Date of acceptance Grade

Instructor

IMSE, A content based image retrieval system

Sayantan Hore

Helsinki July 1, 2014

UNIVERSITY OF HELSINKI
Department of Computer Science

HELSINGIN YLIOPISTO — HELSINGFORS UNIVERSITET — UNIVERSITY OF HELSINKI

Tiedekunta — Fakultet — Faculty	Laitos — Institution — Department
Faculty of Science	Department of Computer Science

Tekijä — Författare — Author
Sayantan Hore

Työn nimi — Arbetets titel — Title

IMSE, A content based image retrieval system

Oppiaine — Läroämne — Subject
Computer Science

Työn laji — Arbetets art — Level	Aika — Datum — Month and year	Sivumäärä — Sidoantal — Number of pages
	July 1, 2014	10 pages + 1 appendices

Tiivistelmä — Referat — Abstract

Content based image retrieval systems are out there for a few years now. We present a system, IMSE, here which searches out images using Gaussian Process Upper Confidence Bound algorithm. This algorithm has performed better in comparison to random search and plain exploration. We present a comparison of how the algorithm performs on CPU as well as GPU. We also evaluate the user experience and with the search interface.

ACM Computing Classification System (CCS):
A.1 [Introductory and Survey],
I.7.m [Document and text processing]

Avainsanat — Nyckelord — Keywords
layout, summary, list of references

Säilytyspaikka — Förvaringsställe — Where deposited

Muita tietoja — övriga uppgifter — Additional information

Contents

1 Introduction

Content based image retrieval systems have been proven to be better fits in retrieval performance. Text or tag based matching systems rely only on texts. Therefore the images themselves are secondary in the search procedure. The user sees the image and selects one by the objects and color the image represents. The combination of those objects and colors should actually be searched for instead of tags. Tags hardly represent an entire image. An image portraying a cloudy afternoon can be tagged as "Sad", based on the mood and understanding of the person creating the tags. Another images of the same genre, tagged by different persons, might miss the tag. If a user is in search for "Sad" images and finds one with the tag, selects it. The other "Sad" image would not appear in the search results as it lacks the tag. Had features and content of the image been searched for, this situation could be avoided.

IMSE is just another content based image retrieval system, using Gaussian Process Upper Confidence Bound (GP-UCB) as a retrieval algorithm. We are using MIR-FLICKR 25000 image set for testing. As this produces a huge kernel, GP-UCB is slow on CPU. We have written a GPU version of the algorithm and made a comparative study of running time over CPU and GPU. The system is written in Python (and Django for the web interface). Any system has to have a pleasing and assisting user interface. As our system is targeted towards web users, we cared for having a decent web interface, written in Twitter Bootstrap (v3.0) framework, jQuery (v1.11.1), CSS and javascript. We have followed Human Computer Interaction guidelines for user interaction.

In subsequent section, we will present the theoretical background of Reinforcement Learning, Gaussian Process, Upper Confidence Bound, Image Retrieval, GPU implementation details, user experience and performance measures.

2 The need for content based retrieval in detail

...

2.1 Shortcomings of traditional image search

...

2.2 Why content based retrieval is better

...

2.3 Approaches

...

3 Theoritical background

...

3.1 Reinforcement Learning

...

3.2 Regression Problems

Regression problems are supervised learning problems where we have multiple random variables. At least one of those variables are dependant on a subset of the rest. Lets' assume that we have to predict rainfall prediction in a city for the coming monsoon based on average summer temperature. We first take a dataset consisting of year by year rainfall and average summer temperature recorded over the past few years. We denote the temperature by x and rainfall by y. Clearly x is an independent random variable and y is dependent on x. We take a set $S = \{x_i, y_i\}$ for the past years. The goal here is to learn the relation between x and y. We will apply the relation on unknown x values to get the corresponding y values. Here relation basically means a mathematical function.

