Machine Learning

Homework 5

Not collected, not graded.

1 AdaBoost

1. What is a weak classifier?
2. What is a strong classifier?
3. How does AdaBoost select the weak classifiers?
4. What is the importance of modifying the sample weights between weak classifier selection?

2 K-means clustering

We are given points $\mathbf{x}_1, \dots, \mathbf{x}_N \in \mathbb{R}^D$ and an integer $K > 1$, and our goal is to minimize the within cluster variance:

$$J(\{\boldsymbol{\mu}_k\}, \{l_n\}) = \sum_{n=1}^{N} |\mathbf{x}_n - \boldsymbol{\mu}_{l_n}|^2$$

where $\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_K \in \mathbb{R}^D$ are the K cluster centers, and $l_1, \dots, l_N \in \{1, \dots, K\}$ are the cluster assignments. You have seen the popular Lloyd's algorithm in class, to find an approximate solution by optimizing $\{\boldsymbol{\mu}_k\}$ and $\{l_n\}$, alternately.

1. Show that Lloyd's algorithm is always guaranteed to converge (=stop) in a finite number of steps. For simplicity assume that $\forall n \in \{1, \dots, N\}: \forall j \neq k: |\mathbf{x}_n - \boldsymbol{\mu}_k|^2 \neq |\mathbf{x}_n - \boldsymbol{\mu}_j|^2$, i.e. there will never be a tie to be broken when assigning clusters. *Hint*: think about how many possible cluster assignments there are, and how the objective function $J(\{\boldsymbol{\mu}_k\}, \{l_n\})$ changes at each step.
2. Does this mean that Lloyd's algorithm always converges to the global optimum?
3. We have assumed that the number of clusters, K, is being revealed to us by some benevolent oracle (in other words: picking the right K is a difficult problem). Explain how the exact minimum of the K-means objective function behaves (on any dataset) as we increase K from 1 to N.
4. If we were to run K-means on a given dataset for all K from 1 to N, and plot the objective function value for each K, could you think of a strategy to deduce the optimal number of clusters from that plot?

3 Expectation–Maximization for Gaussian mixture models

A Gaussian mixture model is a family of distributions of the following form:

$$p(\mathbf{x}) := \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

where the mixture weights satisfy

$$\sum_{k=1}^{K} \pi_k = 1, \qquad \pi_k \geq 0, \quad k \in \{1, \dots, K\}$$

We introduce a latent variable $z_n \in \{1, \dots, K\}$, that encodes the class assignment of sample n.
Starting from an initialization for $\boldsymbol{\mu}_k$, $\boldsymbol{\Sigma}_k$, and π_k, $k \in \{1, \dots, K\}$, E-M iterates the following two steps:

1. E-step:
$$\gamma_{n,j} := p\left(z_n = j \mid \mathbf{x}_n, \{(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \pi_k)\}_{k=1}^K\right)$$
i.e., we estimate the "responsibility" $\gamma_{n,j}$ of component j for sample n.

2. M-step:
$$\{(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \pi_k)\}_{k=1}^K = \arg\max \left\{ \sum_{n=1}^N \sum_{k=1}^K \gamma_{n,k} \ln\left(\pi_k \mathcal{N}(\mathbf{x}_n \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\right) \right\}$$
i.e., we maximize the parameters of each component k considering the n samples weighted by the responsibility $\gamma_{n,k}$.

1. Derive the M-step updates for $\boldsymbol{\mu}_k$.

2. Derive the M-step updates for π_k. *Hint*: remember the summation constraint on π_k!

3. Consider a simplified Gaussian mixture model, where all components share the same covariance matrix $\boldsymbol{\Sigma}_k = \boldsymbol{\Sigma}$. Derive the update rule for $\boldsymbol{\Sigma}$ in the M-step. This answer can rely on the value $\boldsymbol{\mu}_k$ already estimated at the current M-step.

