A Student's-t Distribution

Recall that a k-dimensional Student's-t distribution $St(x|\mu, \Sigma, v)$ with $1 < v < +\infty$ degrees of freedom has the following probability density function:

$$St(x|\mu, \Sigma, v) = \frac{\Gamma\left((v+k)/2\right)}{(\pi v)^{k/2}\,\Gamma(v/2)|\Sigma|^{1/2}}\left(1 + (x-\mu)^\top (v\Sigma)^{-1}(x-\mu)\right)^{-(v+k)/2}. \tag{31}$$

Here $\Gamma(\cdot)$ denotes the usual Gamma function. In fact, the Student's-t distribution is a member of the t-exponential family. To see this we first set $-(v+k)/2 = 1/(1-t)$ and

$$\Psi = \left(\frac{\Gamma\left((v+k)/2\right)}{(\pi v)^{k/2}\,\Gamma(v/2)|\Sigma|^{1/2}}\right)^{-2/(v+k)}$$

to rewrite (31) as

$$St(x|\mu, \Sigma, v) = \left(\Psi + \Psi \cdot (x-\mu)^\top (v\Sigma)^{-1}(x-\mu)\right)^{1/(1-t)}. \tag{32}$$

Next we set $\phi(x) = [x; xx^\top]$, $\theta = [\theta_1, \theta_2]$, where $K = (v\Sigma)^{-1}$ and $\theta_1 = -2\Psi K\mu/(1-t)$ while $\theta_2 = \Psi K/(1-t)$. Then we define

$$\langle \phi(x), \theta \rangle = \left(\frac{\Psi}{1-t}\right)\left(x^\top K x - 2\mu^\top K x\right) \text{ and}$$
$$g_t(\theta) = -\left(\frac{\Psi}{1-t}\right)\left(\mu^\top K \mu + 1\right) + \frac{1}{1-t}$$

to rewrite (32) as

$$St(x|\mu, \Sigma, v) = \left(1 + (1-t)\left(\langle \phi(x), \theta \rangle - g_t(\theta)\right)\right)^{1/(1-t)}.$$

Comparing with (11) clearly shows that

$$St(x|\mu, \Sigma, v) = p_t(x;\theta) = \exp_t\left(\langle \phi(x), \theta \rangle - g_t(\theta)\right).$$

Furthermore, using this fact and some simple algebra yields the escort distribution of Student's-t distribution:

$$q_t(x;\theta) = St(x|\mu, v\Sigma/(v+2), v+2)$$

Interestingly, the mean of the Student's-t pdf is μ, and its variance is $v\Sigma/(v-2)$ while the mean and variance of the escort are μ and Σ respectively.

B Properties of $g_t(\theta)$

Although $g_t(\theta)$ is not the cumulant function of the t-exponential family, it still preserves convexity. As the following theorem asserts, its first derivative can still be written as an expectation of $\phi(x)$ but now with respect to the escort distribution. Note that the theorem and proof here is only a special case of a more general one appeared in Sears [13] and [9].

Theorem 1 *The function $g_t(\theta)$ is convex. Moreover, if the following regularity condition*

$$\int \nabla_\theta p(x;\theta)dx = \nabla_\theta \int p(x;\theta)dx \tag{33}$$

holds, then

$$\nabla_\theta g_t(\theta) = \mathbb{E}_{q_t(x;\theta)}\left[\phi(x)\right], \tag{34}$$

where $q_t(x;\theta)$ is the escort distribution (14).

Proof To prove convexity, we rely on the elementary arguments. Recall that $\exp_t$ is an increasing and strictly convex function. Choose θ_1 and θ_2 such that $g_t(\theta_i) < \infty$ for $i = 1, 2$, and let $\alpha \in (0, 1)$. Set $\theta_\alpha = \alpha\theta_1 + (1-\alpha)\theta_2$, and observe that

$$\begin{aligned}&\int \exp_t\left(\langle\phi(x), \theta_\alpha\rangle - \alpha g_t(\theta_1) - (1-\alpha)g_t(\theta_2)\right) dx \\ &< \alpha \int \exp_t(\langle\phi(x), \theta_1\rangle - g_t(\theta_1))dx + (1-\alpha)\int \exp_t(\langle\phi(x), \theta_2\rangle - g_t(\theta_2))dx = 1.\end{aligned}$$

On the other hand, we also have

$$\int \exp_t\left(\langle\phi(x), \theta_\alpha\rangle - g_t(\theta_\alpha)\right) dx = 1.$$

Again, using the fact that $\exp_t$ is an increasing function, we can conclude from the above two equations that

$$g_t(\theta_\alpha) < \alpha g_t(\theta_1) + (1-\alpha)g_t(\theta_2).$$

This shows that g_t is a strictly convex function.

