FROM NEYMAN-PEARSON LEMMA TO NEYMAN-PEARSON CLASSIFIER

HENGRUI LUO

Generalized Neyman-Pearson Lemma[1]

The Neyman-Pearson Lemma is a well-known result in statistical hypothesis testing theories. Its generalized version could be stated as below. This piece of review illustrates how this stream of thought flows and investigate the performance of Neyman-Pearson Classifier(NPC)[1] based on kernel density estimation with simulation evidence.

Theorem 1. *(Theorem 3.6.1 in* [7]*) Let $f_1, ..., f_{m+1}$ be real-valued functions defined on a Euclidean space X and integrable μ, and suppose that for given constants $c_1, \cdots c_m$ there exists a critical function ϕ satisfying $\int_X \phi(x) f_i(x) d\mu = c_i, \forall i = 1, \cdots m$, let $\mathcal{C}$ be the class of critical functions ϕ for which the condition holds.*

(i) Among all members of $\mathcal{C}$ there exists one that maximizes $\int_X \phi(x) f_{m+1}(x) d\mu$

(ii) A sufficient condition for a member of $\mathcal{C}$ to maximize $\int_X \phi(x) f_{m+1}(x) d\mu$ is the existence of constants $k_1, ..., k_m$ such that

$$\phi(X) = \begin{cases} 1 & f_{m+1}(x) > \sum_{i=1}^m k_i f_i(x) \\ 0 & f_{m+1}(x) < \sum_{i=1}^m k_i f_i(x) \end{cases}$$

If a member of $\mathcal{C}$ satisfies this form with $k_1, \cdots, k_m \geq 0 \geq 0$, then it maximizes $\int_X \phi(x) f_{m+1}(x) d\mu$ among all critical functions satisfying $\int_X \phi(x) f_i(x) d\mu = c_i, \forall i = 1, \cdots m$.

(iii) The set M of points in m-dimensional space whose coordinates are

$$\left(\int_X \phi(x) f_1(x) d\mu, \int_X \phi(x) f_2(x) d\mu, \cdots \int_X \phi(x) f_m(x) d\mu \right)$$

for some critical function ϕ. M is convex and closed.

If $(c_1, \cdots, c_m)$ is an inner point of M, then there exist constants $k_1, \cdots, k_m$ and a test ϕ satisfying the form above. And a necessary condition for ϕ to maximize $\int_X \phi f_{m+1} d\mu$ is that it is of the form as above a.e. μ.

This result generalized the Neyman-Pearson Lemma in sense that it can provide the uniformly most powerful(UMP) test of a prescribed significance level α for a hypothesis testing with a composite null hypothesis against simple alternative while the Neyman-Pearson Lemma itself only asserts the form of UMP test for simple null hypothesis against simple alternative hypothesis.

A second look into the proof of the Neyman-Pearson Lemma indicates that this result is actually a statement that the extreme points on a convex closed set M must lie on the boundary of the convex set M. For the hypothesis testing problem $H : f_0, f_1, \cdots, f_m$ v.s. $K : f_{m+1}$ where f_i's are densities with respect to the dominating

[1]Part of the formalism comes from [6]

measure μ, each critical function ϕ is represented by an m-dim coordinate risk point $\left(\int_X \phi(x)f_1(x)d\mu, \int_X \phi(x)f_2(x)d\mu, \cdots \int_X \phi(x)f_m(x)d\mu\right) \in M$. To control the type I error of the test ϕ, we must put constraints on each of m coordinates. The generalized Neyman-Pearson Lemma tells us that this set M is convex and closed. In order to maximize the power function $\beta_\phi(f_{m+1}) = \int_X \phi(x)f_{m+1}(x)d\mu$ under alternative hypothesis is equivalent to maximize it over the closed convex set M and hence the optimizer ϕ must have coordinates corresponding to $\partial M \subset M$, the boundary of the convex set [2][9].

Neyman-Pearson Paradigm

Meanwhile, the testing problem can also be formulated as a decision-theoretic problem by choosing the generalized 0-1 loss function [8]. The problem of finding a UMP test ϕ is equivalent to finding a minimax estimator under the constricted risk set $(R_1, \cdots R_m) \in M$. Once we formulated the testing problem as a decision-theoretic problem, the same decision rules can be applied to other problems with the same decision-theoretic formulation.