In its simplest form the function could be a linear one like,

$$f(x_i) = \theta_1 x_i + \theta_0 \tag{1}$$

But often in real life scenarios we have more than one independent variables. Besides temperature we can have amount of CO_2 emission, amount of deforestation (in

square kilometres) and so forth. Therefore in these cases instead of having a single x_i we have a vector $\mathbf{x}_i = \{x_{i1}, x_{i2}, x_{i3}, ..., x_{im}\}$. We rewrite equation 1 as,

$$f(\mathbf{x}_i) = \sum_{j=1}^{m} \theta_j x_{ij} + \theta_0 \tag{2}$$

It is not possible to match each and every y_i because of the randomness of the data. We try to go as close as possible so that $f(x)$ can represent the pattern of the output. The closeness is measured by *least square* method. The goal is to minimize the distance between y_i and $f(x_i)$ over the entire dataset. Say d_i^2 represents the squared distance between y_i and $f(x_i)$.

$$d_i^2 = [y_i - f(\mathbf{x}_i)]^2 \tag{3}$$

Let d^2 denote the summation of all d_i^2,

$$\begin{aligned} d^2 &= \sum_i [y_i - f(\mathbf{x}_i)]^2 \\ d^2 &= \sum_i [y_i - \sum_{j=1}^{m} \theta_j x_{ij} - \theta_0]^2 \\ d^2 &= \sum_i [y_i - \theta_1 x_{i1} - \theta_2 x_{i2} - \theta_3 x_{i3} - ... - \theta_m x_{im} - \theta_0]^2 \end{aligned} \tag{4}$$

To obtain minimum d^2 we take partial derivative with respect to each θ_i and set those to zero. Therefore we get a set of partial differential equations as follows,

$$\begin{aligned} \frac{\partial}{\partial \theta_0}(d^2) &= -2 \sum_i [y_i - \theta_1 x_{i1} - \theta_2 x_{i2} - \theta_3 x_{i3} - ... - \theta_m x_{im} - \theta_0] = 0 \\ \frac{\partial}{\partial \theta_1}(d^2) &= -2 \sum_i [y_i - \theta_1 x_{i1} - \theta_2 x_{i2} - \theta_3 x_{i3} - ... - \theta_m x_{im} - \theta_0] x_{i1} = 0 \\ \frac{\partial}{\partial \theta_2}(d^2) &= -2 \sum_i [y_i - \theta_1 x_{i1} - \theta_2 x_{i2} - \theta_3 x_{i3} - ... - \theta_m x_{im} - \theta_0] x_{i2} = 0 \\ &\vdots \\ \frac{\partial}{\partial \theta_m}(d^2) &= -2 \sum_i [y_i - \theta_1 x_{i1} - \theta_2 x_{i2} - \theta_3 x_{i3} - ... - \theta_m x_{im} - \theta_0] x_{im} = 0 \end{aligned} \tag{5}$$

Assuming we have n observed data points, We can write this set of equations as,

$$\begin{aligned}
\theta_0 n + \theta_1 \sum_{i=1}^{n} x_{i1} + \theta_2 \sum_{i=1}^{n} x_{i2} + ... + \theta_m \sum_{i=1}^{n} x_{im} &= \sum_{i=1}^{n} y_i \\
\theta_0 \sum_{i=1}^{n} x_{i1} + \theta_1 \sum_{i=1}^{n} x_{i1}^2 + \theta_2 \sum_{i=1}^{n} x_{i1}x_{i2} + ... + \theta_m \sum_{i=1}^{n} x_{i1}x_{im} &= \sum_{i=1}^{n} y_i x_1 \\
\theta_0 \sum_{i=1}^{n} x_{i2} + \theta_1 \sum_{i=1}^{n} x_{i1}x_{i2} + \theta_1 \sum_{i=1}^{n} x_{i2}^2 + ... + \theta_m \sum_{i=1}^{n} x_{i2}x_{im} &= \sum_{i=1}^{n} y_i x_2 \\
&\vdots \\
\theta_0 \sum_{i=1}^{n} x_{im} + \theta_1 \sum_{i=1}^{n} x_{i1}x_{im} + \theta_1 \sum_{i=1}^{n} x_{i2}x_{im} + ... + \theta_m \sum_{i=1}^{n} x_{im}^2 &= \sum_{i=1}^{n} y_i x_m
\end{aligned} \tag{6}$$

