Machine Learning for Computer Vision

Exercise 2

Kodai Matsuoke, Yuyan Li

May 5, 2017

1 Iterated Conditional Models

The missing code is:

```
# unary terms
energy += unaries[x0,x1,l]

# pairwise terms
energy += 4 - [labels[x0-1,x1], labels[x0+1,x1],
            labels[x0,x1-1], labels[x0,x1+1]].count(l)
```

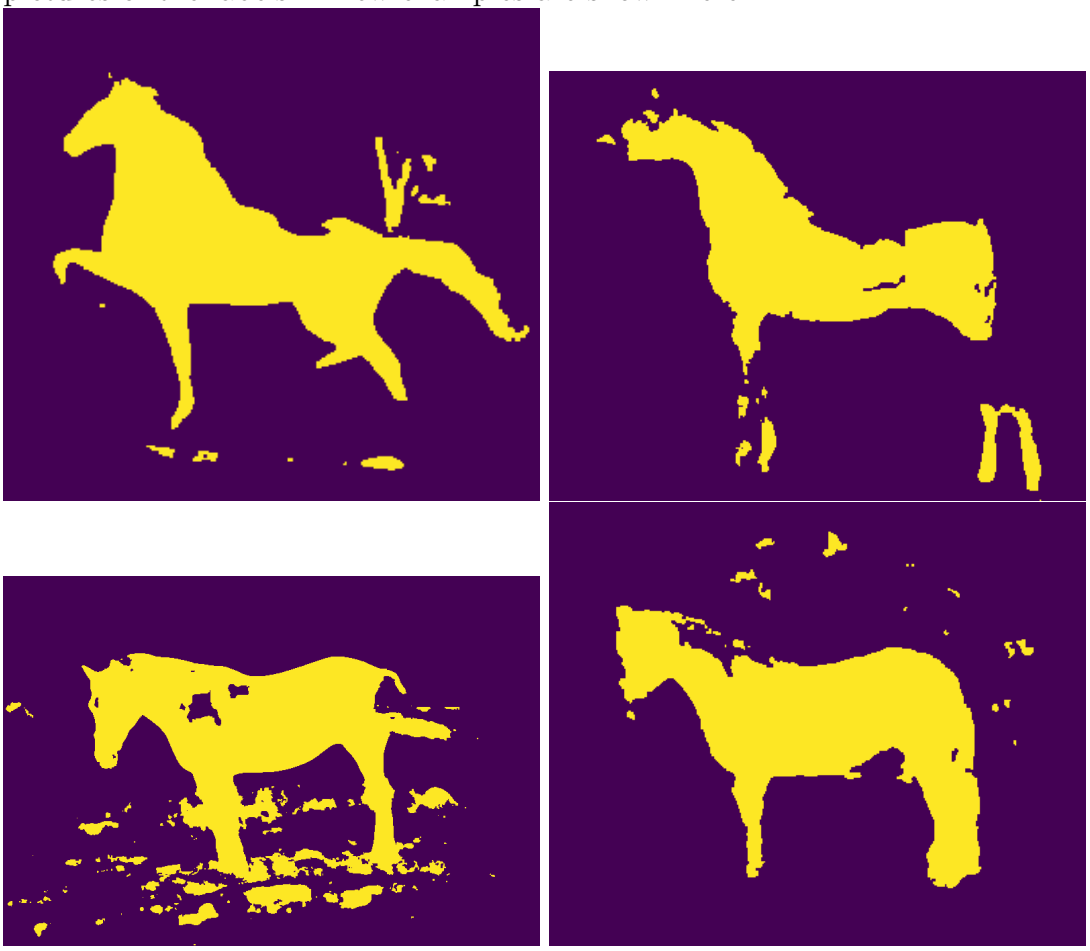
The code to use probability pictures as unaries is:

```
# import predictions from exercise1
# prediction images are in folder predictions/
pred_paths = glob.glob("predictions/*")
pred = [skimage.img_as_float(skimage.io.imread(f)) for f in pred_paths]

# Getting rid of the zeros
for x in numpy.nditer(pred[0], op_flags=['readwrite']):
    if x == 0:
        x[...] = 1e-100
    if x == 1:
        x[...] = 1. - 1e-16

fg = -numpy.log(pred[0])
bg = -numpy.log(1.-pred[0])
unaries = numpy.dstack((fg, bg))
```

Changing beta doesn't do anything since it isn't used in the function.
In the whole program (source below) there is also an addition at the end to produce pictures of the labels. A few examples are shown here:



Listing 1: icm.py

```
import numpy

import os
import glob
import skimage.io
from scipy import ndimage

def iterated_conditonal_modes(unaries, beta, labels = None):
    shape = unaries.shape[0:2]
    n_labels = unaries.shape[2]

    if labels is None :
        labels = numpy.argmin(unaries, axis=2)
```

```
    continue_search = True
    while(continue_search):
        continue_search = False
        for x0 in range (1, shape[0]-1):
            for x1 in range(1 ,shape[1]-1):

                current_label = labels[x0,x1]
                min_energy = float('inf')
                best_label = None

                for l in range(n_labels):
                    # evaluate cost
                    energy = 0.0

                    # unary terms
                    energy += unaries[x0,x1,l]

                    # pairwise terms
                    energy += 4 - [labels[x0-1,x1], labels[x0+1,x1],
                                   labels[x0,x1-1], labels[x0,x1+1]].coun

                    if energy < min_energy:
                        min_energy = energy
                        best_label = l

                if best_label != current_label:
                    labels [x0 ,x1] = best_label
                    continue_search = True
        return labels

if __name__ == "__main__":
    import matplotlib.pyplot as plt

    shape = [100, 100]
    n_labels = 2

    # regularizer strength
    beta = 0.01

    # unaries
    # unaries = numpy.random.rand(shape[0], shape[1], n_labels )

    # import predictions from exercise1
```

```
    # prediction images are in folder predictions/
    pred_paths = glob.glob("predictions/*")
    pred = [skimage.img_as_float(skimage.io.imread(f)) for
                f in pred_paths]

    # Getting rid of the zeros
    for x in numpy.nditer(pred[0], op_flags=['readwrite']):
        if x == 0:
            x[...] = 1e-100
        if x == 1:
            x[...] = 1. - 1e-16

    fg = -numpy.log(pred[0])
    bg = -numpy.log(1.-pred[0])
    unaries = numpy.dstack((fg, bg))

    labels = iterated_conditonal_modes(unaries, beta=beta)

    # plt.imshow(labels)
    # plt.show()

    index = 0
    for p in pred:
        # Getting rid of zeros for the log
        for x in numpy.nditer(p, op_flags=['readwrite']):
            if x == 0:
                x[...] = 1e-100
            if x == 1:
                x[...] = 1. - 1e-16

        fg = -numpy.log(p)
        bg = -numpy.log(1.-p)
        unaries = numpy.dstack((fg, bg))
        labels = iterated_conditonal_modes(unaries, beta=beta)
        plt.imsave('label%d.png'%index, labels)
        index += 1
```

2 Higher order factors

The domain of x_z is $0, 1, 2, 3, 4, 5, 6, 7$. Each variable value represents one energy state. The pairwise factors are given in the following table:

x_z	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
x_0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
ϕ_{0z}	a	b	c	d			∞				∞		e	f	g	h

x_z	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
x_1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
ϕ_{1z}	0	0		∞		0	0		∞			∞		0	0	

x_z	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
x_1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
ϕ_{1z}	0	∞	0	∞	0	∞	0	∞	0	∞	0	∞	0	∞	0	∞