To show (34) use (33) and observe that

$$\int \nabla_\theta p(x;\theta)dx = \nabla_\theta \int p(x;\theta)dx = \nabla_\theta 1 = 0.$$

Combining with the fact that $\frac{d}{dx}\exp_t(x) = \exp_t^t(x)$, use (14) and the chain rule to write

$$\begin{aligned}\int \nabla_\theta p(x;\theta)dx &= \int \nabla_\theta \exp_t\left(\langle\phi(x), \theta\rangle - g_t(\theta)\right) dx \\ &= \int \exp_t^t\left(\langle\phi(x), \theta\rangle - g_t(\theta)\right)\left(\phi(x) - \nabla_\theta g_t(\theta)\right)dx \\ &\propto \int q_t(x;\theta)(\phi(x) - \nabla_\theta g_t(\theta))dx = 0.\end{aligned}$$

Rearranging terms and using $\int q_t(x;\theta)dx = 1$ directly yields (34). ■

C Convergence of Convex Multiplicative Programming

In the Convex Multiplicative Programming, we convert the problem:

$$\operatorname*{argmin}_{\boldsymbol{\theta}} \mathrm{P}(\boldsymbol{\theta}) \triangleq \prod_{n=1}^{N} z_n(\boldsymbol{\theta})$$

into the problem:

$$\operatorname*{argmin}_{\boldsymbol{\theta},\boldsymbol{\xi}} \mathrm{MP}(\boldsymbol{\theta}, \boldsymbol{\xi}) \triangleq \sum_{n=1}^{N} \xi_n z_n(\boldsymbol{\theta}) \quad \text{s.t.} \quad \prod_{n=1}^{N} \xi_n = 1 \text{ and } \boldsymbol{\xi} > 0$$

by introducing the latent variable $\boldsymbol{\xi}$. In the kth $\boldsymbol{\xi}$-step, assuming the current variables are $\boldsymbol{\theta}^{(k-1)}$ and $\boldsymbol{\xi}^{(k-1)}$, we fix $\boldsymbol{\theta}^{(k-1)}$, denote $\tilde{z} = z(\boldsymbol{\theta}^{(k-1)})$, and minimize over $\boldsymbol{\xi}$. It turns out that:

$$\xi_n^{(k)} = \frac{1}{\tilde{z}_n} \prod_{n=1}^{N} \tilde{z}_n^{\frac{1}{N}}$$

Therefore,

$$\mathrm{MP}(\boldsymbol{\theta}^{(k-1)}, \boldsymbol{\xi}^{(k)}) = \min_{\boldsymbol{\xi}} \mathrm{MP}(\boldsymbol{\theta}^{(k-1)}, \boldsymbol{\xi}) = N\mathrm{P}(\boldsymbol{\theta}^{(k-1)})^{1/N} \leq \mathrm{MP}(\boldsymbol{\theta}^{(k-1)}, \boldsymbol{\xi}^{(k-1)})$$

The $\boldsymbol{\theta}$-step is to fix $\boldsymbol{\xi}^{(k)}$ and minimize $\boldsymbol{\theta}$, the result is

$$\mathrm{MP}(\boldsymbol{\theta}^{(k)}, \boldsymbol{\xi}^{(k)}) = \min_{\boldsymbol{\theta}} \mathrm{MP}(\boldsymbol{\theta}, \boldsymbol{\xi}^{(k)}) \leq \mathrm{MP}(\boldsymbol{\theta}^{(k-1)}, \boldsymbol{\xi}^{(k)}) = N\mathrm{P}(\boldsymbol{\theta}^{(k-1)})^{1/N}$$

The above two equalities hold if and only if $\boldsymbol{\xi}^k = \boldsymbol{\xi}^{k-1}$ and $\boldsymbol{\theta}^k = \boldsymbol{\theta}^{k-1}$, which follows the convergence of the algorithm at the kth iteration. Therefore, before convergence we have

$$\mathrm{MP}(\boldsymbol{\theta}^{(k)}, \boldsymbol{\xi}^{(k)}) < N\mathrm{P}(\boldsymbol{\theta}^{(k-1)})^{1/N} < \mathrm{MP}(\boldsymbol{\theta}^{(k-1)}, \boldsymbol{\xi}^{(k-1)}).$$

But since $\mathrm{P}(\theta) > 0$, the algorithm must converge at some point.