This can be written in matrix form,

$$\begin{pmatrix} \sum_{i=1}^{n} y_i \\ \sum_{i=1}^{n} y_i x_i \\ \sum_{i=1}^{n} y_i x_2 \\ \vdots \\ \sum_{i=1}^{n} y_i x_m \end{pmatrix} = \begin{pmatrix} n & \sum_{i=1}^{n} x_{i1} & \sum_{i=1}^{n} x_{i2} \dots \sum_{i=1}^{n} x_{im} \\ \sum_{i=1}^{n} x_{i1} & \sum_{i=1}^{n} x_{i1}^2 & \sum_{i=1}^{n} x_{i1}x_{i2} \dots \sum_{i=1}^{n} x_{i1}x_{im} \\ \sum_{i=1}^{n} x_{i2} & \sum_{i=1}^{n} x_{i1}x_{i2} & \sum_{i=1}^{n} x_{i2}^2 \dots \sum_{i=1}^{n} x_{i2}x_{im} \\ \vdots & & \\ \sum_{i=1}^{n} x_{im} & \sum_{i=1}^{n} x_{i1}x_{im} & \sum_{i=1}^{n} x_{i2}x_{im} \dots \sum_{i=1}^{n} x_{im}^2 \end{pmatrix} \begin{pmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \vdots \\ \theta_m \end{pmatrix}$$

In simplified notation,

$$\begin{aligned}
\mathbf{Y} &= \mathbf{X}\theta \\
\mathbf{X^T Y} &= \mathbf{X^T X}\theta \\
\theta &= (\mathbf{X^T X})^{-1}\mathbf{X^T Y}
\end{aligned} \tag{7}$$

We obtained the optimal θ.

3.3 Bayesian Probability Model

The traditional linear regression method is rigid in terms of learning the parameters. It gives fixed values for parameters for one observed dataset. Therefore for every new observed datapoint, the parameters are bound to change. It would be effective to learn a probability distribution over the parameters rather than fixed values. Having

a distribution over parameters gives us space where the parameters can move, which is convenient to understand a stochastic process. Bayesian probability model starts with a prior distribution over parameters and changes the distribution based on observations.

Bayesian probability model is built after *Bayes'* rule. It allows us to start with a prior belief on the data, which is an initial probability distribution associated to the data. Experiments generate evidences, which are used to change the prior belief, i.e. the initial distribution. The new distribution we get after incorporating the evidences is called posterior. The formulation is given below.

3.3.1 Bayes' Rule

We start with the expression of Bayes' rule. Say we have two random variables, x and y, where x is an independent variable but y depends on x. The probability of x is given by $p(x)$, the joint probability of x and y is given by $p(y, x)$. We can break $p(y, x)$ in $p(y|x)p(x)$ or $p(x|y)p(y)$ (Here $p(x|y)$ cannot be written as $p(x)$ if x is conditionally dependant on y). Therefore,

$$\begin{aligned} p(y, x) = p(y|x)p(x) = p(x|y)p(y) \\ p(y|x) = \frac{p(x|y)p(y)}{p(x)} \end{aligned} \tag{8}$$

The term $p(x)$ is marginalized over all possible values of y, so we can write $p(x)$ as,

$$p(x) = \sum_i p(x|y_i)p(y_i) \tag{9}$$

Combining Eqn. 8 and Eqn. 9,

$$p(y|x) = \frac{p(x|y)p(y)}{\sum_i p(x|y_i)p(y_i)} \tag{10}$$

In case y is a continuous variable,

$$p(y|x) = \frac{p(x|y)p(y)}{\int_y p(x|y)p(y)dy} \tag{11}$$

We start with the assumption that y_i differs from $f(\mathbf{x}_i)$ because of noise, also Each noise term ϵ_i follows Gaussian i.i.d. $\mathcal{N}(0, \sigma^2)$. Therefore the i^{th} can be written as,

$$
\begin{aligned}
y_i &= f(\mathbf{x}_i) + \epsilon_i \\
y_i &= \theta^T \mathbf{x}_i + \epsilon_i \qquad \text{where } \theta, \mathbf{x}_i \in \mathcal{R}^m
\end{aligned} \tag{12}
$$

