Histograms of Oriented Gradients for cats-dogs detection

Jiaxing Wu, Zixuan Yang, Ting Wang
Department of Software Engineering, LiaoNing University, LiaoNing 110031, China
Department of Computer Science And Technology Engineering, LiaoNing University, LiaoNing 110031, China
Department of Communication Engineering, LiaoNing University, LiaoNing 110031, China
17610059551@163.com

Abstract. We study the question based on grids of Histograms of Oriented Gradient (HOG) and support vector machine(SVM) ,adopting different kernel functions to distinguish cat from dog with robust visual object recognition [1]. We show experimentally that the quadratic descriptors significantly outperform others kernel functions and with the increasing number of the samples , the performance of the quadratic is getting better and better. Beside, in order to reduce the complexity, we also apply principal Component Analysis, which is known as pca, and we also study the influence of the number of features on performance, concluding that the line is quadratic.

1. Introduction

Histogram of Oriented Gradient, HOG, has attracted much attention in recent years, mainly because it is widely used in image recognition when it work with Support Vector Machine, SVM, which need to calculate support vector and generate a separation hyperplane in order to classify. [2] HOG is a feature that describe object through computer science and image processing by calculating and collecting some important information in the part of picture [5]. In a picture, a part of appearance and shape can be described in detail by the gradient or the density of oriented edge. So we can use these features that they all have to find their difference and divide them into different groups.

In this paper, we use the same number of dogs and cats pictures to train our program and we also select two ways to record the result, one is that use the sum of testing number of 20% from picture have been trained and the other one is that use the sum of testing number of 20% from picture are not trained. With increasing number of pictures from 100 to 2000, in each group, we experimentally test the accuracy in different types of kernel function.

1.1 Support Vector Machine Learning Algorithm

If we have a space ,and we also have the training sample:
\[ D = \{(x_1, y_1), (x_2, y_2), ..., (x_m, y_m)\}, y_i \in \{-1, +1\} \]

+1 is regard as positive samples and -1 is considered as some negative samples, we may think about how we divide the positive samples from the negative samples. Some people would say that a straight line could be a good choice, but which straight line is the question. So when it can be seen visually, it is clear that in the widest street that separates the positive samples from the negative samples should be
the chosen one, because the division of hyperplane has the best tolerance for local disturbance of training samples. For examples, as there are some the non-exclusion factors like boundedness and noise, the widest street is the least affected [6].

If we have got a vector \( \mathbf{w} \) any length constrained to be perpendicular to the medium line of the street, and then we also have some unknown and there is a vector \( \mathbf{u} \) that points to it, we will be interested in is whether or not that the vector \( \mathbf{u} \) is on the right side of the street(positive) or on the left side of the street(negative). So we can project that vector \( \mathbf{u} \) down on the \( \mathbf{w} \), because then we will have the distance in this direction and if the distance is as far as possible, the right of the street(positive) is easily arrived. So, in the space, the division of hyperplane can be described by the following linear equation:

\[
\mathbf{W} \cdot \mathbf{u} \geq c
\]

Among them, we take the \( \mathbf{w} \) and dot it with \( \mathbf{u} \), and the result whether or not is equal to or greater than some constant \( c \). The dot product is taking the projection onto \( \mathbf{w} \) and the bigger projection is the further out along this line, and eventually that it must crosses the medium line of the street(positive). So we can also convert the formula into:

\[
\mathbf{w} \cdot \mathbf{u} + b \geq 0, \quad c = -b \quad [1]
\]

If this formula is true, then it is a positive sample. From what has been discussed above is the decision rule.

We assume that hyperplane(\( \mathbf{w}, b \)) can classify the positive samples from negative samples, so when it comes to \( (x_i, y_i) \in D \). If \( y_i = +1 \), it is absolutely that \( \mathbf{w} \cdot \mathbf{u} + b \geq 0 \). If \( y_i = -1 \), it is also obvious that \( \mathbf{w} \cdot \mathbf{u} + b \leq 0 \). So

\[
\begin{align*}
\frac{\mathbf{w} \cdot \mathbf{x}_i + b \geq +1, y_i = +1}{\mathbf{w} \cdot \mathbf{x}_i + b \leq +1, y_i = -1}
\end{align*}
\]

\[
y_i (\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1, i = 1, 2, 3, ..., n \quad [2]
\]