Next, we want to show that $\tilde{\boldsymbol{\theta}}$ after convergence is a stable point of the $\mathrm{P}(\boldsymbol{\theta})$. Assume that $\tilde{\boldsymbol{\theta}}$ and $\tilde{\boldsymbol{\xi}}$ is the convergence point, then the $\boldsymbol{\theta}$-step is:

$$0 = \sum_{n=1}^{N} \prod_{n=1}^{N} z_n(\tilde{\boldsymbol{\theta}})^{\frac{1}{N}} \frac{1}{z_n(\tilde{\boldsymbol{\theta}})} \frac{dz_n(\boldsymbol{\theta})|_{\boldsymbol{\theta}=\tilde{\boldsymbol{\theta}}}}{d\,\boldsymbol{\theta}} = \left(\prod_{n=1}^{N} z_n(\tilde{\boldsymbol{\theta}})^{\frac{1}{N}}\right) \left(\sum_{n=1}^{N} \frac{1}{z_n(\tilde{\boldsymbol{\theta}})} \frac{dz_n(\boldsymbol{\theta})|_{\boldsymbol{\theta}=\tilde{\boldsymbol{\theta}}}}{d\,\boldsymbol{\theta}}\right)$$

which implies that,

$$0 = \sum_{n=1}^{N} \frac{1}{z_n(\boldsymbol{\theta})|_{\boldsymbol{\theta}=\tilde{\boldsymbol{\theta}}}} \frac{dz_n(\boldsymbol{\theta})|_{\boldsymbol{\theta}=\tilde{\boldsymbol{\theta}}}}{d\,\boldsymbol{\theta}} = \sum_{n=1}^{N} \frac{\prod_{n=1}^{N} z_n(\tilde{\boldsymbol{\theta}})}{z_n(\tilde{\boldsymbol{\theta}})} \frac{dz_n(\boldsymbol{\theta})|_{\boldsymbol{\theta}=\tilde{\boldsymbol{\theta}}}}{d\,\boldsymbol{\theta}} = \frac{d\mathrm{P}(\boldsymbol{\theta})|_{\boldsymbol{\theta}=\tilde{\boldsymbol{\theta}}}}{d\,\boldsymbol{\theta}}$$

Therefore, $\tilde{\boldsymbol{\theta}}$ is a stable point of $\mathrm{P}(\boldsymbol{\theta})$.

D Gradient Based method

It is also possible to directly use the gradient based method such as L-BFGS to solve (23). To do this, it is convenient to take log of (23).

$$\log \mathrm{P}(\boldsymbol{\theta}) = \sum_{n=1}^{N} \log z_n(\boldsymbol{\theta}) = \sum_{j=1}^{d} \log r_j(\boldsymbol{\theta}) + \sum_{i=1}^{m} \log l_i(\boldsymbol{\theta}) \tag{35}$$

$$= \sum_{j=1}^{d} \log\left(1 + (1-t)(-\tilde{\lambda}\theta_j^2/2 - \tilde{g}_t)\right) + \sum_{i=1}^{m} \log\left(1 + (1-t)(\left\langle \frac{y_i}{2}\phi(\mathbf{x}_i), \boldsymbol{\theta}\right\rangle - g_t(\boldsymbol{\theta} \,|\, \mathbf{x}_i))\right) \tag{36}$$

Take the derivative,

$$\text{for } n = 1, \ldots, d \quad \nabla_{\boldsymbol{\theta}} \log z_n(\boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}} \log r_n(\boldsymbol{\theta}) = \left(\frac{t-1}{r_n(\boldsymbol{\theta})}\right) \cdot \tilde{\lambda}\theta_n \, \mathbf{e}_n \tag{37}$$

$$\text{for } n = 1, \ldots, m \quad \nabla_{\boldsymbol{\theta}} \log z_{n+d}(\boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}} \log l_n(\boldsymbol{\theta})$$
$$= \left(\frac{1-t}{l_n(\boldsymbol{\theta})}\right) \cdot \left(\frac{y_n}{2}\phi(\mathbf{x}_n) - \mathbb{E}_{q_t(y_n|\,\mathbf{x}_n;\boldsymbol{\theta})}\left[\frac{y_n}{2}\phi(\mathbf{x}_n)\right]\right) \tag{38}$$

where $\mathbf{e}_n$ denotes the d dimensional vector with one at the n-th coordinate and zeros elsewhere (n-th unit vector). There is an obvious relation of (37) (38), and the previous routine, given that $\xi_n = 1/z_n(\boldsymbol{\theta})$ here and $\tilde{\xi}_n \propto 1/\tilde{z}_n$ in (29).

