

1 Gaussian Task

Consider a supervised learning problem where n samples $S = \{X_i, Y_i\}_{i=1}^n$ are drawn from $P(\mathcal{X}, \mathcal{Y})$. We consider a family of classification problems

$$P(x, y) = \delta(y = 1)X_1 + \delta(y = -1)X_{-1}$$

where $X_1 \sim \mathcal{N}(\mu + \Delta, 1)$ and $X_{-1} \sim \mathcal{N}(-\mu + \Delta, 1)$.

We have n samples from a task with $\mu = 1, \Delta = 0$; We aim to generalize on this task. In addition, we have access to m samples from a task with $\mu = 1, \Delta > 0$. Let P_t and P_{ood} denote the two tasks respectively.

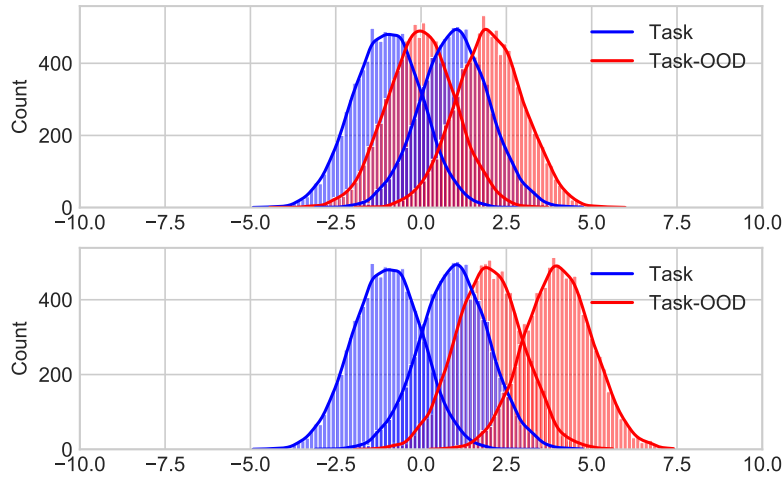


Figure 1: The two figures correspond to two different settings for the “Gaussian OOD data experiment” **(top)** Two tasks with very small Δ , which should make transfer easier **(bottom)** Two tasks with larger Δ where the transfer is expected to be smaller.

We are interested in understanding the behavior of learning from multiple tasks algorithms when m , n and Δ are varied.

2 Analysis of Algorithms that use OOD Data

We consider the linear hypothesis space H that splits the data into two classes using a single point. Formally, $f \in H$ is a hypothesis such that

$$f(x) = \begin{cases} y = 1 & x > h \\ y = -1 & x \leq h \end{cases}.$$

Every $f \in H$ corresponds to a scalar $h \in \mathbb{R}$. We will henceforth use f and h interchangeably and overload notation for h .

An OOD algorithm $A : (\mathcal{X}_t \times \mathcal{Y}_t)^n \times (\mathcal{X}_{\text{ood}} \times \mathcal{Y}_{\text{ood}})^m \mapsto \mathcal{H}$, maps n samples from the target task and m samples from an out-of-distribution task to a hypothesis that aims to minimize the population risk

$$e_t(h) = \mathbb{E}_{(x,y) \sim P_t} [\mathbf{1}[h(x) \neq y]].$$

Different draws of the training set S yield different hypotheses. Hence, we average over all possible draws of the training set to arrive at the objective

$$\mathcal{E}(A) = \mathbb{E}_{S \sim P^n} [e_t(A(S))].$$

2.1 Computing \mathcal{E}

Let ϕ denote the density of the standard normal and let Φ denote the CDF of the standard normal. The population risk of hypothesis h is denoted by

$$e_t(h) = \frac{1}{2} ((1 - \Phi(h + \mu)) + \Phi(h - \mu)).$$

Assume that $\bar{h} = A(S)$ is a normally distributed random variable. We are interested in minimizing

