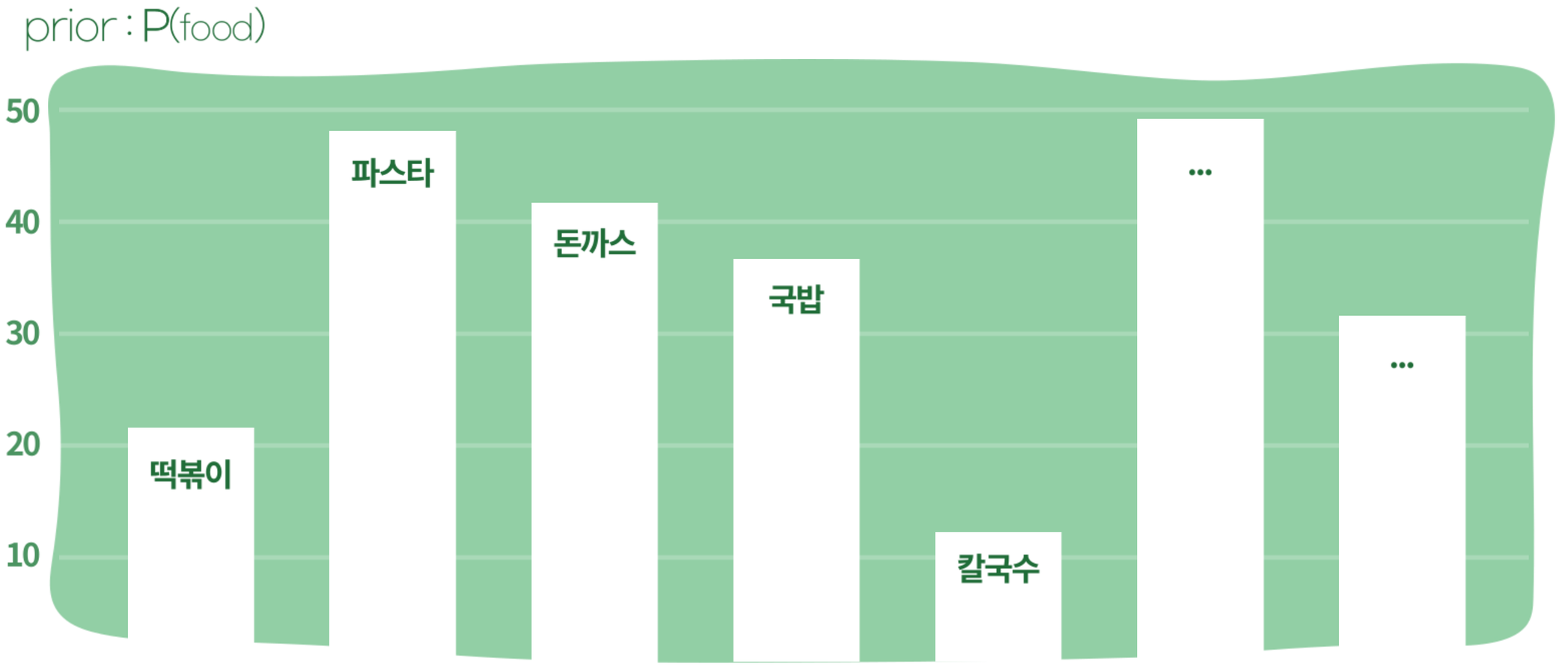
MLE 방식



MLE(Maximum Likelihood Estimation)


?
딴거 또 추천해줘!

지금까지 데이트하며 먹은 음식



Data

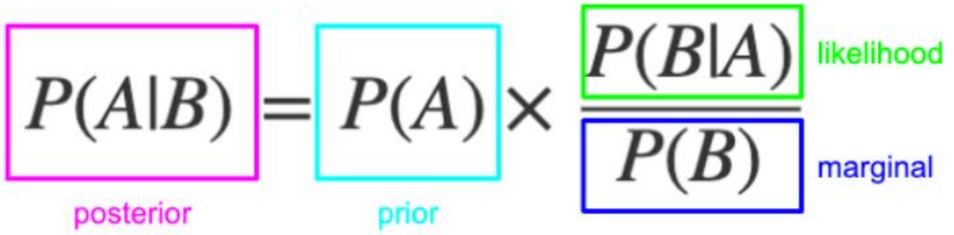
P(today's condition | food)
=> P({14, 3, 12, 5 ... } | 떡볶이)
=> P({14, 3, 12, 5 ... } | 국밥)

.
.
.

먹은 음식	기온	기분	식사 시간	노동량
떡볶이	15도	안좋음	14:00	5
국밥	-10도	평범	13:00	8
돈까스	17도	좋음	12:00	4
파스타	22도	좋음	12:00	3
칼국수	-15도	평범	13:00	7
. . .	. . .	. . .	. . .	. . .


파스타 먹는건 어때?
오늘은 파스타가
먹고싶지않아~?