Then we also need to know how to express the distance between the two gutters. First, we need get a vector in negative border(\( \mathbf{x}_- \)) and a vector in positive border(\( \mathbf{x}_+ \)), and then take the difference of those two vectors(\( \mathbf{x}_+ - \mathbf{x}_- \)), so the width is equal to

\[
\frac{(\mathbf{x}_+ - \mathbf{x}_-) \cdot \mathbf{w}}{\|\mathbf{w}\|} = \mathbf{x}_+ \cdot \frac{\mathbf{w}}{\|\mathbf{w}\|} - \mathbf{x}_- \cdot \frac{\mathbf{w}}{\|\mathbf{w}\|}
\]

\[
\begin{align*}
\mathbf{x}_+ \cdot \mathbf{w} &= 1 + b \\
- \mathbf{x}_- \cdot \mathbf{w} &= 1 - b
\end{align*}
\]

Finally, width is equal to

\[
y = \frac{2}{\|\mathbf{w}\|}
\]

The width is known as margin [7].

1.2 Histogram of Oriented Gradient

The feature of HOG, Histogram of Oriented Gradient, is a kind of feature descriptor which used as detecting object in the field of computer vision and image processing. Characteristics of local area are
formed by calculating and statistical histogram of gradient direction. Hog feature combined with SVM classifier has been widely used in image recognition, especially in pedestrian detection. We choose to use HOG because the descriptor provide excellent performance than other existing features. Since in an image, the appearance and shape of a local target can be well described by the directional density distribution of the gradient or edge. in other words, the essence is statistical gradient information and the gradient information is mainly located in the edge. \[4\]

Now, we can figure out how can we fulfill it. Before we extract the features, the pictures need to do some preprocessing prevent the noisy influence our result. Firstly, Graying the pictures and Using Gamma calibration to standardize the color space of the input image. \[3\] This aim is to reduce the light and noise impact, and adjust the contrast of image. Because color is no any special meaning, we turn them to gray.

Gamma formula:
\[I(x, y) = I(x, y)_{\text{gamma}}\]

And then, we calculate the gradient of abscissa and ordinate, and the direction of gradient of each pixel is calculated accordingly. The derivation operation can not only capture contour, shadow and some texture information, but also further weaken the influence of light.

So the gradient of pixel \((x, y)\) is:
\[G_x(x, y) = H(x + 1, y) - H(x - 1, y)\]
\[G_y(x, y) = H(x, y + 1) - H(x, y - 1)\]

In this formula, \(G_x(x, y), G_y(x, y), H(x, y)\) is respectively as the gradient of abscissa, the gradient of ordinate and pixel value. So the gradient amplitude and direction are:
\[G(x, y) = \sqrt{G_x(x, y)^2 + G_y(x, y)^2}\]
\[\alpha(x, y) = \tan^{-1}\left(\frac{G_y(x, y)}{G_x(x, y)}\right)\]

The common way is that we convolute the original image with the gradient operator \([-1,0,1]\) to get x direction gradient called “gradscalx” and use \([-1,0,1]^T\) to get y direction gradient called “gradscaly”. And then we can use the formula to calculate the gradient amplitude and direction. Besides, we also need divide the image into some small connected areas and these are known as “cell”. Then in cell unit, some gradient or edge orientation histogram of each pixel need to be collected. At last, in order to create feature descriptor, we combine this histogram \[2\].

1.3 Principle Component Analysis
For the problem of dimensionality reduction, by far the most popular, by the most commonly used algorithm is Principle Component Analysis which is often abbreviated PCA \[9\].

PCA is to find a lower-dimensional surface and onto project the data, so that the sum of squares of these projection error which is distance between points and projections is minimized. For instance, if we want to reduce from 2-dimension to 1-dimesion, we need to find a direction (a vector \(u^{(1)} \in \mathbb{R}^n\)) onto which to project the data so as to minimize the projection error. So if we want to reduce from n-dimension to k-dimesion, we need to find k vectors \(u^{(1)}, u^{(2)}, ..., u^{(k)}\) onto which to project the data so as to minimize the projection error. Before applying PCA, the standard practice is to first perform mean normalization and feature scaling so that the features \(X_1\) and \(X_2\) should have zero mean and should have comparable ranges of values and this we all have already done \[10\].
2. Experiment

Before making this model, we divide the group with different number, which increasing to 1500 pictures. In each group we ensure that the number of cat’s pictures and dog’s pictures is equal. After we confirm these modifications, we choose 20% samples which is not training before as testing samples to test many kinds of kernel function including linear, quadratic, polynomial, RBF and MLP. Besides, we also use the training samples to test accuracy. As the number of images grows, we only can find some regular patterns in linear and quadratic.

In addition, in order to improve the effect of searching, we select distinguish number of characteristic and we hope we can find a range that it is the optimum quantity which can make the accuracy the highest and reduce the influence of noise.

