Introduction to computer vision: Models, Learning, and Inference

Chaloemrat Boonthanawat and Chaiwoot Boonyasiriwat

ABSTRACT

The problem of computer vision can be approached mathematically in term of probabilistic model. The probabilistic context is necessary for the real world image data which are always full with noise and uncertainty. There are three main components of the solution. The first one is the probabilistic model that related the image data to the world state. The model can be classified into two types, discriminative and generative model. The second component is the learning algorithm. The learning algorithm is used to find the parameters in the model using many paired training examples. The third component is the inference algorithm. The inference algorithm is used to infer the world state from the new image data.

INTRODUCTION

The goal of computer vision is to build the machine that can extract useful information from images. The kind of task we are interested in is object classification, image recognition, motion analysis, image restoration, and etc. To put it simply, we seek to automate tasks that human visual system can do.

Mathematically, the problem can be stated as "given the image data $\boldsymbol{x}$, the program should be able to infer the world state $\boldsymbol{w}$". The world state $\boldsymbol{w}$ may be continuous (the 3D pose of a body model) or discrete (the presence or absence of a particular object). When the world state is continuous, we call regression model and when it is discrete, we call classification model. The image data $\boldsymbol{x}$ can be discrete (RGB integer value from 0 to 255) or continuous (pixel intensity by real value).

The model of interest is based on probabilistic approach. It is useful because usually the relation between image data and world state is many to one. Moreover,the image data is often full with noise from environment and we need probabilistic model to capture this information.

There are three components for the solution of the problem (Prince, 2012) :

- Model: relates the visual data $\boldsymbol{x}$ and the world state $\boldsymbol{w}$ specify by model parameters $\boldsymbol{\theta} = \{\boldsymbol{\theta_i}, \boldsymbol{\theta_j}, \ldots\}$.
- Learning algorithm: allows us to fit the parameters $\boldsymbol{\theta}$ using paired training examples $\mathcal{D} = \{\boldsymbol{x_i}, \boldsymbol{w_i}\}_{i=1}^{I}$.
- Inference algorithm: takes a new observation $\boldsymbol{x^*}$ and uses the model with a trained parameters $\hat{\boldsymbol{\theta}}$ to return the posterior probability $\Pr(\boldsymbol{w}|\boldsymbol{x^*}, \hat{\boldsymbol{\theta}})$ over the world state $\boldsymbol{w}$.

MODELS

Probability distribution and conjugacy

The choices of probability distribution depend on the task of interest. Generally, it is depending on the domain of image data $\boldsymbol{x}$ or the domain of world state $\boldsymbol{w}$ (continuous or discrete and the range of possible value) (Table 1.)

Each probability distribution will have parameters ($\boldsymbol{\theta} = \{\boldsymbol{\theta_1}, \boldsymbol{\theta_2}, \ldots\}$) corresponding to the model itself. These parameters will be specified by leaning algorithm from paired training examples. In addition, we can also have the probability distribution of parameters from the main model meaning the main model doesn't necessary have a unique parameters. It can have a wide range of possible parameters value. Its parameters will be called hyperparameters.

The choices of probability distribution are chosen such that each is a conjugacy to each other (Table 2.)

The property of conjugacy is that when two distributions multiply, it will give the same distribution with some constant. This property is extremely useful in both learning and inference algorithm by arranging the expression in a closed form.

Table 1: Domain of probability distribution

Distribution	Domain
Bernoulli	$y \in \{0, 1\}$
categorical	$y \in \{1, 2, \ldots, K\}$
univariate normal	$y \in \mathbb{R}$
multivariate normal	$\boldsymbol{y} \in \mathbb{R}^k$

Table 2: Example of possible distribution and its conjugacy

Distribution of model	Distribution of model parameters
Bernoulli	beta
categorical	Dirichlet
univariate normal	normal inverse gamma
multivariate normal	normal inverse Wishart

Hidden variable and more complex probability distribution

Even though, the probability distribution showed in Table 1 is useful for many tasks. It is still limited in real world application because real world image data is complex. To model this type of complex data, we need an additional probability distribution.