$$\begin{aligned} \mathbb{E}_{\bar{h}}[e_t(\bar{h})] &= \mathbb{E}_{\bar{h}} \left[\frac{1}{2} ((1 - \Phi(h + \mu)) + \Phi(h - \mu)) \right] \\ &= \int_{-\infty}^{\infty} \left[\frac{1}{2} ((1 - \Phi(h + \mu)) + \Phi(h - \mu)) \right] \phi \left(\frac{h - \bar{\mu}}{\bar{\sigma}} \right) dh \\ &= \int_{-\infty}^{\infty} \frac{1}{2} [(1 - \Phi(y\bar{\sigma} + \bar{\mu} + \mu)) + \Phi(y\bar{\sigma} + \bar{\mu} - \mu)] \phi(y) dy \\ &= \frac{1}{2} - \frac{1}{2} \int_{-\infty}^{\infty} \Phi(y\bar{\sigma} + \bar{\mu} + \mu) \phi(y) dy + \frac{1}{2} \int_{-\infty}^{\infty} \Phi(y\bar{\sigma} + \bar{\mu} - \mu) \phi(y) dy \end{aligned}$$

We use the identity

$$\int_{-\infty}^{\infty} \Phi \left(\frac{x - a}{b} \right) \phi(x) dx = \Phi \left(\frac{-a}{\sqrt{1 + b^2}} \right).$$

We can re-write the identity into a more convenient expression

$$\int_{-\infty}^{\infty} \Phi(cx + d) \phi(x) dx = \int_{-\infty}^{\infty} \Phi \left(\frac{x + d/c}{1/c} \right) \phi(x) dx = \Phi \left(\frac{d}{\sqrt{1 + c^2}} \right).$$

Using the above

$$\begin{aligned} \mathbb{E}_{\bar{h}}[e_t(\bar{h})] &= \frac{1}{2} - \frac{1}{2} \int_{-\infty}^{\infty} \Phi(y\bar{\sigma} + \bar{\mu} + \mu) \phi(y) dy + \frac{1}{2} \int_{-\infty}^{\infty} \Phi(y\bar{\sigma} + \bar{\mu} - \mu) \phi(y) dy \\ &= \frac{1}{2} - \frac{1}{2} \Phi \left(\frac{\bar{\mu} + \mu}{\sqrt{1 + \bar{\sigma}^2}} \right) + \frac{1}{2} \Phi \left(\frac{\bar{\mu} - \mu}{\sqrt{1 + \bar{\sigma}^2}} \right) \end{aligned} \tag{1}$$

We optimize the objective using brute-force search. The variables to optimize are $\bar{\mu}$ and $\bar{\sigma}$.

2.2 Single-head Classifier: LDA

LDA finds a linear sub-space that maximizes intra-class variance while minimizing inter-class variance. It shares similarities to PCA, and can be understood to be its "label-aware" counterpart. This blog post is a good resource on the topic.

LDA can also be used as a classifier although this is less common. Consider a binary classification problem and assume $P(y = 1|x)$ and $P(y = -1|x)$ are both Gaussians with identical variances (Σ). The classifier is given by:

$$f(x) = \mathbf{1}(w \cdot x > c)$$

where

$$w = \Sigma^{-1}(\bar{\mu}_1 - \bar{\mu}_{-1}) \quad \text{and} \quad c = w \cdot \frac{\bar{\mu}_1 + \bar{\mu}_{-1}}{2}$$

For the 1-dimensional case, the above classifier reduces to

$$f(x) = \mathbf{1}\left(\frac{\bar{\mu}_1 + \bar{\mu}_{-1}}{2}\right)$$

Consider tasks P_t and P_{ood} and samples from two tasks S_t and S_{ood} . We "mix" the two datasets to get

$$S = \alpha S_t + (1 - \alpha) S_{ood}.$$

This affects the calculation of class means, which are random variables distributed as follows:

$$\begin{aligned} \bar{\mu}_{+1} &\sim \mathcal{N}\left(\frac{+\alpha n \mu + (1 - \alpha)m(+\mu + \Delta)}{\alpha n + (1 - \alpha)m}, \sqrt{\frac{2((1 - \alpha)^2 m + \alpha^2 n)}{(\alpha n + (1 - \alpha)m)^2}}\right), \\ \bar{\mu}_{-1} &\sim \mathcal{N}\left(\frac{-\alpha n \mu + (1 - \alpha)m(-\mu + \Delta)}{\alpha n + (1 - \alpha)m}, \sqrt{\frac{2((1 - \alpha)^2 m + \alpha^2 n)}{(\alpha n + (1 - \alpha)m)^2}}\right). \end{aligned}$$

