The Effect of Image Interpolation on the Generalization of Generative and Discriminative Models

William Watson
Johns Hopkins University
billwatson@jhu.edu

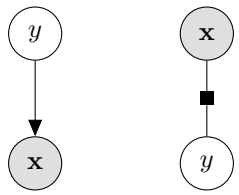
Abstract

In statistical classification, there are two main approaches to learning: *generative* and *discriminative*. In computer vision, *image interpolation* is a method to rescale images analgous to dimensionality reduction. We explore the generalization ability of *generative-discriminative pairs* when interpolating images to smaller sizes.

1 Introduction

2 Statistical Classification

Classification is the task of assigning a label y to a set of observed features $\mathbf{x}$. Yet the way we approach modeling these relationships can be broken down into either the *generative* or the *discriminative* approach.



Generative Discriminative

The generative approach models the joint distribution $p(y, \mathbf{x})$, and can assign labels through Bayes rule [6].

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

Generative models are attractive in that distributions are learned over the feature space for each indvidual class. However, an important limitation mentioned by Callum et. al [5] is that modeling a distribution *per feature* quickly becomes intractable as the dimensionality of our observed variables $\mathbf{x}$ grows. While simple models can mitigate these issues by assuming independence among the features, allowing complex dependencies between inputs offers the ability to increase performance.

An alternative approach is to model the conditonal probabilities directly, ignoring the feature distributions. This is the discriminative approach, and is sometimes referred to as a *distribution-free classifier*. By ignoring the feature distributions, parameters are learned only on the conditional likelihood $p(y|\mathbf{x})$, resulting in a compact model that can handle large feature spaces, as dependencies are ignored [5].

To compare generative and discriminative modeling, one can experiment with pairs of classifiers that can be considered analgous to each other. More formally, a *generative-discriminative pair* is

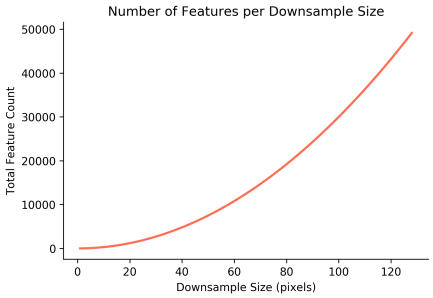


Figure 2: Exponentional growth in the number features quickly makes modeling intractable.

a parametric family of probabilistic models that can either be fit to maximize the joint probability $p(y, \mathbf{x})$ or the conditional likelihood $p(y|\mathbf{x})$ [6]. The simplest pairing is the Naive Bayes classifier (generative) and Logistic Regression (discriminative). This paradigm can be extended to sequential and general models to form more pairs. We discuss our experimental pairs in Sections 5 and 6.

3 Image Interpolation

The most common problem encountered in machine learning is the *curse of dimensionality*. Esssentially, as our feature space $\mathbf{x}$ grows in size, we require a larger number of parameters to estimate. This can quickly become intractable in complex models, and reasonably difficult in simple ones. We can see the growth in features as we increase the image size in Figure 2. One method to reduce the number of features for a data set is Principle Component Analysis, which projects data to a smaller dimension while maximining the variance. However, interapibility is lost, and the structure of the painting is unitnelligble.

Image interpolation offers a convient way to downsample images according to their local neighborhood, aggregating otherwise noisy estimates into a single value. This may be beneficial for classification tasks, as we combine a method analgous to dimsionality reduction while preserving the fundamental structure of the orignal content. We use OpenCV's implementation of interpolation [1].

Formally, image interpolation is the task of rescaling an image of one size to another. This can be used to both decimate and expand the image. Several methods exists for this task, including linear, bicubic, nearest neighbor, and *area relation*.

The perferred method for image decimation, or downsampling, is to resampling using pixel area relation as it gives moire'-free results. We can see in Figure 3 that the general space of the pixels in relation to there values is replicated, albeit with information loss.

4 Data

4.1 Descriptive Statistics

4.2 Preprocessing Pipeline

5 Baseline Models

5.1 Naive Bayes

Naive Bayes is a simple generative classifier based on applying Bayes' theorem to extract the conditional likelihood $p(y \mid \mathbf{x})$. Its' fundamental assumption for the features is "naive" in that it assumes that every pair of features is conditionally independent given then class label y for K features.