The idea of hidden variables is useful for creating more complex distribution from an simpler one. The probability of the model with hidden variables can be written as,

$$\Pr(y) = \int \Pr(y,h)dh \tag{1}$$

$$= \int \Pr(y|h)P(h)dh \tag{2}$$

for continuous case or,

$$\Pr(y) = \sum \Pr(y|h) \tag{3}$$

$$= \sum \Pr(y|h)P(h) \tag{4}$$

for discrete case.

The idea is that we choose probability distribution of $\Pr(y|h)$ and $\Pr(h)$ such that when we compute $\Pr(y)$, it will give the new probability distribution (Table 3.)

Table 3: Complex probability distribution

Distribution	$\Pr(y\|h)$	$\Pr(h)$
Mixture of Gaussian	multivariate normal	categorical
t-distribution	multivariate normal	gamma
Subspace model	multivariate normal	normal

With the knowledge of probability distribution, every model that related the image data and world state fall into either one of this two categories.

Discriminative model

In this model, we will choose the probability distribution on the world state based on the image data.

$$\Pr(\boldsymbol{w}|\boldsymbol{x},\boldsymbol{\theta_m}) \tag{5}$$

The advantage of this approach is that the inference algorithm is extremely easy by just using the new image data x^* in the discriminative model.

Generative model

In this model, we will choose the probability distribution on the image data based on the world state. We have to model both the probability distribution between image data and world state and also the probability distribution of the world state alone.

$$\Pr(\boldsymbol{x}|\boldsymbol{w},\boldsymbol{\theta_m}) \tag{6}$$

$$\Pr(\boldsymbol{w}|\boldsymbol{\theta_w}) \tag{7}$$

The advantage of this approach is that we can incorporate the knowledge about how world state generate the image data into the model. But the disadvantage is that inference algorithm is more complicated for computing.

LEARNING

There are three main ways we can used to fit the parameters $\boldsymbol{\theta}$ by using I paired training examples $\mathcal{D} = \{\boldsymbol{x_i}, \boldsymbol{w_i}\}_{i=1}^{I}$. The assumption we used is that each pair of data and world state is independent from each other.

Maximum likelihood

The maximum likelihood (ML) method finds the set of parameters $\hat{\theta}$ in which paired training examples $\mathcal{D} = \{\boldsymbol{x_i}, \boldsymbol{w_i}\}_{i=1}^{I}$ are most likely. For a discriminative model, we can write the equation as,

$$\hat{\boldsymbol{\theta}}_{\boldsymbol{m}} = \underset{\boldsymbol{\theta_m}}{\operatorname{argmax}} \left[\Pr(\boldsymbol{w_1},\ldots,\boldsymbol{w_I}|\boldsymbol{x_1},\ldots,\boldsymbol{x_I},\boldsymbol{\theta_m})\right] \tag{8}$$

$$= \underset{\boldsymbol{\theta_m}}{\operatorname{argmax}} \left[\prod_{i=1}^{I} \Pr(\boldsymbol{w_i}|\boldsymbol{x_i},\boldsymbol{\theta_m})\right] \tag{9}$$

For a generative model, we can write the equation as,

$$\hat{\boldsymbol{\theta}}_{\boldsymbol{m}} = \underset{\boldsymbol{\theta_m}}{\operatorname{argmax}} \left[\Pr(\boldsymbol{x_1},\ldots,\boldsymbol{x_I}|\boldsymbol{w_1},\ldots,\boldsymbol{w_I},\boldsymbol{\theta_m})\right] \tag{10}$$

$$= \underset{\boldsymbol{\theta_m}}{\operatorname{argmax}} \left[\prod_{i=1}^{I} \Pr(\boldsymbol{x_i}|\boldsymbol{w_i},\boldsymbol{\theta_m})\right] \tag{11}$$