4. Consider an even simpler Gaussian mixture model, where the shared covariance matrix takes the isotropic form, $\boldsymbol{\Sigma} = \sigma^2 I_D$, for some $\sigma^2 > 0$ *known*. Given the data, we estimate the means $\{\boldsymbol{\mu}_k\}$ of each component and the mixture weights $\{\pi_k\}$ with E-M. Assume that

   - the mixture weights $\{\pi_k\}$ are bounded away from zero, i.e. $\exists \varepsilon > 0$ such that $\forall k \in \{1, \ldots, K\}: \pi_k \geq \varepsilon$, throughout iterations.
   - there will be no ties throughout the iterations: $\forall n \in \{1, \ldots, N\}: \forall j \neq k: |\mathbf{x}_n - \boldsymbol{\mu}_k|^2 \neq |\mathbf{x}_n - \boldsymbol{\mu}_j|^2$

   Show that as $\sigma^2 \to 0$, the E-step converges to the update rule for the labels l in Lloyd's algorithm, i.e., the soft assignment becomes hard. *Hint*: the responsibility is computed like a "soft-max"...

4 Kernel lego

1. Consider the candidate kernel $k(\mathbf{x}, \mathbf{y}) := (c + \mathbf{x}^T \mathbf{y})^2$, $c > 0$.
   (a) Using the lego table seen in class (also: PRML p. 296), show that this is an actual kernel.
   (b) Determine the corresponding mapping $\mathbf{x} \mapsto \phi(\mathbf{x})$.

2. Consider the kernel $k(\mathbf{x}, \mathbf{y}) := (\mathbf{x}^T \mathbf{y})^3$. Find the corresponding mapping $\mathbf{x} \mapsto \phi(\mathbf{x})$.

3. Can you find a general expression for the mappings $\mathbf{x} \mapsto \phi(\mathbf{x})$ corresponding to kernels
$$k(\mathbf{x}, \mathbf{y}) := (\mathbf{x}^T \mathbf{y})^M$$
for $M = 1, 2, \ldots$?

4. Sketch a proof for $\exp(\mathbf{x}^T \mathbf{y})$ being a valid kernel. Can you sketch a corresponding (infinite dimensional) mapping?

5. The radial basis function kernel, $k(\mathbf{x}, \mathbf{y}) := \exp(-\gamma \|\mathbf{x} - \mathbf{y}\|^2)$ is very popular. Can you prove it is an actual kernel? *Hint*: All "lego"-rules seen in class are fair game. It may be helpful to expand $\exp$ into a series.

6. What about the more complete version, the "Gaussian" kernel, $k(\mathbf{x}, \mathbf{y}) := \exp(-\frac{1}{2}(\mathbf{x} - \mathbf{y})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \mathbf{y}))$?

5 Gaussian processes for regression

Consider the regression model $t_n = y_n + \varepsilon_n$, where y_n is the model prediction and ε is Gaussian noise with precision β. We want to reproduce and re-derive in detail the results seen in class.

1. Give the likelihood $p(t_n \mid y_n)$. *Hint*: it's Gaussian...
2. Looking at an entire set of $\mathbf{t} = (t_1, \dots, t_N)^T$ and $\mathbf{y} = (y_1, \dots, y_N)^T$, give their joint likelihood $p(\mathbf{t} \mid \mathbf{y})$ **as a multivariate pdf**. *Hint*: it's still Gaussian. Consider the samples to be independent.
3. The principal idea of the Gaussian process tells us that $p(\mathbf{y}) = \mathcal{N}(\mathbf{y} \mid 0, \mathbf{K})$, where $\mathbf{K}$ is the kernel-based Gram matrix, with $k_{mn} = k(\mathbf{x}_m, \mathbf{x}_n)$. Compute the marginal $p(\mathbf{t}) := \int p(\mathbf{t} \mid \mathbf{y}) p(\mathbf{y})\, d\mathbf{y}$.
4. Extend the previous result to give the $(N+1)$-result for $p(t_{(N+1)}, \mathbf{t})$ seen in class.
5. Based on the joint $p(t_{(N+1)}, \mathbf{t})$, use pages 87-90 of PRML to get the conditional $p(t_{(N+1)} \mid \mathbf{t})$, which will allow to make predictions for new $\mathbf{x}_{(N+1)}$.