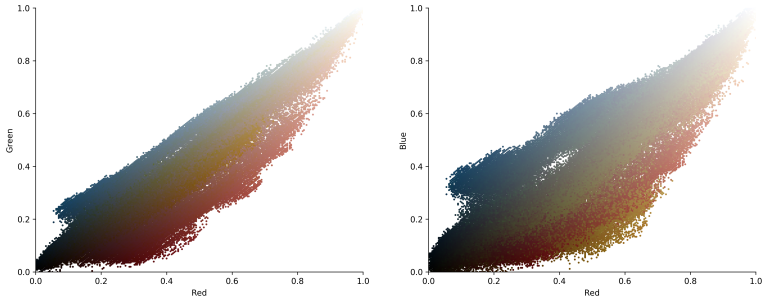
We can formulate our model from the joint distribution:

$$p(y, \mathbf{x}) \;=\; p(y) \cdot \prod_{k=1}^{K} p(x_k \mid y) \tag{2}$$

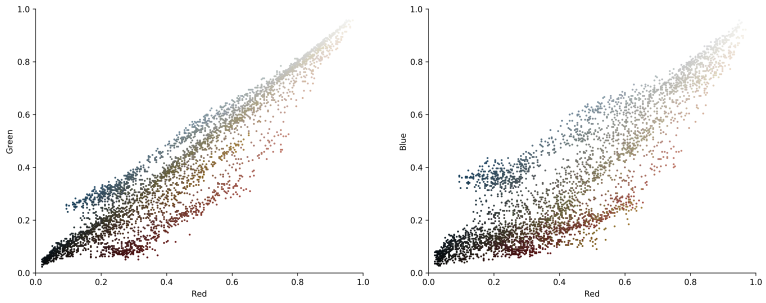
Figure 3: Results of Interpolating a 249,000 pixel image to a 4,096 pixel image.

(a) Original Image (498x500)

(b) Decimated Image (64x64)



(c) Pixel Plot for the Original Image



(d) Pixel Plot for the Decimated Image

Our task is to find the class label y that maximizes the posterior conditional likelihood:

$$\begin{aligned} \hat{y} &= \underset{y}{\text{argmax}}\ p(y|\mathbf{x}) \\ &= \underset{y}{\text{argmax}}\ \frac{p(y) \cdot \prod_{k=1}^{K} p(x_k \mid y)}{p(\mathbf{x})} \\ &= \underset{y}{\text{argmax}}\ p(y) \cdot \prod_{k=1}^{K} p(x_k \mid y) \end{aligned} \tag{3}$$

We make use of the conditional independence assumption that all features are soely dependent on y alonge, and that $p(\mathbf{x})$ is constant.

Here, $p(y)$ is the frequency of class y is the training set. For our purpuses we assume the features are normally distributed per class. Hence, our conditonal feature likelihood is:

$$p(x_i \mid y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right) \tag{4}$$

To learn a Gaussian Naive Bayes model, we use Maximum Likelihood Estimates for the parameters μ_{kc} and σ_{kc} per feature k and class c, and class weights θ_c. Hence the MLE estimates for N samples are:

$$\begin{aligned}\theta_c &= \frac{1}{N}\sum_{i=1}^{N}\mathbb{1}\left\{y_i = y_c\right\} = \frac{N_c}{N} \\ \mu_{kc} &= \frac{1}{N_c}\sum_{i=1}^{N}\mathbb{1}\left\{y_i = y_c\right\}\cdot x_{ik} \\ \sigma_{kc} &= \frac{1}{N_c}\sum_{i=1}^{N}\mathbb{1}\left\{y_i = y_c\right\}\cdot\left(x_{ik}-\mu_{kc}\right)^2\end{aligned} \tag{5}$$

where $N_c = \sum_{i=1}^{N}\mathbb{1}\left\{y_i = y_c\right\}$, or the number of data points belonging to class c [3].