and for parameters of the world state probability distribution $\Pr(\boldsymbol{w}|\boldsymbol{\theta_w})$, we use the training world state $\{\boldsymbol{w_i}\}_{i=1}^{I}$

$$\hat{\boldsymbol{\theta}}_{\boldsymbol{w}} = \underset{\boldsymbol{\theta_w}}{\operatorname{argmax}} \left[\Pr(\boldsymbol{w_1},\ldots,\boldsymbol{w_I}|\boldsymbol{\theta_w})\right] \tag{12}$$

$$= \underset{\boldsymbol{\theta_w}}{\operatorname{argmax}} \left[\prod_{i=1}^{I} \Pr(\boldsymbol{w_i}|\boldsymbol{\theta_w})\right] \tag{13}$$

Maximum a posteriori

In maximum a posteriori (MAP), we introduce prior information about the parameters $\boldsymbol{\theta}$. These may come from previous experience or knowledge. For a discriminative

model, we can write the equation as,

$$\hat{\boldsymbol{\theta}}_{\boldsymbol{m}} = \underset{\boldsymbol{\theta_m}}{\operatorname{argmax}} \left[\Pr(\boldsymbol{\theta_m}|\{\boldsymbol{x_i}, \boldsymbol{w_i}\}_{i=1}^{I})\right] \tag{14}$$

$$= \underset{\boldsymbol{\theta_m}}{\operatorname{argmax}} \left[\Pr(\{\boldsymbol{x_i}, \boldsymbol{w_i}\}_{i=1}^{I}|\boldsymbol{\theta_m}) \Pr(\boldsymbol{\theta_m})\right] \tag{15}$$

$$= \underset{\boldsymbol{\theta_m}}{\operatorname{argmax}} \left[\prod_{i=1}^{I} \Pr(\boldsymbol{w_i}, \boldsymbol{x_i}|\boldsymbol{\theta_m}) \Pr(\boldsymbol{\theta_m})\right] \tag{16}$$

$$= \underset{\boldsymbol{\theta_m}}{\operatorname{argmax}} \left[\prod_{i=1}^{I} \Pr(\boldsymbol{w_i}|\boldsymbol{x_i}, \boldsymbol{\theta_m}) \Pr(\boldsymbol{x_i}|\boldsymbol{\theta_m}) \Pr(\boldsymbol{\theta_m})\right] \tag{17}$$

$$= \underset{\boldsymbol{\theta_m}}{\operatorname{argmax}} \left[\prod_{i=1}^{I} \Pr(\boldsymbol{w_i}|\boldsymbol{x_i}, \boldsymbol{\theta_m}) \Pr(\boldsymbol{\theta_m})\right] \tag{18}$$

For a generative model, we can write the equation as,

$$\hat{\theta}_m = \underset{\boldsymbol{\theta_m}}{\operatorname{argmax}} \left[\Pr(\boldsymbol{\theta_m}|\{\boldsymbol{x_i}, \boldsymbol{w_i}\}_{i=1}^{I})\right] \tag{19}$$

$$= \underset{\boldsymbol{\theta_m}}{\operatorname{argmax}} \left[\prod_{i=1}^{I} \Pr(\boldsymbol{w_i}, \boldsymbol{x_i}|\boldsymbol{\theta_m}) \Pr(\boldsymbol{\theta_m})\right] \tag{20}$$

$$= \underset{\boldsymbol{\theta_m}}{\operatorname{argmax}} \left[\prod_{i=1}^{I} \Pr(\boldsymbol{x_i}|\boldsymbol{w_i}, \boldsymbol{\theta_m}) \Pr(\boldsymbol{w_i}|\boldsymbol{\theta_m}) \Pr(\boldsymbol{\theta_m})\right] \tag{21}$$

$$= \underset{\boldsymbol{\theta_m}}{\operatorname{argmax}} \left[\prod_{i=1}^{I} \Pr(\boldsymbol{x_i}|\boldsymbol{w_i}, \boldsymbol{\theta_m}) \Pr(\boldsymbol{\theta_m})\right] \tag{22}$$

We can see that maximum likelihood is a special case of maximum a posteriori when we don't have any information about parameters in advance.