As is shown in the line graph, the ratio are quite distinguished from the different number of pictures. To be more exact, there are two nearly same upward trends in the initial number of pictures from 0.66 in 100 to 0.68 in 200. However, after these amount of pictures, the proportion in quadratic increases moderately from 0.7 in 300 to 0.7806 in 2000. When it regards to Linear, it is quite stable until 600, a sharp decrease can be seen from 0.65 in 600 to 0.5 in 800. There is a gap of over 15% in 2000 between Linear and Quadratic (0.5 and 0.7806 respectively).

The picture above is the introducing and the reasons for this support of this view are outlined as follows. The performance of the quadratic increases with the number of samples, because when the number of samples are rising, the nonlinearity of features strengthens. So the quadratic kernel is more applicable.

As is shown in the line graph, the ratio are quite distinguished from the different number of pictures. To be more exact, there are two nearly same upward trends in the initial number of pictures from 0.66 in 100 to 0.68 in 200. However, after these amount of pictures, the proportion in quadratic increases moderately from 0.7 in 300 to 0.7806 in 2000. When it regards to Linear, it is quite stable until 600, a sharp decrease can be seen from 0.65 in 600 to 0.5 in 800. There is a gap of over 15% in 2000 between Linear and Quadratic (0.5 and 0.7806 respectively).

The picture above is the introducing and the reasons for this support of this view are outlined as follows. The performance of the quadratic increases with the number of samples, because when the number of samples are rising, the nonlinearity of features strengthens. So the quadratic kernel is more applicable.
According to the another line graph, the accuracy rate witnesses both similarities and differences in the different number of pictures and features. More precisely, there are no great difference between 1000, 1500 and 2000 at 75, accounting for 0.7243, 0.7175 and 0.7339 respectively, after which they all increase slightly to the peak of 0.7298 in 1000, 0.7418 in 1500 and 0.7429 in 2000 at 150. However, a sharp decrease in any number of pictures is seen especially in 1000 from 0.7177 at 300 to 0.6252 at 500. Besides, it is also clear that the accuracy rate shows an upward trend with increasing number of pictures.

The picture above is the introducing and there are several points to be listed as follows to justify this phenomenon. Few feature numbers will obliterate some important information, while many of them will take too much noise. So it is clear that the line is quadratic. Besides, when there are many pictures especially the number of pictures are higher (twice or more) than the features number, it is a good choice to use pca. The reason is that it can help us to reduce the amount of calculation and make work more efficiently.

3. Conclusion:
As the number of pictures grows, it is reliable for us to use quadratic to distinguish the cats and dogs. Also, we find that the around 100 characteristic would be a good choice to improve searching effect and reduce the influence of noise.

References
HOG:
[1] Dalal N, Triggs B. Histograms of oriented gradients for human detection[C]//Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. IEEE, 2005, 1: 886-893. (2016:Google Citation: 14046)
[2] Vondrick C, Khosla A, Malisiewicz T, et al. Hoggles: Visualizing object detection features[C]//Proceedings of the IEEE International Conference on Computer Vision. 2013: 1-8. (Google Citation: 98)
[3] W. T. Freeman and M. Roth. Orientation histograms for hand gesture recognition. Intl. Workshop on Automatic Face and Gesture Recognition, IEEE Computer Society, Zurich, Switzerland, pages 296–301, June 1995.
[4] W. T. Freeman, K. Tanaka, J. Ohta, and K. Kyuma. Computer vision for computer games. 2nd International Conference on Automatic Face and Gesture Recognition, Killington, VT, USA, pages 100–105, October 1996.

SVM:

[5] Dengnaiyang and Tianyingjie(2009). Support vector machines: Theory \ algorithm and expansion. Science press \ Beijing.

[6] Joachims,T.(2006)."Training linear SVMs in linear time."In Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD),217-226,Philadelphia,PA.

[7] Tsang,I. W., J. T. Kwok, and P.Cheung.(2006)."Core vector machines:Fast SVM training on very large data sets." Journal of Machine Learning Research,6:363-392.

[8] Burges, C.J.C.(1998). “A tutorial on support vector machines for pattern recognition.” Data Mining and Knowledge Discovery, 2(1):121-167.

PCA:

[9] Yang,J., D.Zhang,A.F. Frangi, and J.-Y. Yang.(2004)."Two-dimensional PCA:A new approach to appearance-based face representation and recognition .” IEEE Transactions on Pattern Analysis and Machine Intelligence,26(1):131-137.

[10] Zhang,D. and Z.-H. Zhou. (2005).“(2D)^2 PCA:2-directional 2-dimensional PCA for efficient face representation and recognition.”Neurocomputing, 69(1-3):224-231.