In a general binary classification the goal is to learn a classifier(decision rule) ϕ from training data labeled 0 and 1 and make sure that ϕ performs the best under certain criteria(loss function). We can construct such a criteria that the binary classification problem has the same decision-theoretic structure as we described above. But as pointed out by [2], "most existing binary classification methods target on the optimization of the overall risk

$$R(\phi) = P(Y = 0)R_0(\phi) + P(Y = 1)R_1(\phi)$$

and may fail to serve the purpose when users' relative priorities over type I/II errors differ significantly from those implied by the marginal probabilities of the two classes". Therefore it is necessary to provide a new paradigm that incorporate such a concern.

Neyman-Pearson paradigm allows us to control these two types of errors by requiring an *oracle* such that $\phi^* \in argmin_{R_0(\phi)\leq\alpha} R_1(\phi)$. That is the (decision-theoretic) motivation of the Neyman-Pearson paradigm (NP-paradigm). In many classification problems such as genome-wide analysis, it is more natural to specify type I error [2, 4]. Further extensions of this idea of NP-paradigm onto more complex classification problems can be found in [4, 6].

If we want to choose an oracle binary classifier ϕ^* in a testing problem, we need to control the R_0, R_1 simultaneously if we are in the simple v.s. simple case by Neyman-Pearson Lemma; and we need to control $R_0, R_1, \cdots R_m$ and R_{m+1} simultaneously if we are in the composite v.s. simple case by generalized Neyman-Pearson Lemma above. But we also know that if the null parameter space Ω_H and alternative hypothesis space Ω_K become more complicated, then [5] proved that such oracle binary classifier with probability one only exists for underlying distributions that are one-parameter exponential families. Therefore we need a set of more flexible criteria in order to find a "near-oracle" classifier. Therefore we turn to NP oracle inequalities that characterize the classifiers' theoretical performances

[2]By minimax theory [6], such an optimal rule as a minimax point must corresponding to a supporting hyperplane at some point of the risk set$\partial\mathcal{C}$. For finite dimensional parameter space AND bounded risk set $\mathcal{C}$, the minimax point is unique iff $\partial\mathcal{C}$ is differentiable ([9] Chap 25). For infinite dimensional parameter space OR unbounded risk set $\mathcal{C}$, the conclusion is not known in general yet. This is pointed out to me by Prof.Deane Yang.

under the NP paradigm [2]. The oracle inequalities highlight a class of classifiers $\hat{\phi}$ such that

(1) The type I error constraint is respected, i.e. $R_0(\hat{\phi}) \leq \alpha$
(2) The excess type II error $R_1(\hat{\phi}) - R_1(\phi^*)$ diminishes with explicit rates (w.r.t. sample size n) where the oracle $\phi^* \in argmin_{R_0(\phi)\leq\alpha} R_1(\phi)$. Or we say the binary classifier converges to the oracle in terms of R_1.

Intuitively, these oracle inequalities ask for a classifier $\hat{\phi}$ that is "close to" the oracle classifier with high probability. This makes the restrictive assertion in [5] no longer holds and allows the $\hat{\phi}$ to be applied on many situations with various families of underlying distributions with little loss of power. If we view this as a testing problem, then $\hat{\phi}$ will converge to UMP test ϕ^* as the sample size $n \to \infty$ while preserving significance level α.

Neyman-Pearson Classifier

Nonparametric Neyman-Pearson Classifier. The article [2] constructed such a classfier $\hat{\phi}$ that meets the NP oracle inequalities. Theoretically, the method they proposed is to construct a Neyman-Pearson classifier using the kernel estimation techniques [2]. Practically, their method performs better than naive Bayes classification in terms of ROC curves in high dimensional settings as shown in simulation and real data analysis [1]. Their procedure can be divided into two steps, the first step is to compute the likelihood ratio statistics using kernel density estimation methods; the second step is to compute the threshold value of Neyman-Pearson rejection test. To achieve these two steps, we need to perform a random split of our training sample with correct labels. Suppose that the training sample contains n i.i.d. observations $S^1 = \{U_1, \cdots U_n\}$ from class 1 with density p, and m i.i.d. observations $S^0 = \{V_1, \cdots V_m\}$ from class 0 with density q. We split these two training sets into 2 and 3 partitions respectively. $S^1 = \begin{cases} S_1^1 = \{U_1, U_2 \cdots U_{n_1}\} & |S_1^1| = n_1 \\ S_2^1 = \{U_{n_1+1}, U_{n_1+2} \cdots U_{n_2}\} & |S_2^1| = n_2 \end{cases}$ and $S^0 = \begin{cases} S_1^0 = \{V_1, V_2 \cdots V_{m_1}\} & |S_1^0| = m_1 \\ S_2^0 = \{V_{m_1+1}, V_{m_1+2} \cdots V_{m_2}\} & |S_2^0| = m_2 \\ S_3^0 = \{V_{m_2+1}, V_{m_2+2} \cdots V_{m_3}\} & |S_3^0| = m_3 \end{cases}$. The S_1^0, S_2^0 and S_1^1, S_2^1 are used to construct kernel density estimators and a density ratio estimator $\hat{r}(x)$.