In order to simplify the learning algorithm, we can use the logarithm of the expression because it is monotonic function. The maximum parameters of transformed function will still remain the same, but it will give an easier expression to dealt with.

Expectation maximization algorithm

The EM algorithm is used to find the parameters $\hat{\boldsymbol{\theta}}_{\boldsymbol{m}}$ with the equation of the form (maximum likelihood),

$$\hat{\boldsymbol{\theta}}_{\boldsymbol{m}} = \underset{\boldsymbol{\theta_m}}{\operatorname{argmax}} \left\{\sum_{i=1}^{I} \log \left[\sum \Pr(\boldsymbol{x_i}, \boldsymbol{w_i}|\boldsymbol{h_i}, \boldsymbol{\theta_m}) \Pr(\boldsymbol{h_i})\right]\right\} \tag{23}$$

or in this form (maximum a posteriori)

$$= \underset{\boldsymbol{\theta_m}}{\operatorname{argmax}} \left\{\sum_{i=1}^{I} \log \left[\sum \Pr(\boldsymbol{x_i}, \boldsymbol{w_i}|\boldsymbol{h_i}, \boldsymbol{\theta_m}) \Pr(\boldsymbol{h_i})\right] \Pr(\boldsymbol{\theta_m})\right\} \tag{24}$$

The same is true for the case when hidden variables are continuous. There are two main steps in the algorithm which will be iterated until the maximum parameters are found. 1.) the E-step

$$q_i(\boldsymbol{h_j}) = \frac{\Pr(\boldsymbol{x_i}, \boldsymbol{w_i}|\boldsymbol{h_j}, \boldsymbol{\theta_m^{[t]}}) \Pr(\boldsymbol{h_j}|\boldsymbol{\theta_m^{[t]}})}{\Pr(\boldsymbol{x_i}, \boldsymbol{w_i})} \tag{25}$$

2.) the M-step

$$\boldsymbol{\theta_m^{[t+1]}} = \underset{\boldsymbol{\theta_m}}{\operatorname{argmax}} \left\{\sum_{i=1}^{I} \sum q_i(\boldsymbol{h_j}) \log \left[\Pr(\boldsymbol{x_i}, \boldsymbol{w_i}, \boldsymbol{h_i}|\boldsymbol{\theta_m})\right]\right\} \tag{26}$$

The Bayesian approach

The Bayesian approach is similar to the maximum a posteriori. Instead of finding a unique parameters $\hat{\boldsymbol{\theta}}$, we find the probability of all parameters. Each parameters will contribute to the inference based on its probability, the expression is

$$\Pr(\boldsymbol{\theta}|\mathcal{D}) = \Pr(\boldsymbol{\theta}|\{\boldsymbol{x_i}, \boldsymbol{w_i}\}_{i=1}^{I}) \tag{27}$$

$$= \frac{\Pr(\{\boldsymbol{x_i}, \boldsymbol{w_i}\}_{i=1}^{I}|\boldsymbol{\theta}) \Pr(\boldsymbol{\theta})}{\Pr(\{\boldsymbol{x_i}, \boldsymbol{w_i}\}_{i=1}^{I})} \tag{28}$$

$$= \frac{\prod_{i=1}^{I} \Pr(\boldsymbol{w_i}, \boldsymbol{x_i}|\boldsymbol{\theta}) \Pr(\boldsymbol{\theta})}{\Pr(\{\boldsymbol{x_i}, \boldsymbol{w_i}\}_{i=1}^{I})} \tag{29}$$

INFERENCE

In the maximum likelihood (ML) and maximum a posteriori method (MAP), we would get a unique parameters $\hat{\boldsymbol{\theta}}$. These parameters is used to find the probability of each possible world state $\boldsymbol{w}$ given the new image data $\boldsymbol{x}^*$.