We report the performance of the t-logistic regression by directly using L-BFGS as the optimizer in Table 5. The algorithms use the same parameters as in Table 1. Since L-BFGS is not designed to optimize non-convex functions, it may fail sometimes, in which case we randomly restart with a different initialization.

E Higher Label Noise

Our algorithm appears to be more robust than logistic regression (especially when $t = 1.9$) against the label noise (10%). A natural question to ask then is how well it performs when the label noise is larger than 10%. In Figure 6, we compare the t-logistic regression with logistic regression and the probit in the cases when 20% and 30% label noise is added. We also report the test error in Table 6.

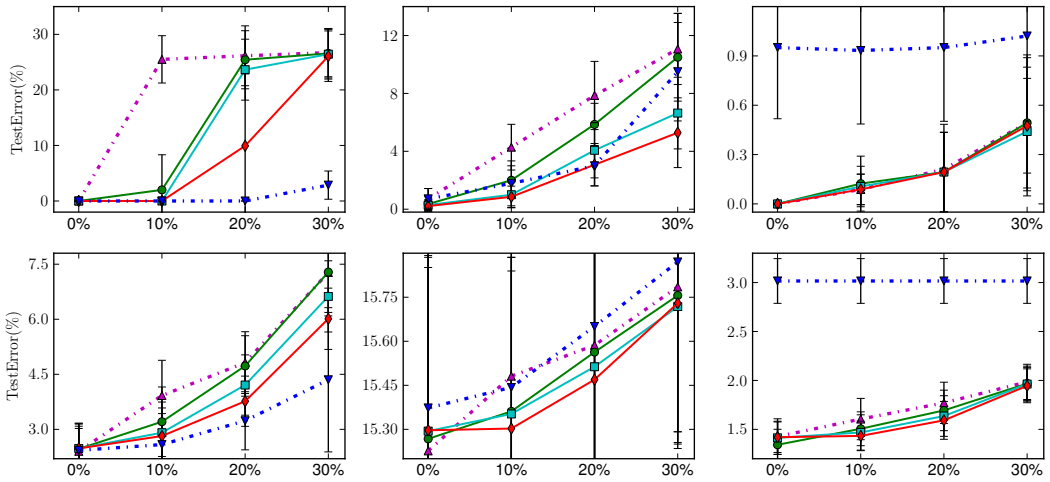

Figure 6: The test performance with the change of the label noise (left to right, top: Long-Servedio, Mease-Wyner, Mushroom; bottom: USPS-N, Adult, Web). The magenta dash line with upper-triangles is the logistic regression; the green line with circles is $t = 1.3$; the cyan line with squares is $t = 1.6$; the red line with diamonds is $t = 1.9$; and the blue dash line with lower-triangles is the probit.

F Significance Test

We performed the paired T-test of the test error rates for each dataset obtained by the t-logistic regression with $t = 1.9$ and by the other algorithms. To do this, we take the difference of the error rate for each split in each dataset by any two algorithms. The hypothesis is that the difference of the two algorithms for each split is drawn from a zero-mean normal distribution with unknown variance in the same dataset. We report the significance test results in Table 2.

G Selected Results from RampSVM

Unfortunately, we do not obtain good results by using RampSVM [20] on our datasets. We used the UniverSVM package [23], and performed a grid search of the parameters C and s. It appears that the optimal parameter C is consistent with that used by the Linear SVM, since the UniverSVM uses the solution of Linear SVM to initialize. We report the test performance for $s = 0, -0.1, -1, -10, -100$ in Table 7. Usually the results are significantly worse than the other algorithms, except for the Long-Servedio dataset where RampSVM performs as well as other non-convex losses with label noise. We therefore do not report results the RampSVM in the main body of the paper.