$$\hat{r}(\boldsymbol{x}) = \hat{r}((x_1, \cdots x_d)) = \prod_{j=1}^{d} \frac{\hat{p_j}(x_j)}{\hat{q_j}(x_j)} = \prod_{j=1}^{d} \frac{\frac{1}{(n_1+n_2)h_1} \sum_{i=1}^{n_1+n_2} K(\frac{U_{i,j}-x_j}{h_1})}{\frac{1}{(m_1+m_2)h_0} \sum_{i=1}^{m_1+m_2} K(\frac{V_{i,j}-x_j}{h_0})}$$

where $U_{i,j}, V_{i,j}$ are the j-th component of the i-th sample; $K(\bullet)$ is the associated kernel and h_1, h_0 are the bandwidth chosen by KDE procedure. Now by Neyman-Pearson Lemma, we can proceed the second step and construct a threshold $\hat{C_\alpha} = \hat{r}_{(k)}(S_3^0)$ and correspondingly a critical function $\hat{\phi_k}(x) = \mathbf{1}_{\{\hat{r}(x) \geq \hat{r}_{(k)}(S_3^0)\}}$ where $\hat{r}_{(k)}(S_3^0)$ is the the k-th order statistics of $\hat{r}(S_3^0) = \left\{\hat{r}(x) : x \in S_3^0\right\}$. For a significant level α we have to choose a threshold $\hat{C_\alpha}$, and thus to attain certain power, we need sufficient sample sizes, and that is why the phenomenon shown in Figure 2 will occur when the training sample contains too few sample labeled 0. Intuitively, the requirement on the sample size of training sample is equivalent to

TABLE 1. Variants of NP procedures. (Table 2 in [2])

	Screening-based	Non-screening
Non-parametric	$\hat{\phi}_{\mathrm{NSN}^2}(x) = \mathbb{1}\left\{\hat{r}_{\mathrm{N}}^S(x) \geq (\hat{r}_{\mathrm{N}}^S)_{(k_{\min})}\right\}$	$\hat{\phi}_{\mathrm{NN}^2}(x) = \mathbb{1}\left\{\hat{r}_{\mathrm{N}}(x) \geq (\hat{r}_{\mathrm{N}})_{(k_{\min})}\right\}$
Parametric	$\hat{\phi}_{\mathrm{PSN}^2}(x) = \mathbb{1}\left\{\hat{r}_{\mathrm{P}}^S(x) \geq (\hat{r}_{\mathrm{P}}^S)_{(k_{\min})}\right\}$	$\hat{\phi}_{\mathrm{PN}^2}(x) = \mathbb{1}\left\{\hat{r}_{\mathrm{P}}(x) \geq (\hat{r}_{\mathrm{P}})_{(k_{\min})}\right\}$

requiring a certain sample size for attaining higher power for a given UMP level α test. The basic thinking of this procedure is neat and straight-forward.