Inference algorithm for discriminative model (ML, MAP method)

For discriminative model, we have already directly constructed an expression for the posterior distribution, and we simply evaluate it with the new data.

$$\Pr(\boldsymbol{w}|\boldsymbol{x}^*, \hat{\boldsymbol{\theta}}_{\boldsymbol{m}}) \tag{30}$$

Inference algorithm for generative model (ML, MAP method)

For generative model, we use Bayes' rule to calculate the posterior distribution $\Pr(\boldsymbol{w}|\boldsymbol{x}^*, \hat{\boldsymbol{\theta}})$. For continuous case, we got

$$\Pr(\boldsymbol{w}|\boldsymbol{x}^*, \hat{\boldsymbol{\theta}}) = \frac{\Pr(\boldsymbol{x}^*|\boldsymbol{w}, \hat{\boldsymbol{\theta}}_{\boldsymbol{m}}) \Pr(\boldsymbol{w}|\hat{\boldsymbol{\theta}}_{\boldsymbol{w}})}{\int \Pr(\boldsymbol{x}^*|\boldsymbol{w}, \hat{\boldsymbol{\theta}}_{\boldsymbol{m}}) \Pr(\boldsymbol{w}|\hat{\boldsymbol{\theta}}_{\boldsymbol{w}}) d\boldsymbol{w}} \tag{31}$$

For discrete case, we got

$$\Pr(\boldsymbol{w}|\boldsymbol{x}^*, \hat{\boldsymbol{\theta}}) = \frac{\Pr(\boldsymbol{x}^*|\boldsymbol{w}, \hat{\boldsymbol{\theta}}_{\boldsymbol{m}}) \Pr(\boldsymbol{w}|\hat{\boldsymbol{\theta}}_{\boldsymbol{w}})}{\sum_{\boldsymbol{w}} \Pr(\boldsymbol{x}^*|\boldsymbol{w}, \hat{\boldsymbol{\theta}}_{\boldsymbol{m}}) \Pr(\boldsymbol{w}|\hat{\boldsymbol{\theta}}_{\boldsymbol{w}})} \tag{32}$$

Inference algorithm for Bayesian approach

Evaluating the predictive distribution is more difficult for the Bayesian case since we have not estimated a unique parameter for a model. But we instead have a probability distribution over all possible models (all possible parameters). To calculate the probability of world state given the new data $\boldsymbol{x^*}$, we need to weight the probability of all possible models. For a discriminative model, we can write the equation as (Bishop, 2006),

$$\begin{aligned}\Pr(\boldsymbol{w^*}|\boldsymbol{x^*}) &= \int \Pr(\boldsymbol{\theta_m}|\mathcal{D}) \Pr(\boldsymbol{w^*}|\boldsymbol{x^*}, \boldsymbol{\theta_m}) d\boldsymbol{\theta_m} \\ &= \int \left[\frac{\prod_{i=1}^{I} \Pr(\boldsymbol{w_i}, \boldsymbol{x_i}|\boldsymbol{\theta_m}) \Pr(\boldsymbol{\theta_m})}{\Pr(\{\boldsymbol{x_i}, \boldsymbol{w_i}\}_{i=1}^{I})} \right] \Pr(\boldsymbol{w^*}|\boldsymbol{x^*}, \boldsymbol{\theta_m}) d\boldsymbol{\theta_m} \\ &= \int \left[\frac{\prod_{i=1}^{I} \Pr(\boldsymbol{w_i}|\boldsymbol{x_i}, \boldsymbol{\theta_m}) \Pr(\boldsymbol{\theta_m})}{\prod_{i=1}^{I} \Pr(\boldsymbol{w_i}|\boldsymbol{x_i})} \right] \Pr(\boldsymbol{w^*}|\boldsymbol{x^*}, \boldsymbol{\theta_m}) d\boldsymbol{\theta_m} \end{aligned} \tag{33}$$