Hence,

$$\bar{h} = \frac{\bar{\mu}_1 + \bar{\mu}_{-1}}{2} \sim \mathcal{N}\left(\frac{(1 - \alpha)m\Delta}{\alpha n + (1 - \alpha)m}, \frac{\sqrt{(1 - \alpha)^2 m + \alpha^2 n}}{(\alpha n + (1 - \alpha)m)}\right)$$

or in other words $\bar{\mu} = \frac{(1 - \alpha)m\Delta}{\alpha n + (1 - \alpha)m}$ and $\bar{\sigma}^2 = \frac{(1 - \alpha)^2 m + \alpha^2 n}{(\alpha n + (1 - \alpha)m)^2}$.

fig. 2 includes results on this algorithm.

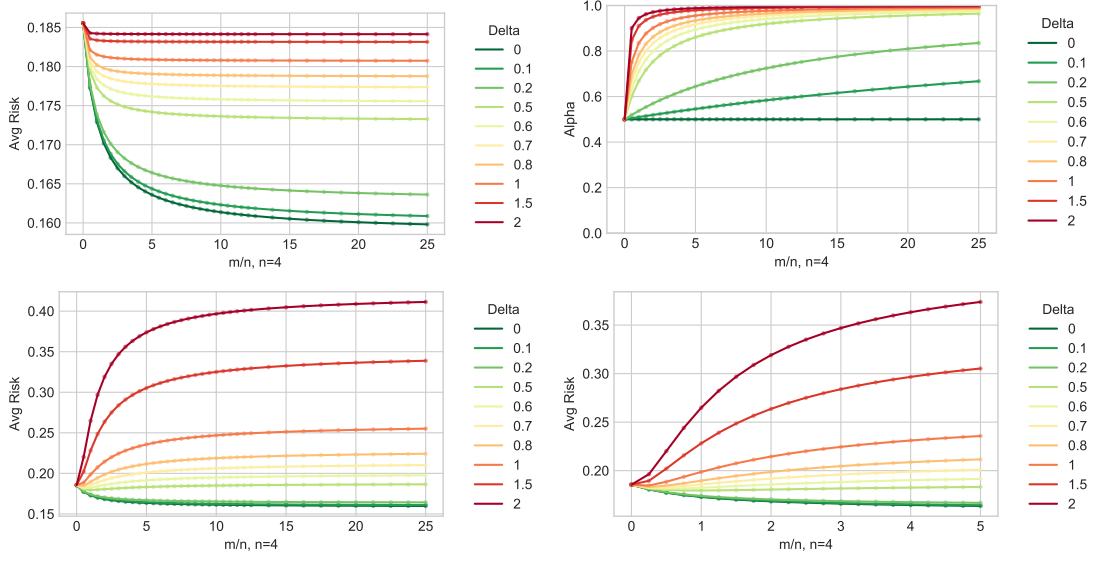


Figure 2: **(top-left)** Risk under optimal alpha and **(top-right)** The value of optimal alpha under the LDA model applied to the target and OOD tasks. Population risk when $\alpha = 0.5$ **(bot-right)** for larger range of m **(bot-right)** and smaller range of m emulating Ashwin’s setup.

Note: The optimal value of α is obtained by minimizing $E[e_t(\bar{h})]$. An analytic expression of α is difficult to compute with both $\bar{\mu}$ and $\bar{\sigma}$ depending on α . Hence we use brute-force search to find the solution. We expect eq. (1) to be a smooth function so heuristic random search algorithms will be effective.

2.3 Multi-head Model: Interpolating the Hypotheses

Consider an algorithm which estimates h_t using n samples using the equation

$$h_t = \frac{\hat{\mathbb{E}}(X|y=1) + \hat{\mathbb{E}}(X|y=-1)}{2} = \frac{1}{n} \sum_{i=1}^n X_i$$

In other words, we estimate the mean of Gaussians corresponding to each class and consider the bisector of the two means to be the hypothesis.