Neyman-Pearson Classifier with screening. However, due to the insightful criticism mentioned in [3] for poor performance in high dimension, NPC cannot be directly applied to high dimensional data without chopping down some noise features. One further step [2] took to address the problem of noise accumulation in high dimension setting mentioned by [3][3] is to impose a screening procedure before density estimation. The screening procedure is a technique proposed by [3] and consequential papers by the same authors to reduce the irrelevant features of a high dimensional dataset. To screen features using S_1^0, S_1^1, we use the $\hat{A}_\tau = \left\{1 \leq j \leq d : \left\|\widehat{F_j^0} - \widehat{F_j^1}\right\|_\infty \geq \tau\right\}$ to choose a subset of d features and $\widehat{F_j^0}, \widehat{F_j^1}$ are the empirical cumulative distribution functions for the j-th feature, which is $\widehat{F_j^0}(x_j) = \frac{1}{m_1}\sum_{i=1}^{m_1} \mathbf{1}_{\{V_{i,j} \leq x_j\}}, \widehat{F_j^1}(x_j) = \frac{1}{n_1}\sum_{i=1}^{n_1} \mathbf{1}_{\{U_{i,j} \leq x_j\}}$. The resulting $\hat{A}_\tau$ is a subset of $\{1, 2, \cdots d\}$ represent the index set of the selected features with screening value τ. We only perform NPC as if the dataset has only features corresponding to the index set $\hat{A}_\tau$.

With this kind of modification, the NPC can be proven to perform and fits the NP-paradigm under mild assumptions and performs better than the naive Bayes due to the additional screening [1, 2]. [4][5] And it also out-performs the naive Bayes classifier in high dimensional settings and therefore favorable.

Based on the primitive Neyman-Pearson procedure described above, [2] proposed a few variants of the NPC. These variants are different in sense that they use different density ratio estimators $\hat{r}$ in NP procedure. The threshold k_{min} is a function of the type I/II error the user want to control under NP-paradigm. Also, the oracle inequality provided for each type of classifiers will vary.

SIMULATION AND DISCUSSIONS

Neyman-Pearson Classifier applied on a Real Dataset.

Example. (Arcene Dataset from UCI repository) Samples include patients with cancer (ovarian or prostate cancer), and healthy or control patients. There are 100 samples in the training set; 100 samples in the test set; with 10000 features and two classes of patients with and without cancer. The original features indicate the

[3]"...even independence rules can be as poor as random guessing due to noise accumulation."[2] This is one of major reaons why naive Bayes performs poorly in high dimensional settings.

[4]The most important one is to assume $P_0\{r(\hat{x}) \leq t\}$ is continuous in t almost surely in order to control the explicit convergence rate through enveloping this quantity by $Beta_{k,m3+1-k}(1-t)$. However, this can be done by a proper choice of kernels in $\hat{r}$.

[5]Another important assumption is Assumption 1 in [2] where they assumed i.i.d. samples in order to perform a sample-splitting.

FIGURE 1. The NP classifier(LHS) versus the Naive Bayes classifier using the Arecene data(100 features).

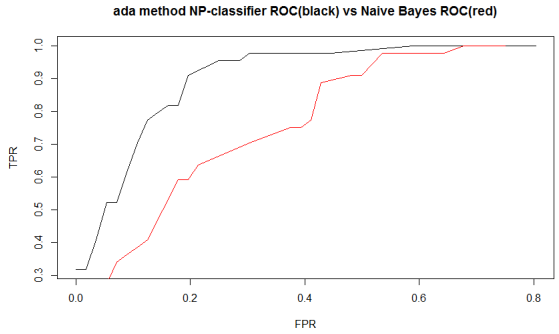


TABLE 2. Performance of NPC with various number of features on Arcene dataset.

Numbers of Features (covariates)	Overall Accuracy $\frac{correct}{all}$	Type I error on test set
50	0.58	≈ 0
100	0.63	0.05357143
500	0.63	0.08928571
1000	0.56	0.01785714
10000	0.56	≈ 0

TABLE 3. Average error rates over 1000 random splits for neuroblastoma dataset. (Table 1 in [2])