MAP 방식



MAP(Maximum A posterior Probability)

전공분야 적용

Fast Image Deconvolution using Hyper-Laplacian Priors

Dilip Krishnan,
Dept. of Computer Science,
Courant Institute,
New York University
dilip@cs.nyu.edu

Rob Fergus,
Dept. of Computer Science,
Courant Institute,
New York University
fergus@cs.nyu.edu

Abstract

The heavy-tailed distribution of gradients in natural scenes have proven effective priors for a range of problems such as denoising, deblurring and super-resolution. These distributions are well modeled by a hyper-Laplacian ($p(x) \propto e^{-k|x|^\alpha}$), typically with $0.5 \leq \alpha \leq 0.8$. However, the use of sparse distributions makes the problem non-convex and impractically slow to solve for multi-megapixel images. In this paper we describe a deconvolution approach that is several orders of magnitude faster than existing techniques that use hyper-Laplacian priors. We adopt an alternating minimization scheme where one of the two phases is a non-convex problem that is separable over pixels. This per-pixel sub-problem may be solved with a lookup table (LUT). Alternatively, for two specific values of α, $1/2$ and $2/3$ an analytic solution can be found, by finding the roots of a cubic and quartic polynomial, respectively. Our approach (using either LUTs or analytic formulae) is able to deconvolve a 1 megapixel image in less than $\sim$3 seconds, achieving comparable quality to existing methods such as iteratively reweighted least squares (IRLS) that take $\sim$20 minutes. Furthermore, our method is quite general and can easily be extended to related image processing problems, beyond the deconvolution application demonstrated.

1 Introduction

Natural image statistics are a powerful tool in image processing, computer vision and computational photography. Denoising [14], deblurring [3], transparency separation [11] and super-resolution [20], are all tasks that are inherently ill-posed. Priors based on natural image statistics can regularize these problems to yield high-quality results. However, digital cameras now have sensors that record images with tens of megapixels (MP), e.g. the latest Canon DSLRs have over 20MP. Solving the above tasks for such images in a reasonable time frame (i.e. a few minutes or less), poses a severe challenge to existing algorithms. In this paper we focus on one particular problem: non-blind deconvolution, and propose an algorithm that is practical for very large images while still yielding high quality results.

Numerous deconvolution approaches exist, varying greatly in their speed and sophistication. Simple filtering operations are very fast but typically yield poor results. Most of the best-performing approaches solve globally for the corrected image, encouraging the marginal statistics of a set of filter outputs to match those of uncorrupted images, which act as a prior to regularize the problem. For these methods, a trade-off exists between accurately modeling the image statistics and being able to solve the ensuing optimization problem efficiently. If the marginal distributions are assumed to be Gaussian, a closed-form solution exists in the frequency domain and FFTs can be used to recover the image very quickly. However, real-world images typically have marginals that are non-Gaussian, as shown in Fig. 1, and thus the output is often of mediocre quality. A common approach is to assume the marginals have a Laplacian distribution. This allows a number of fast ℓ_1 and related TV-norm methods [17, 22] to be deployed, which give good results in a reasonable time. However, studies

전공분야 적용

2 Algorithm

We now introduce the non-blind deconvolution problem. $\mathbf{x}$ is the original uncorrupted linear grayscale image of N pixels; $\mathbf{y}$ is an image degraded by blur and/or noise, which we assume to be produced by convolving $\mathbf{x}$ with a blur kernel $\mathbf{k}$ and adding zero mean Gaussian noise. We assume that $\mathbf{y}$ and $\mathbf{k}$ are given and seek to reconstruct $\mathbf{x}$. Given the ill-posed nature of the task, we regularize using a penalty function $|.|^\alpha$ that acts on the output of a set of filters $f_1, \ldots, f_j$ applied to $\mathbf{x}$. A weighting term λ controls the strength of the regularization. From a probabilistic perspective, we seek the MAP estimate of $\mathbf{x}$: $p(\mathbf{x}|\mathbf{y},\mathbf{k}) \propto p(\mathbf{y}|\mathbf{x},\mathbf{k})p(\mathbf{x})$, the first term being a Gaussian likelihood and second being the [illegible] to minimizing the cost $-\log p(\mathbf{x}|\mathbf{y},\mathbf{k})$:

$$\min_{\mathbf{x}} \sum_{i=1}^{N} \left(\frac{\lambda}{2} (\mathbf{x} \oplus \mathbf{k} - \mathbf{y})_i^2 + \sum_{j=1}^{J} |(\mathbf{x} \oplus f_j)_i|^\alpha \right) \tag{1}$$

where i is the pixel index, and $\oplus$ is the 2-dimensional convolution operator. For simplicity, we use two first-order derivative filters $f_1 = [1\ -1]$ and $f_2 = [1\ -1]^T$, although additional ones can easily be added (e.g. learned filters [13, 16], or higher order derivatives). For brevity, we denote $F_i^j \mathbf{x} \equiv (\mathbf{x} \oplus f_j)_i$ for $j = 1, .., J$.

Using the half-quadratic penalty method [5, 6, 22], we now introduce auxiliary variables w_i^1 and w_i^2 (together denoted as $\mathbf{w}$) at each pixel that allow us to move the $F_i^j \mathbf{x}$ terms outside the $|.|^\alpha$ expression, giving a new cost function:

$$\min_{\mathbf{x},\mathbf{w}} \sum_i \left(\frac{\lambda}{2} (\mathbf{x} \oplus \mathbf{k} - \mathbf{y})_i^2 + \frac{\beta}{2} \left(\|F_i^1 \mathbf{x} - w_i^1\|_2^2 + \|F_i^2 \mathbf{x} - w_i^2\|_2^2 \right) + |w_i^1|^\alpha + |w_i^2|^\alpha \right) \tag{2}$$

where β is a weight that we will vary during the optimization, as described in Section 2.3. As $\beta \to \infty$, the solution of Eqn. 2 converges to that of Eqn. 1. Minimizing Eqn. 2 for a fixed β can be performed by alternating between two steps, one where we solve for $\mathbf{x}$, given values of $\mathbf{w}$ and vice-versa. The novel part of our algorithm lies in the $\mathbf{w}$ sub-problem, but first we briefly describe the $\mathbf{x}$ sub-problem and its straightforward solution.

감사합니다

질문이 있다면 말씀해주세요.