Exam 1 solutions

COMS 4771 Spring 2015

The order of the problems in your exam might've been different from what it is here.

Problem 1 (Training algorithms)

Online Perceptron (1) $\rightarrow C$**.** Neither SVM nor ERM will return a classifier with non-zero training error when the data is linearly separable by a homogeneous linear classifier.

ERM for homogeneous linear classifiers (2) $\rightarrow A$**.** This algorithm only considers homogeneous linear classifiers and returns a linear classifier with zero training error whenever one exists.

SVM (3) $\rightarrow B$**.** Process of elimination (and B is the hyperplane with maximum margin).

Problem 2 (Costs)

Part (a)

for $\ell = 1, 2, \dots, 20$ **do**

Let H_ℓ be the number of examples $(\boldsymbol{x}, y) \in S_A$ with $\hat{\ell}(\boldsymbol{x}) = \ell$ and $y = 1$.

Let T_ℓ be the number of examples $(\boldsymbol{x}, y) \in S_A$ with $\hat{\ell}(\boldsymbol{x}) = \ell$ and $y = 0$.

Let $\hat{p}'(\ell) := \mathbb{1}\{H_\ell > 10T_\ell\}$.

end for

The training cost of $\hat{f}'$ (w.r.t. S_A) can be written as

$$\begin{aligned}
\operatorname{cost}(\hat{f}', S_A) &= \frac{1}{|S_A|} \sum_{\ell=1}^{20} \sum_{\substack{(\boldsymbol{x},y) \in S_A: \\ \hat{\ell}(\boldsymbol{x}) = \ell}} \left(10 \cdot \mathbb{1}\{\hat{p}'(\ell) = 1 \, \wedge \, y = 0\} + \mathbb{1}\{\hat{p}'(\ell) = 0 \, \wedge \, y = 1\}\right) \\
&= \frac{1}{|S_A|} \sum_{\ell=1}^{20} \left(10 \cdot T_\ell \cdot \mathbb{1}\{\hat{p}'(\ell) = 1\} + H_\ell \cdot \mathbb{1}\{\hat{p}'(\ell) = 0\}\right).
\end{aligned}$$

So for each $\ell = 1, 2, \dots, 20$, we set $\hat{p}'(\ell)$ to the prediction that minimizes the training cost.

Part (b)

No, $\operatorname{cost}(\hat{f}', S_B)$ **is not generally an unbiased estimator of** $\operatorname{cost}(\hat{f}')$**.** The rest of the classifier (in particular, the splitting rules determining $\hat{\ell}$) depends on S_B.

Problem 3 (SVM)

Part (a)

No, they are not the same. Soft-margin SVM always has a solution, even with $\lambda = 1$. Hard-margin SVM sometimes has no solution—in particular, when the training data is not linearly separable.

Part (b)

Recall that when the training data are linearly separable, the solution to the SVM optimization problem is unique.

(i) **Yes, they are the same.** Shifting the feature vectors and the max-margin hyperplane by a fixed displacement vector doesn't change which side of the hyperplane the feature vectors lie on.

(ii) **Yes, they are the same.** All feature values are scaled down by a factor of 10, so weights just need to be multiplied by a factor of 10 to compensate.

(iii) **Yes, they are the same.** This is essentially a special case of (i) above ...

Problem 4 (MLE)

Part (a)

The derivative of the log-likelihood w.r.t. λ is

$$\begin{aligned}\frac{\partial}{\partial\lambda}\left\{\sum_{i=1}^{n}\ln\frac{\lambda^{x_i}e^{-\lambda}}{x_i!}\right\} &= \sum_{i=1}^{n}\frac{\partial}{\partial\lambda}\Big\{x_i\ln(\lambda)-\lambda-\ln(x_i!)\Big\}\\ &= \sum_{i=1}^{n}\Big\{\frac{x_i}{\lambda}-1\Big\}.\end{aligned}$$

This is zero when $\lambda = \frac{1}{n}\sum_{i=1}^{n} x_i$. So the MLE of λ given $x_1, x_2, \ldots, x_n$ is $\hat{\lambda} = \frac{1}{n}\sum_{i=1}^{n} x_i$.

Part (b)

$$\begin{aligned}\mathbb{E}(\hat{\lambda}) &= \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}(X_i) && \text{(by linearity of expectation)}\\ &= \mathbb{E}(X_1) && \text{(since } X_1, X_2, \ldots, X_n \text{ are i.i.d.)}\\ &= \lambda && \text{(since } X_1 \sim P_\lambda\text{).}\end{aligned}$$

This last step uses the fact that the mean of a Poisson random variable is equal to the rate parameter. (It's fine to leave it at $\mathbb{E}(\hat{\lambda}) = \mathbb{E}(X_1)$—this says that $\hat{\lambda}$ is an unbiased estimator of the mean, which happens to be λ.)