Table 1: Datasets used in our experiments.(λ_l denotes λ for the logistic regression. $\lambda_{1.3}$, $\lambda_{1.6}$, $\lambda_{1.9}$ are the λ values used for t-logistic regression with $t = 1.3$, 1.6, and 1.9 respectively. λ_p is λ for the probit algorithm. C is the parameter C for the C-SVC which is equivalent to $1/\lambda$.)

Name	Noise	Dimensions	Num. of examples	λ_l	$\lambda_{1.3}$	$\lambda_{1.6}$	$\lambda_{1.9}$	λ_p	C
Long-Servedio	0%	21	2000	2^{-7}	2^{-7}	2^{-7}	2^{-7}	2^{-7}	2^{-5}
Mease-Wyner	0%	20	2000	2^{-7}	2^{6}	2^{4}	2^{3}	2^{-7}	2^{7}
Mushroom	0%	112	8124	2^{-7}	2^{-7}	2^{-2}	2^{-5}	2^{-2}	2^{-2}
USPS-N	0%	256	11000	2^{4}	2^{2}	2^{1}	2^{0}	2^{4}	2^{-5}
Adult	0%	123	48842	2^{5}	2^{5}	2^{4}	2^{3}	2^{2}	2^{-3}
Web	0%	300	64700	2^{-5}	2^{-7}	2^{0}	2^{-1}	2^{-7}	2^{7}
Long-Servedio	10%	21	2000	2^{-7}	2^{-7}	2^{-1}	2^{0}	2^{-2}	2^{3}
Mease-Wyner	10%	20	2000	2^{-2}	2^{7}	2^{7}	2^{6}	2^{-7}	2^{3}
Mushroom	10%	112	8124	2^{1}	2^{3}	2^{3}	2^{3}	2^{1}	2^{4}
USPS-N	10%	256	11000	2^{7}	2^{7}	2^{7}	2^{6}	2^{5}	2^{-3}
Adult	10%	123	48842	2^{4}	2^{5}	2^{4}	2^{3}	2^{3}	2^{-1}
Web	10%	300	64700	2^{-7}	2^{-7}	2^{-7}	2^{-7}	2^{-7}	2^{1}

Table 2: Significance Test of the test error rates by the t-logistic regression with $t = 1.9$ and the other algorithms. The significance factor α is set as 0.05. 'Y' means that the difference is significant. 'N' means the difference is not significant.

Dataset	Noise	Logistic	t=1.3	t=1.6	Probit	SVM
Long-Servedio	0%	N	N	N	N	N
Mease-Wyner	0%	**Y**	N	N	**Y**	**Y**
Mushroom	0%	N	N	N	**Y**	N
USPS-N	0%	N	N	N	N	N
Adult	0%	N	N	N	N	N
Web	0%	N	**Y**	N	**Y**	N
Long-Servedio	10%	**Y**	N	N	N	**Y**
Mease-Wyner	10%	**Y**	**Y**	N	N	**Y**
Mushroom	10%	N	N	N	**Y**	N
USPS-N	10%	**Y**	**Y**	N	N	**Y**
Adult	10%	**Y**	N	N	N	**Y**
Web	10%	**Y**	**Y**	N	**Y**	**Y**

