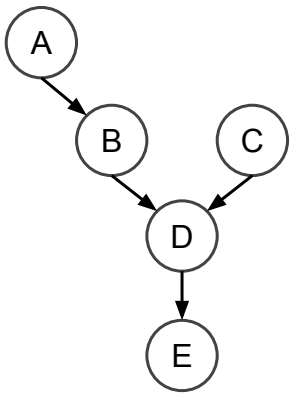
CSE 527, Fall 2017
Problem Set #3

(Due Nov 27th 11:59pm)

---

1. **[20 points] Conditional Independence in Bayesian Networks**

   Given the Bayes net below, answer if the following statements are **true** or **false**.

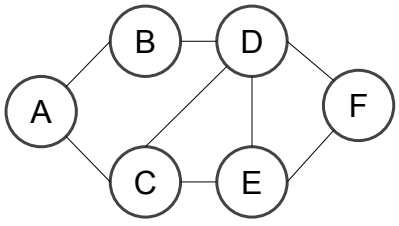


   (a) **[10 points]** $A \perp\!\!\!\perp C \mid B$

   (b) **[10 points]** $B \perp\!\!\!\perp C \mid D$

2. **[20 points] Conditional Independence in Undirected Graphical Models**

   Given the undirected graph (a.k.a. Markov network or Markov random field) below, answer if the following statements are **true** or **false**.



   (a) **[10 points]** $A \perp\!\!\!\perp F \mid B, C$

   (b) **[10 points]** $A \perp\!\!\!\perp D \mid B$

3. **[20 points] Graphical LASSO**

   We learned that the graphical Lasso method aims to find the value of θ that maximizes $f(\theta)$ below:

   $$f(\theta) = log(det(\theta)) - tr(S\theta) - \lambda||\theta||_1.$$

   Derive this objective function based on the probability density function of the multivariate Gaussian distribution and maximum likelihood estimation (MLE).

   Hint 1: Start with $ln(p(X|\mu, \Sigma)) = \sum_{i=1}^{N} ln(p(x_i|\mu, \Sigma))$

   Hint 2: Use the fact that $det(A)det(A^{-1}) = 1$.

4. **[40 points] Deep Learning**

   In class, we learned how to use Keras, a high-level neural networks Python API, to build your own multi-layer perceptron (MLP) model to classify MNIST digits. For this problem, you will build a convolutional neural net (CNN) with Keras and see how the performance compares to that of MLP's. If you prefer to implement this in languages other than Python without Keras, you are more than welcome to do so. Make sure you submit your code with your writeup.

   Getting ready for the questions

   (a) **Installing Python**
   If you not yet have Python installed on your machine, we recommend you to try out Anaconda (https://anaconda.org/). Anaconda is a python distribution that comes with various modules that are useful for scientific computing (e.g. numpy, scipy, matplotlib). Under the "Download Anaconda" tab, you can find an installer that is compatible with your OS. Make sure to get the Python 3.* version.

   (b) **Installing Keras**
   The easiest way to install Keras is to use "pip" or "pip3". In your terminal/command prompt, simply type "(sudo) pip install tensorflow" to install tensorflow, and then type "(sudo) pip install keras" to install Keras. This will install keras with the tensorflow backend.

   (c) **MNIST Data**
   You can download "mnist.data.tar.gz" and "mnist.labels.tar.gz" from our course website. When extracted, "mnist.data.tar.gz" is a csv file that contains a matrix of size 55000 by 784. Each row represents a single MNIST image. It can be reshaped into a matrix of size 28 by 28 to recover the original image. Use first 50000 images as a training set and the rest as a validation set.

Questions

(a) **[5 points]** Plot a random image from MNIST data to see if you are loading the data correctly.

(b) **[15 points]** Implement and run a network of multi-layer perceptions with a single hidden layer containing 15 nodes. You can refer to the implementation shown in class. Plot training and validation loss for about 10 epochs. Feel free to experiment with different activation functions, regularization methods, and architectures.

(c) **[15 points]** Implement and run a convolutional neural net with the following architecture; the first layer is a 2D convolution layer with 32 filters (size: 3 by 3, stride: 2 by 2). The second layer is a 2D convolution with 16 filters (size: 3 by 3, stride: 2 by 2). The third layer is a fully-connected layer with 10 softmax outputs. This third layer is exactly the same as the last layer of our MLP implementation. Plot training and validation loss for about 10 epochs. Feel free to experiment with different activation functions, regularization methods, and architectures. Does your CNN implementation perform better than your MLP implementation?

(d) **[5 points]** Name two advantages of CNNs over MLPs in short sentences.