For a generative model, we can write the equation as,

$$\Pr(\boldsymbol{w^*}|\boldsymbol{x^*}) = \int \Pr(\boldsymbol{\theta}|\mathcal{D}) \Pr(\boldsymbol{w^*}|\boldsymbol{x^*}, \boldsymbol{\theta}) d\boldsymbol{\theta} \tag{34}$$

$$= \int \Pr(\boldsymbol{\theta_m}, \boldsymbol{\theta_w}|\{\boldsymbol{x_i}, \boldsymbol{w_i}\}_{i=1}^{I}) \left[\frac{\Pr(\boldsymbol{x^*}|\boldsymbol{w}, \boldsymbol{\theta_m}) \Pr(\boldsymbol{w}|\boldsymbol{\theta_w})}{\int \Pr(\boldsymbol{x^*}|\boldsymbol{w}, \boldsymbol{\theta_m}) \Pr(\boldsymbol{w}|\boldsymbol{\theta_w}) d\boldsymbol{w}} \right] d\boldsymbol{\theta_m} d\boldsymbol{\theta_w} \tag{35}$$

$$= \int \Pr(\boldsymbol{\theta_m}|\{\boldsymbol{x_i}, \boldsymbol{w_i}\}_{i=1}^{I}) \Pr(\boldsymbol{\theta_w}|\{\boldsymbol{x_i}, \boldsymbol{w_i}\}_{i=1}^{I}) \left[\frac{\Pr(\boldsymbol{x^*}|\boldsymbol{w}, \boldsymbol{\theta_m}) \Pr(\boldsymbol{w}|\boldsymbol{\theta_w})}{\int \Pr(\boldsymbol{x^*}|\boldsymbol{w}, \boldsymbol{\theta_m}) \Pr(\boldsymbol{w}|\boldsymbol{\theta_w}) d\boldsymbol{w}} \right] d\boldsymbol{\theta_m} d\boldsymbol{\theta_w} \tag{36}$$

$$= \int \left[\frac{\prod_{i=1}^{I} \Pr(\boldsymbol{x_i}|\boldsymbol{w_i}, \boldsymbol{\theta_m}) \Pr(\boldsymbol{\theta_m})}{\prod_{i=1}^{I} \Pr(\boldsymbol{x_i}|\boldsymbol{w_i})} \right] \left[\prod_{i=1}^{I} \Pr(\boldsymbol{w}|\boldsymbol{\theta_w}) \Pr(\boldsymbol{\theta_w}) \right] \left[\frac{\Pr(\boldsymbol{x^*}|\boldsymbol{w}, \boldsymbol{\theta_m}) \Pr(\boldsymbol{w}|\boldsymbol{\theta_w})}{\int \Pr(\boldsymbol{x^*}|\boldsymbol{w}, \boldsymbol{\theta_m}) \Pr(\boldsymbol{w}|\boldsymbol{\theta_w}) d\boldsymbol{w}} \right] d\boldsymbol{\theta_m} d\boldsymbol{\theta_w} \tag{37}$$

In practice, if we choose an appropriate probability distribution (conjugacy), we do not necessary have to calculate the integral directly.

CONCLUSION

Even though the three components are the foundation of the solution, there are still many issues about performance, robustness and the amount of training data required for learning algorithm. More technique and method is needed to ensure that the program work nicely in real life. Nevertheless, these knowledge provided a foundation that can be applied to any task at hand.

ACKNOWLEDGMENTS

We would like to thank members of the Mahidol University Center for Scientic Computing (MCSC) group for comments and suggestions.

REFERENCES

Bishop, C. M., 2006, Pattern recognition and machine learning: Springer.

Prince, S. J. D., 2012, Computer vision: Models, learning, and inference: Cambridge University Press, New York.