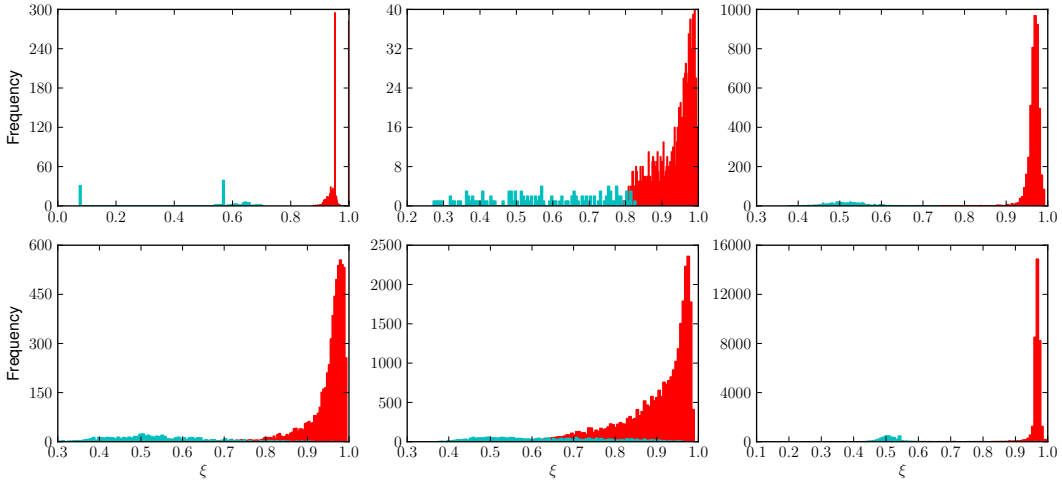

Figure 7: The ξ distribution with 10% label noise added. $t = 1.3$. Left to right, top: Long-Servedio, Mease-Wyner, Mushroom; bottom: USPS-N, Adult, Web. The red bars (resp. cyan bars) indicate the ξ assigned to points without (resp. with) label noise.

Dataset	Noise	Logistic	t=1.3	t=1.6	t=1.9	Probit	SVM
Long-Servedio	0%	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$
Mease-Wyner	0%	0.71 ± 0.71	0.36 ± 0.51	0.29 ± 0.37	$\mathbf{0.21 \pm 0.35}$	0.71 ± 0.71	1.90 ± 1.09
Mushroom	0%	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	0.95 ± 0.43	$\mathbf{0.00 \pm 0.00}$
USPS-N	0%	2.40 ± 0.62	2.47 ± 0.58	2.48 ± 0.67	2.48 ± 0.67	2.43 ± 0.74	$\mathbf{2.26 \pm 0.64}$
Adult	0%	$\mathbf{15.23 \pm 0.62}$	15.27 ± 0.62	15.30 ± 0.59	15.31 ± 0.62	15.37 ± 0.62	15.38 ± 0.64
Web	0%	1.43 ± 0.18	$\mathbf{1.34 \pm 0.16}$	1.41 ± 0.17	1.42 ± 0.16	3.02 ± 0.23	1.38 ± 0.17
Long-Servedio	10%	25.50 ± 4.26	2.00 ± 6.32	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	$\mathbf{0.00 \pm 0.00}$	23.36 ± 8.90
Mease-Wyner	10%	4.29 ± 1.58	2.00 ± 1.00	1.00 ± 0.90	$\mathbf{0.86 \pm 0.74}$	1.79 ± 1.55	4.71 ± 1.18
Mushroom	10%	$\mathbf{0.09 \pm 0.09}$	0.12 ± 0.17	0.11 ± 0.12	$\mathbf{0.09 \pm 0.09}$	0.93 ± 0.45	0.11 ± 0.09
USPS-N	10%	3.92 ± 0.96	3.21 ± 0.95	2.91 ± 0.84	2.82 ± 0.76	$\mathbf{2.58 \pm 0.84}$	4.10 ± 1.02
Adult	10%	15.48 ± 0.53	15.36 ± 0.52	15.35 ± 0.49	$\mathbf{15.30 \pm 0.58}$	15.44 ± 0.56	16.19 ± 0.48
Web	10%	1.61 ± 0.21	1.51 ± 0.18	1.47 ± 0.17	$\mathbf{1.43 \pm 0.15}$	3.02 ± 0.23	1.77 ± 0.12

Table 3: Test Error Rate in % for various algorithms. The noisy version is obtained by flipping the labels of randomly chosen fixed fraction of the training data.