Problem 5 (Optimization)

$$\begin{aligned} \min_{\boldsymbol{w} \in \mathbb{R}^2} \quad & \sum_{i=1}^{n} (x_1^{(i)} - w_1)^2 + (x_2^{(i)} - w_2)^2 \\ \text{s.t.} \quad & w_1^2 + w_2^2 - 1 \leq 0. \end{aligned}$$

Yes, this is a convex optimization problem. The functions of the form $\boldsymbol{w} \mapsto (x_1^{(i)} - w_1)^2$, $\boldsymbol{w} \mapsto (x_2^{(i)} - w_2)^2$, $\boldsymbol{w} \mapsto w_1^2$, and $\boldsymbol{w} \mapsto w_2^2$ are convex because they are compositions of the convex function $z \mapsto z^2$ with an affine transformation. The objective and constraint functions are just sums of these convex functions (and possibly a constant -1), and hence themselves are convex.

Problem 6 (Linearity)

Part (a)

Yes, this is a linear classifier. Let λ_k be the rate parameter of the class conditional distribution for class $k \in \{0, 1\}$, let π_k be the class prior for class $k \in \{0, 1\}$. (I have changed the label -1 to 0 to make things more readable.) Then the plug-in classifier predicts 1 on input x precisely when

$$\pi_1 \frac{\lambda_1^x e^{-\lambda_1}}{x!} > \pi_0 \frac{\lambda_0^x e^{-\lambda_0}}{x!}.$$

Taking log of both sides and re-arranging, we see that the above is equivalent to

$$x \underbrace{\ln \frac{\lambda_1}{\lambda_0}}_{w} > \underbrace{\lambda_1 - \lambda_0 + \ln \frac{\pi_0}{\pi_1}}_{\theta}.$$

This is precisely the form of a linear classifier.

Part (b)

Yes, this is a linear classifier. There is a linear separator (the line where $x_2 = -x_1$) that separates $\mathbb{R}^2$ into two halves: the points whose NN (among the training data) is either Example 1 or Example 3 (both of which have a positive label), and the points whose NN is either Example 2 or Example 4 (both of which have a negative label).

Problem 7 (Kernels)

input $\tilde{\boldsymbol{x}} \in \mathbb{R}^d$.

  initialize min_distance $:= +\infty$, $\hat{y} :=$ (any default label).

  for $(\boldsymbol{x}, y) \in S$ **do**

    $\rho(\boldsymbol{x}, \tilde{\boldsymbol{x}}) := K(\boldsymbol{x}, \boldsymbol{x}) - 2K(\boldsymbol{x}, \tilde{\boldsymbol{x}}) + K(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{x}})$.

    if $\rho(\boldsymbol{x}, \tilde{\boldsymbol{x}}) <$ min_distance **then**

      min_distance $:= \rho(\boldsymbol{x}, \tilde{\boldsymbol{x}})$.

      $\hat{y} := y$.

    end if

  end for

  return $\hat{y}$.

This is self-explanatory upon observing that $\rho(\boldsymbol{x}, \tilde{\boldsymbol{x}}) = \|\boldsymbol{\phi}(\boldsymbol{x}) - \boldsymbol{\phi}(\tilde{\boldsymbol{x}})\|_2^2$, the squared Euclidean distance between $\boldsymbol{\phi}(\boldsymbol{x})$ and $\boldsymbol{\phi}(\tilde{\boldsymbol{x}})$. To see why this is true, we use the feature map associated with the kernel function:

$$\begin{aligned}
K(\boldsymbol{x}, \boldsymbol{x}) - 2K(\boldsymbol{x}, \tilde{\boldsymbol{x}}) + K(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{x}}) &= \langle \boldsymbol{\phi}(\boldsymbol{x}), \boldsymbol{\phi}(\boldsymbol{x}) \rangle - 2\langle \boldsymbol{\phi}(\boldsymbol{x}), \boldsymbol{\phi}(\tilde{\boldsymbol{x}}) \rangle + \langle \boldsymbol{\phi}(\tilde{\boldsymbol{x}}), \boldsymbol{\phi}(\tilde{\boldsymbol{x}}) \rangle \\
&= \langle \boldsymbol{\phi}(\boldsymbol{x}) - \boldsymbol{\phi}(\tilde{\boldsymbol{x}}), \boldsymbol{\phi}(\boldsymbol{x}) - \boldsymbol{\phi}(\tilde{\boldsymbol{x}}) \rangle \\
&= \|\boldsymbol{\phi}(\boldsymbol{x}) - \boldsymbol{\phi}(\tilde{\boldsymbol{x}})\|_2^2.
\end{aligned}$$