Analysis of Discriminative and Generative Graphical Models on Image Classification

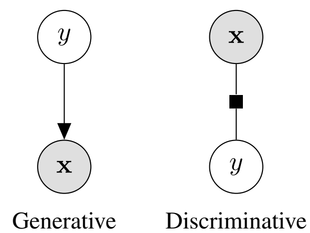
William Watson
Johns Hopkins University
billwatson@jhu.edu

Abstract

In statistical classification, there are two main approaches to learning: *generative* and *discriminative*. This proposal will outline a procedure to explore the performance of corresponding probabilistic graphical models on a simple classification task.

1 Introduction

For a classification task, we are interested in the probability distribution of a class label y conditioned on the sample features $\mathbf{x}$, written as $p(y \,|\, \mathbf{x})$. We can approach modeling this distribution in two ways: *generative* or *discriminative*.



1.1 Generative Approach

A generative model is a statistical process to describe the observed data $\mathbf{x}$. More specifically, generative classifiers seek to model the joint probability $p(\mathbf{x},\, y)$ from our data[6]. Predictions can be made with Bayes rule as follows:

$$\hat{y} \;=\; \operatorname*{argmax}_{y}\; p(\mathbf{x},\, y) \;=\; \operatorname*{argmax}_{y}\; p(y \,|\, \mathbf{x})\, p(\mathbf{x}) \;=\; \operatorname*{argmax}_{y}\; p(\mathbf{x} \,|\, y)\, p(y) \tag{1}$$

1.2 Discriminative Approach

In contrast, a discriminative model is a statistical process to describe relations between the observed data $\mathbf{x}$ and some latent variables $\mathbf{h}$. For discriminative classifiers, the latent variables are class labels y, and model the posterior distribution $p(y \,|\, \mathbf{x})$ directly [6]. Hence, prediction becomes:

$$\hat{y} \;=\; \operatorname*{argmax}_{y}\; p(y \,|\, \mathbf{x}) \tag{2}$$

2 Data

I will use the Rijkmuseum Dataset [3] to perform type categorization. However, to make parameter estimation and inference tractable, the problem will be converted to a binary (paintings/photos) task.

Images will be downsampled in size and color encoding to reduce the variable and state space. There are 3,583 paintings (*schilderij*) and 2,269 photos (*foto*).

3 Models

In order to compare generative and discriminative approaches to classification, I will explore their static and general versions, as outlined in Figure 2.

3.1 Baseline

The baseline model will consist of a *Naive Bayes* classifier, a simple generative Bayesian network that assumes all the features $x_i \in \mathbf{x}$ are independent from one another and the sole parent is the class label y. This assumption simplifies the factorization for the joint distribution.

The discriminative analog to Naive Bayes is *Logistic Regression*, which is also a naive classifier, yet models the conditional probability directly.

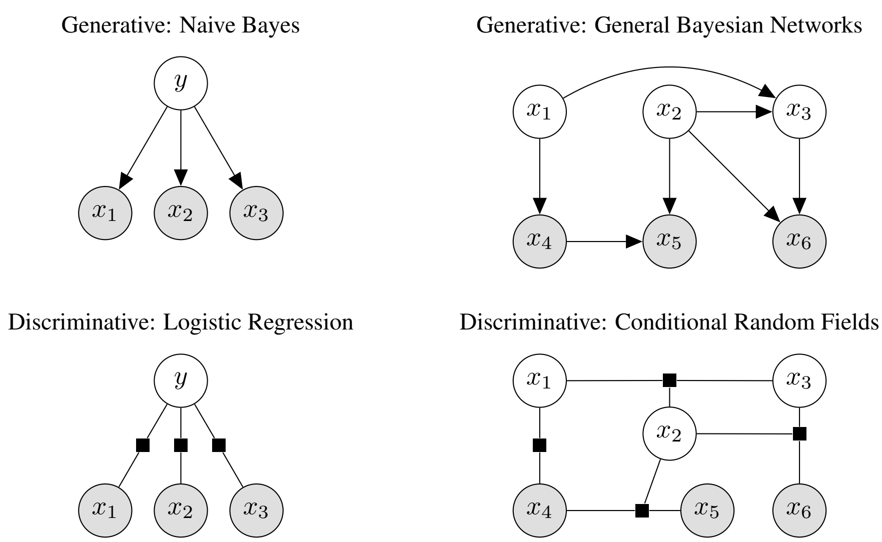
3.2 General

The general counterparts to the baseline models are the generative *General Bayesian Network* and its discriminative analog *Conditional Random Fields (CRFs)*.

General Bayesian Networks are probabilistic models that encode the conditional dependencies of its variables via a directed acyclic graph. Each variable is dependent only on its parents, allowing for more complex interactions between the features.

CRFs are an undirected probabilistic graphical model whose nodes can be partitioned into two disjoint sets: $\mathbf{x}$ and $\mathbf{y}$, and models the conditional distribution $p(\mathbf{y} \mid \mathbf{x})$ directly.

Figure 2: Generative vs. Discriminative Models



4 Parameter Estimation

Parameter estimation for generative models will use Maximum Likelihood Estimation. Logistic Regression will optimize its parameters via the L-BFGS algorithm. CRFs will use an approach depending on the library used (Section 6).

5 Inference

Inference for generative models will use belief propagation, if tractable. Otherwise, variational or sampling methods [2] will be used to approximate the conditional posterior.

CRF inference will use loopy belief propagation or sampling techniques, since exact inference is intractable in general graphs. However, since Logistic Regression is a tree, message passing algorithms yield exact solutions.

6 Software

This project will use OpenCV [1] for image processing. Our generative models will use pgmpy[1] for learning and inference. Logistic Regression can be learned with scitkit-learn [5]. CRFs will either use PyStruct [4] or R's CRF library[2], depending on which library is easier to integrate with the preferred structure of the task.

References

[1] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[2] M. D. Hoffman and A. Gelman. The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo. *arXiv e-prints*, page arXiv:1111.4246, Nov 2011.

[3] T. E. J. Mensink and J. C. van Gemert. The rijksmuseum challenge: Museum-centered visual recognition. In *ACM International Conference on Multimedia Retrieval*, 2014.

[4] A. C. Müller and S. Behnke. pystruct - learning structured prediction in python. *Journal of Machine Learning Research*, 15:2055–2060, 2014.

[5] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[6] A. Y. Ng and M. Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. *Adv. Neural Inf. Process. Sys*, 2, 04 2002.

[1] http://pgmpy.org

[2] https://cran.r-project.org/web/packages/CRF/index.html