Given m samples from P_t and n samples from P_{ood} , we combine data from both datasets to obtain a hypothesis

$$\bar{h} = \alpha h_t + (1 - \alpha) h_{\text{ood}}$$

where α is a function of n , m and Δ . Unlike the previous section, α is a factor that mixes the two hypotheses, as opposed to the two datasets.

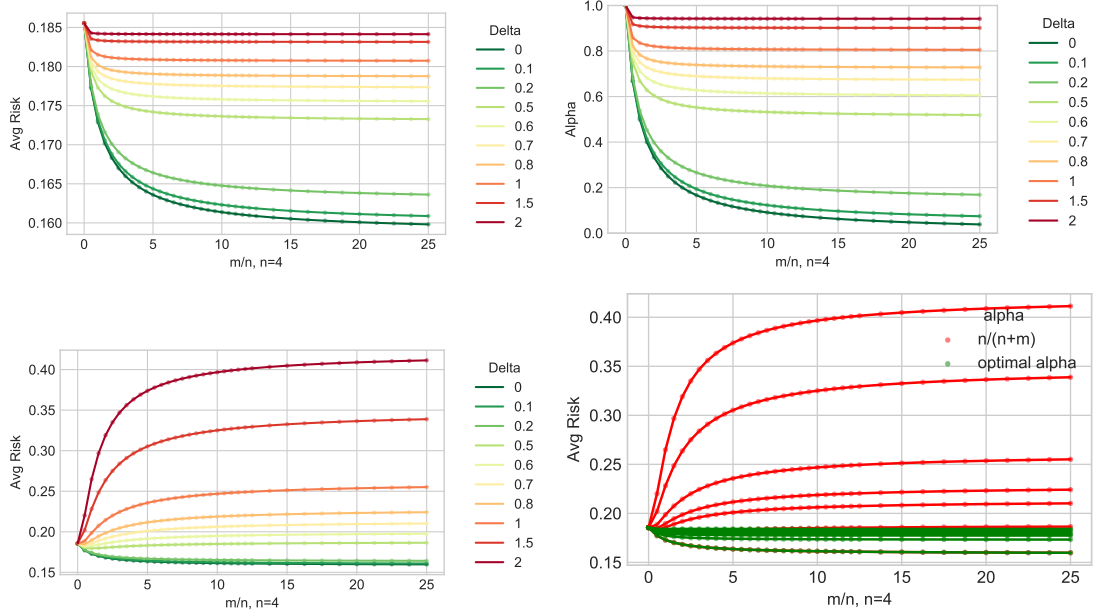


Figure 3: **top-right:** Large the value of Δ , smaller the value of α **top-left:** In all cases, extra data does not hurt if we choose the right value of α . The loss never becomes worse with more samples. (**bot-right**) Loss plotted for $\alpha = n/(n+m)$. **bot-right:** The losses plotted on the same scale when α is tuned/fixed.

Assume that S has $\frac{n}{2}$ samples from each class in both P_t and P_{ood} . h_t is an average of n Gaussian random variables from P_t and is distributed as

$$h_t \sim \mathcal{N}\left(0, \sqrt{\frac{1}{n}}\right).$$

Similarly

$$h_{\text{ood}} \sim \mathcal{N}\left(\Delta, \sqrt{\frac{1}{m}}\right),$$

since h_{ood} is the summation of $\frac{m}{2}$ random variables with law $\mathcal{N}(\mu + \Delta, 1)$ and $\frac{m}{2}$ random variables with law $\mathcal{N}(-\mu + \Delta, 1)$. The resultant hypothesis is also normally distributed over draws of the samples i.e., $\bar{h} \sim \mathcal{N}(\bar{\mu}, \bar{\sigma})$ where

$$\bar{\mu} = (1 - \alpha)\Delta \quad \text{and} \quad \bar{\sigma}^2 = \frac{\alpha^2}{n} + \frac{(1 - \alpha)^2}{m}$$

fig. 3 includes results on this algorithm.