Dataset	Noise	Logistic	t=1.3	t=1.6	t=1.9	Probit	SVM
Long-Servedio	0%	0.02 ± 0.00	0.17 ± 0.12	0.15 ± 0.03	0.15 ± 0.03	0.07 ± 0.04	0.59 ± 0.17
Mease-Wyner	0%	0.02 ± 0.01	1.12 ± 0.30	1.70 ± 0.46	0.98 ± 0.20	0.06 ± 0.06	0.27 ± 0.14
Mushroom	0%	0.33 ± 0.03	0.77 ± 0.10	1.89 ± 0.28	0.94 ± 0.18	0.25 ± 0.08	0.40 ± 0.04
USPS-N	0%	0.87 ± 0.11	7.09 ± 2.20	12.65 ± 3.05	17.81 ± 3.01	0.75 ± 0.07	8.25 ± 0.66
Adult	0%	1.03 ± 0.06	11.11 ± 4.70	27.73 ± 13.10	36.29 ± 8.21	0.96 ± 0.10	111.33 ± 7.92
Web	0%	4.92 ± 0.36	41.52 ± 9.01	31.40 ± 11.44	40.27 ± 10.26	1.86 ± 0.28	488.30 ± 127.42
Long-Servedio	10%	0.01 ± 0.00	0.33 ± 0.09	0.27 ± 0.02	0.33 ± 0.01	0.04 ± 0.01	6.20 ± 4.12
Mease-Wyner	10%	0.01 ± 0.00	0.34 ± 0.08	0.42 ± 0.05	0.59 ± 0.08	0.04 ± 0.03	0.32 ± 0.07
Mushroom	10%	0.33 ± 0.02	2.47 ± 0.54	2.57 ± 0.40	3.46 ± 1.07	0.19 ± 0.03	8.38 ± 1.63
USPS-N	10%	0.78 ± 0.12	3.13 ± 0.82	4.77 ± 0.77	6.36 ± 1.50	0.66 ± 0.08	61.92 ± 2.95
Adult	10%	1.74 ± 0.18	7.83 ± 1.76	13.49 ± 2.92	16.99 ± 5.24	1.05 ± 0.17	265.70 ± 24.55
Web	10%	8.45 ± 0.60	43.79 ± 9.39	51.02 ± 6.96	48.21 ± 6.14	0.46 ± 0.07	3312.38 ± 5455.18

Table 4: CPU Time (in seconds)

Dataset	Noise	Test Error (%)			Time (s)		
		t=1.3	t=1.6	t=1.9	t=1.3	t=1.6	t=1.9
Long-Servedio	0%	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.09 ± 0.05	0.42 ± 0.72	0.09 ± 0.03
Mease-Wyner	0%	0.36 ± 0.51	0.29 ± 0.50	0.43 ± 0.69	0.07 ± 0.02	0.08 ± 0.02	0.10 ± 0.11
Mushroom	0%	0.00 ± 0.00	0.00 ± 0.00	0.02 ± 0.06	0.75 ± 0.07	0.97 ± 0.10	0.98 ± 0.76
USPS-N	0%	2.45 ± 0.59	2.48 ± 0.62	2.55 ± 0.65	3.48 ± 0.46	5.69 ± 0.81	6.27 ± 2.12
Adult	0%	15.27 ± 0.62	15.30 ± 0.60	15.32 ± 0.60	5.13 ± 0.69	7.96 ± 0.99	9.21 ± 0.97
Web	0%	1.34 ± 0.15	1.40 ± 0.16	1.44 ± 0.18	11.79 ± 1.52	8.30 ± 3.59	8.48 ± 7.76
Long-Servedio	10%	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.06 ± 0.04	0.04 ± 0.01	0.05 ± 0.01
Mease-Wyner	10%	2.00 ± 1.00	1.00 ± 0.90	0.86 ± 0.74	0.06 ± 0.03	0.07 ± 0.01	0.08 ± 0.01
Mushroom	10%	0.12 ± 0.17	0.11 ± 0.12	0.09 ± 0.09	0.72 ± 0.09	0.63 ± 0.06	0.71 ± 0.10
USPS-N	10%	3.22 ± 0.93	2.92 ± 0.85	2.87 ± 0.78	1.53 ± 0.21	1.72 ± 0.19	2.11 ± 0.33
Adult	10%	15.36 ± 0.53	15.34 ± 0.49	15.31 ± 0.58	4.23 ± 0.18	5.35 ± 0.65	6.54 ± 0.64
Web	10%	1.50 ± 0.18	1.47 ± 0.17	1.43 ± 0.16	27.04 ± 2.00	23.18 ± 1.77	13.37 ± 6.15

Table 5: Results of t-logistic regression by using L-BFGS directly.