5.2 Logistic Regression

Logistic Regression is the discriminative analog to the Naive Bayes classifer, sometimes referred to as the *maxmimum-entropy classifer* or a normalized *log-linear model*. As a discriminative model, we can optimize directly for the conditional likelihood $p(y \mid \mathbf{x})$, with $Z(\mathbf{x})$ as the normalizing parition function.

$$p(y \mid \mathbf{x}) = \frac{1}{Z(\mathbf{x})}\exp\left\{\sum_{j=1}^{K}\theta_k f_k\left(y, \mathbf{x}\right)\right\} \tag{6}$$

McCallum et. al formulates the logisitic regression model, parameterized by weights $\theta \in \mathbb{R}^K$, in which a single set of weights is shared across all the classes [5]. This is acheived through a set of feature functions that are nonzero for a single class and is defined as follows:

$$\begin{aligned}f_{y',j}\left(y, \mathbf{x}\right) &= \mathbb{1}\{y' = y\}\cdot x_j \\ f_{y'}\left(y, \mathbf{x}\right) &= \mathbb{1}\{y' = y\}\end{aligned} \tag{7}$$

This notation, while not standard, mirrors our later formulation of Conditional Random Fields in Section 6.2.

For parameter estimation, we opted to use scikit-learn's implementation of the L-BFGS algorithm, which approximates the Broyden-Fletcher-Goldfarb-Shanno algoriothm [4]. This is a quasi-Newton optimzation method that apprixmates the Hessian matrix to reduce memory usage. Scikit-learn optimizes the L2 penalized multinomial loss [4] [3].

$$\min_{\theta}\ \frac{1}{2}\cdot\theta^T\theta - \sum_{i=1}^{N}\left[\sum_{c=1}^{C}\left[\mathbb{1}\{y_i = y_c\}\cdot\sum_{k=1}^{K}\theta_k f_k\left(y_i, \mathbf{x}_i\right)\right] - \log Z(\mathbf{x}_i)\right] \tag{8}$$

where y_i is the true label and y_c could be any label.

6 General Models

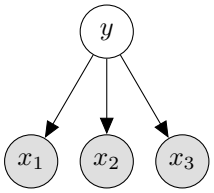
6.1 Bayesian Networks

6.2 Conditonal Random Fields (CRFs)

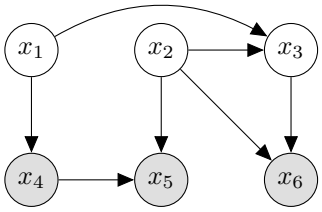
Conditional Random Fields (CRFs) are discriminative graphical models that can model output variables $\mathbf{y}$ conditioned on features $\mathbf{x}$ compactly, in contrast to generative models. CRFs are natural extensions to logistic regression for general graphs. We describe the model formulation, paramter estimation, and inference tasks as outlined by McCallum et. al [5], and introduced by Lafferty et. al [2].

CRFs model a conditional likelihood $p(\mathbf{y} \mid \mathbf{x})$ on a general graph $\mathcal{G}$, composed of a set of factors $F = \{\psi_a\}_{a=1}^{A}$. We reiterate McCallum's description of a CRF with paramter tying, where a set of factors can share a set of feature functions $\{f_{pk}\left(\mathbf{x}_c, \mathbf{y}_c\right)\}_{k=1}^{K(p)}$ and parameters $\theta_p \in \mathbb{R}^{K(p)}$ for a

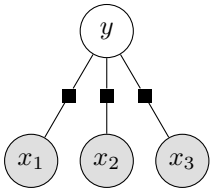
Generative: Naive Bayes



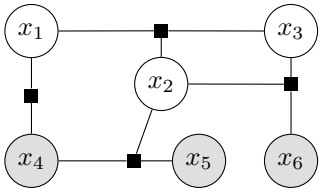
Generative: General Bayesian Networks



Discriminative: Logistic Regression



Discriminative: Conditional Random Fields



given *clique template* C_p. These clique templates come from paritioning the factors of graph $\mathcal{G}$ into $\mathcal{C} = \{C_1, C_2, \ldots, C_p\}$. Hence, our model is composed of the following:

$$\begin{aligned} p(\mathbf{y} \mid \mathbf{x}) &= \frac{1}{Z(\mathbf{x})} \prod_{C_p \in \mathcal{C}} \prod_{\psi_c \in C_p} \psi_c(\mathbf{x}_c, \mathbf{y}_c; \theta_p) \\ \psi_c(\mathbf{x}_c, \mathbf{y}_c; \theta_p) &= \exp\left\{ \sum_{k=1}^{K(p)} \theta_{pk} f_{pk}(\mathbf{x}_c, \mathbf{y}_c) \right\} \\ Z(\mathbf{x}) &= \sum_{\mathbf{y}} \prod_{C_p \in \mathcal{C}} \prod_{\psi_c \in C_p} \psi_c(\mathbf{x}_c, \mathbf{y}_c; \theta_p) \end{aligned} \tag{9}$$

where p is a factor index, f_{pk} is a feature function, and θ_{pk} is the weight associated with it. K is the number of features and $K(p)$ are the features found in factor p. Because our labels are discrete, we can write our feature functions f_pk in terms of observation functions q_{pk} to make *label-observation features*.

$$f_{pk}(\mathbf{y}_c, \mathbf{x}_c) = \mathbb{1}\{\mathbf{y}_c = \widetilde{\mathbf{y}}_c\} \cdot q_{pk}(\mathbf{x}_c) \tag{10}$$

Parameter estimation for CRFs can be done through maximum likelihood estimation, where we can minimize the conditional negative log likelihood:

$$\ell(\theta) = -\sum_{C_p \in \mathcal{C}} \sum_{\psi_c \in C_p} \sum_{k=1}^{K(p)} \theta_{pk} f_{pk}(\mathbf{x}_c, \mathbf{y}_c) + \log Z(\mathbf{x}) \tag{11}$$

Hence, the partial derivative with respect to our paramter θ_{pk} associated with the clique template C_p is:

$$\frac{\partial \ell}{\partial \theta_{pk}} = -\sum_{\psi_c \in C_p} f_{pk}(\mathbf{x}_c, \mathbf{y}_c) + \sum_{\psi_c \in C_p} \sum_{\mathbf{y'}_c} f_{pk}(\mathbf{x}_c, \mathbf{y'}_c) \cdot p(\mathbf{y'}_c \mid \mathbf{x}) \tag{12}$$

We can then minimize our cost function ℓ using L-BFGS optimization algorithm as described in Section 5.2.

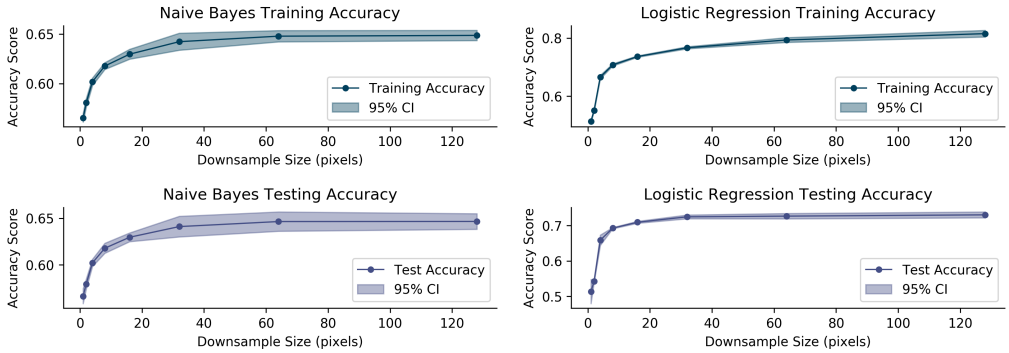


Figure 5: Training/Testing Accuracy Results for Baseline Models.

7 Experimental Task

8 Results: Parameter Estimation

8.1 Baseline

8.2 General

9 Results: Inference

9.1 Baseline

9.2 General

10 Conclusion

References

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

[2] J. Lafferty. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. pages 282–289. Morgan Kaufmann, 2001.

[3] K. P. Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.

[4] 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.

[5] C. Sutton and A. McCallum. An Introduction to Conditional Random Fields. *arXiv e-prints*, page arXiv:1011.4088, Nov 2010.

[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.