The PDF associated with ϵ_i is,

$$
p(\epsilon_i) = \frac{1}{\sqrt{2\pi}\sigma} exp(-\frac{\epsilon^2}{2\sigma^2}) \tag{13}
$$

It follows from Eqn. 9 that $(y_i - \theta^T x_i) \sim \mathcal{N}(0, \sigma^2)$, so by combining Eqn. 9 and Eqn. 10,

$$
p(y_i - \theta^T x_i) = \frac{1}{\sqrt{2\pi}\sigma} exp(-\frac{(y_i - \theta^T x_i)^2}{2\sigma^2}) \tag{14}
$$

From Eqn. 11 we conclude,

$$
\begin{aligned}
p(y_i - \theta^T \mathbf{x}_i) &\sim \mathcal{N}(0, \sigma^2) \\
p(y_i | \mathbf{x}_i, \theta) &\sim \mathcal{N}(\theta^T \mathbf{x}_i, \sigma^2)
\end{aligned} \tag{15}
$$

We need to calculate the distribution over θ based on observed dataset $S = \{\mathbf{x}_i, y_i\}_{i=1}^n$ consisting of n data points. The conditional PDF over θ given S would be,

$$
\begin{aligned}
p(\theta|S) &= \frac{p(\theta)p(S|\theta)}{p(S)} \\
p(\theta|S) &= \frac{p(\theta)p(S|\theta)}{\int p(\theta)p(S|\theta)d\theta} \\
p(\theta|S) &= \frac{p(\theta)\prod_{i=1}^n p(y_i|\mathbf{x}_i, \theta)}{\int p(\theta)\prod_{i=1}^n p(y_i|\mathbf{x}_i, \theta)d\theta}
\end{aligned} \tag{16}
$$

We need to predict y_* for the next unobserved data point $S_* = y_*, x_*$ for some x_* given S which can be formulated as below,

$$
p(y_*|x_*, S) = \int p(y_*|x_*, \theta)p(\theta|S)d\theta \tag{17}
$$

3.4 Gaussian Process

The problem with the multivariate Gaussian distribution is that we are confined to the number of operating variables or the dimension of the input $\mathbf{x_i} \in \mathcal{R}^m$. Therefore to calculate the posterior distribution for y_*, we had to go for a two step solution. Firstly we learned a multivariate Gaussian posterior $p(\theta|S)$ over the set of parameters $\theta \in \mathcal{R}^m$ and secondly used it to calculate the posterior over y_*.

It would be nice to have a mathematical representation that does not try to learn a finite set of parameters. Therefore it does not care about dimension of input, thus truly represents an infinite dimensional multivariate Gaussian distributions. This representation is known as Gaussian process.

3.5 Upper Confidence Bound

...

3.6 Bandit Algorithms

...

3.6.1 Multi-Armed bandit

...

3.6.2 Contextual Bandit

...

3.7 Neural Networks

...

3.7.1 Multi-Layered Neural Network

...

4 Dissecting an Image

...

4.1 Extract objects from an Image

...

4.2 Extract color from an image

...

4.3 Combining objects and color features

...

5 CPU vs GPU

...

5.1 Why CPU is not enough

...

5.2 How does GPU help

...

6 Programming architecture of GPU

...

7 System design

...

7.1 On CPU

...

7.2 On GPU

...

8 Experiments and Results

...

8.1 System setup

...

8.2 Experiments

...

8.3 Performance evaluation on CPU

...

8.4 Performance evaluation On GPU

...

8.5 Performance comparison on CPU and GPU

...

9 User Interaction and Usability

Design Principles -

Gestalt Laws

Norman's principles

Shneiderman's Golden Rules

Evaluation Techniques -

GOMS

KLM

Fitts Law

Appendix 1. Model ABC

The appendices here are just models of the table of contents and the presentation. Each appendix usually starts on its own page, with the name and number of the appendix at the top. Each appendix is paginated separately.

In addition to complementing the main document, each appendix is also its own, independent entity. This means that an appendix cannot be just an image or a piece of programming, but the appendix must explain its contents and meaning.