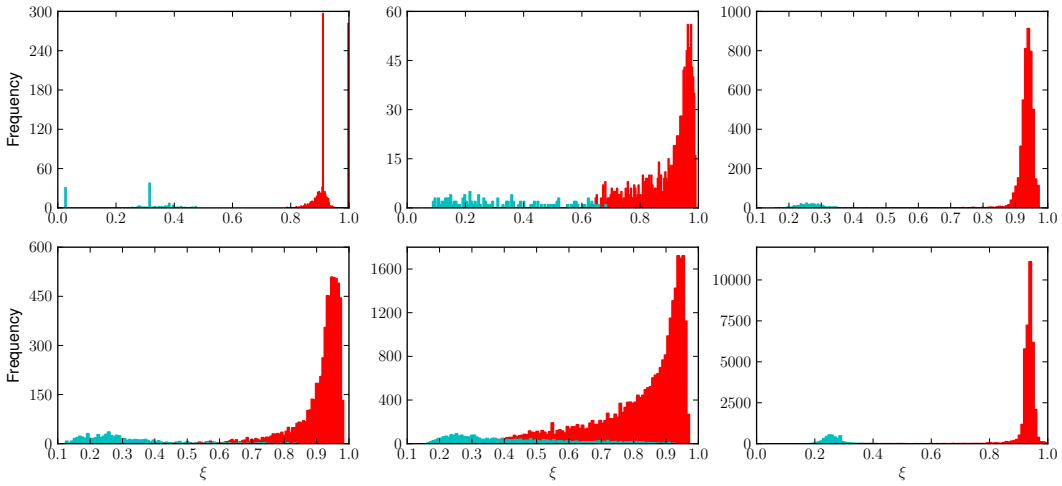

Figure 8: The ξ distribution with 10% label noise added. $t = 1.6$. Left to right, top: Long-Servedio, Mease-Wyner, Mushroom; bottom: USPS-N, Adult, Web. The red bars (resp. cyan bars) indicate the ξ assigned to points without (resp. with) label noise.

Dataset	Noise	Logistic	t=1.3	t=1.6	t=1.9	Probit
Long-Servedio	20%	26.14 ± 5.39	25.43 ± 5.23	23.64 ± 5.49	9.93 ± 12.97	0.00 ± 0.00
Mease-Wyner	20%	7.86 ± 2.36	5.86 ± 1.46	4.07 ± 1.91	3.07 ± 1.47	3.00 ± 1.38
Mushroom	20%	0.21 ± 0.27	0.19 ± 0.24	0.19 ± 0.24	0.19 ± 0.24	0.95 ± 0.45
USPS-N	20%	4.82 ± 0.84	4.73 ± 0.82	4.21 ± 0.82	3.77 ± 0.68	3.23 ± 0.79
Adult	20%	15.59 ± 0.49	15.56 ± 0.50	15.51 ± 0.50	15.47 ± 0.45	15.65 ± 0.62
Web	20%	1.77 ± 0.21	1.69 ± 0.21	1.64 ± 0.21	1.59 ± 0.19	3.02 ± 0.23
Long-Servedio	30%	26.71 ± 4.34	26.57 ± 4.43	26.43 ± 4.48	26.07 ± 4.55	2.86 ± 2.54
Mease-Wyner	30%	11.07 ± 2.46	10.50 ± 3.03	6.64 ± 2.48	5.29 ± 2.41	9.50 ± 3.40
Mushroom	30%	0.49 ± 0.40	0.49 ± 0.41	0.44 ± 0.39	0.48 ± 0.29	1.02 ± 0.52
USPS-N	30%	7.27 ± 1.09	7.29 ± 1.19	6.62 ± 0.97	6.01 ± 0.84	4.35 ± 1.97
Adult	30%	15.79 ± 0.49	15.76 ± 0.46	15.72 ± 0.48	15.73 ± 0.47	15.87 ± 0.62
Web	30%	1.98 ± 0.18	1.97 ± 0.18	1.96 ± 0.17	1.95 ± 0.17	3.02 ± 0.23

Table 6: Test Error Rate in % for various algorithms with higher label noise.

Dataset	Noise	C	s=0	s=-0.1	s=-1	s=-10	s=-100
Long-Servedio	0%	2^{-5}	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
Mease-Wyner	0%	2^{7}	50.5 ± 4.10	50.5 ± 4.10	50.5 ± 4.10	50.5 ± 4.10	50.5 ± 4.10
Mushroom	0%	2^{-2}	0.04 ± 0.08	0.04 ± 0.08	0.04 ± 0.08	0.04 ± 0.08	0.04 ± 0.08
Long-Servedio	10%	2^{3}	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	23.29 ± 8.84	23.29 ± 8.84
Mease-Wyner	10%	2^{3}	50.5 ± 4.10	50.5 ± 4.10	50.5 ± 4.10	50.5 ± 4.10	50.5 ± 4.10
Mushroom	10%	2^{4}	5.25 ± 8.87	5.74 ± 8.68	9.08 ± 10.47	47.19 ± 1.66	47.19 ± 1.66

Table 7: The test error (%) of RampSVM on selected datasets.