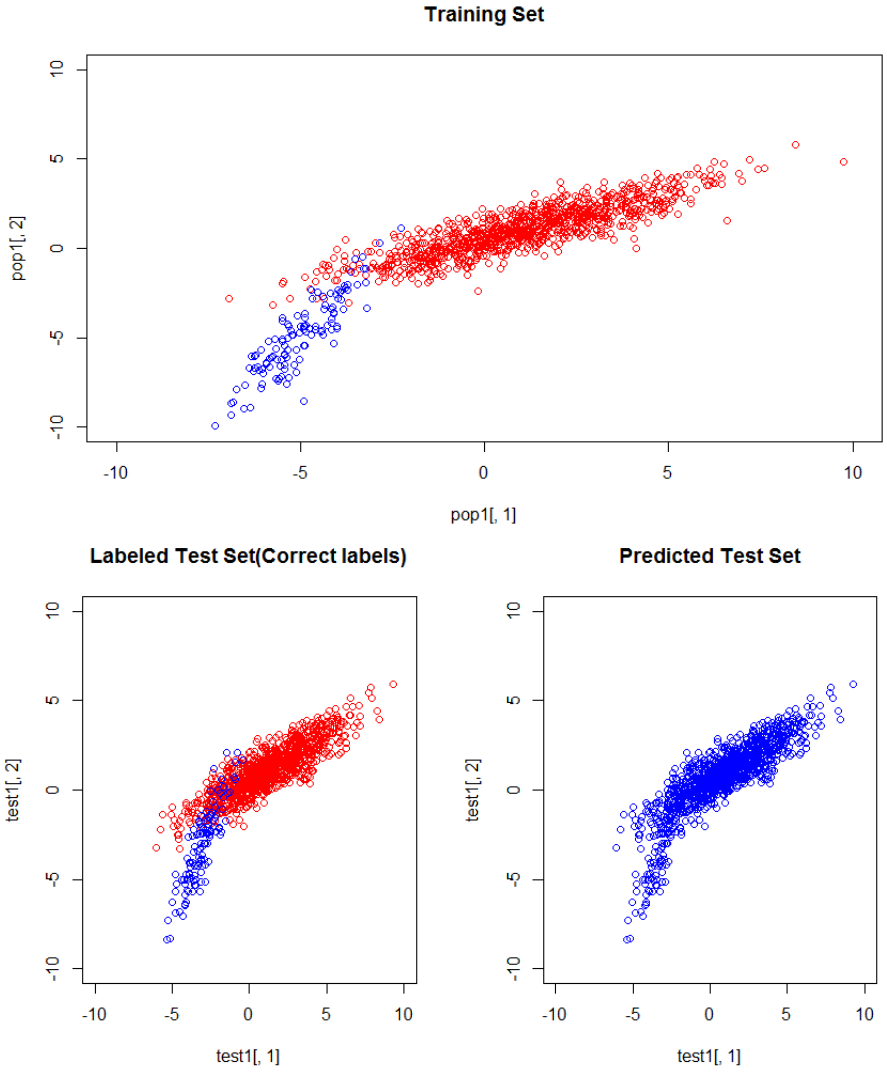
Error Type	PSN[2]	nb	pen-log	svm
type I (0 as 1)	.038	.308	.529	.603
type II (1 as 0)	.761	.150	.103	.573

abundance of proteins in human sera having a given mass value. Based on those features one must separate cancer patients from healthy patients. The donor of the dataset added a number of distractor feature called 'probes' having no predictive power. The order of the features and patterns were randomized.

This example shows that the NPC generally performs quite well in protection of type I error at $\alpha = 0.05$ level and attains relatively high accuracy. Along with another analysis of real dataset provided by [2], we can assert that NPC fits in NP paradgim quite well and can be considered as a decent choice when type I/II errors are of concern in the analysis.

Discussions. However, their method is not perfect. As pointed out in [1] and personal correspondence with one of the authors (Prof. Y.Feng), we both agreed that to preserve the type I error under NP-paradigm, the sample size cannot be too small. However, the NP-classifier cannot preserve type I error even for a 120(labeled 0)-1000(labled 1) training sample, which is quite surprising (Figure 2). From the correspondence, the author suggested that we reverse the labels to attain the type

FIGURE 2. An example that has small sample labeled 0. It is obvious that in the resulting prediction, NPC performs poorly.



I error protection. The performance of NPC is asymmetric due to our desire to control the type I error, and if we adopt the solution of reversing the labels, then NPC will slow down in case the sample sizes are unbalanced. [6] The asymmetric behavior comes from the fact that we want to control the Type I error but not Type II error.

Another drawback of this classification method is that when the underlying distribution is actually discrete, then the performance of the Neyman-Pearson classifier

[6]"If you look at the paper, you will see a lower bound of the class 0 size depending on the value of alpha and delta. You could reverse the class 0 and class 1 label and it should work. Another way is to increase alpha to 0.2 or increase delta....Indeed, the performance is indeed asymmetric, due to the fact that we want to control the type I error. " Email correspondence with Dr.Feng.

FIGURE 3. For discrete underlying distributions, the performance of NPC and naive Bayes classifier is almost indistinguishable.

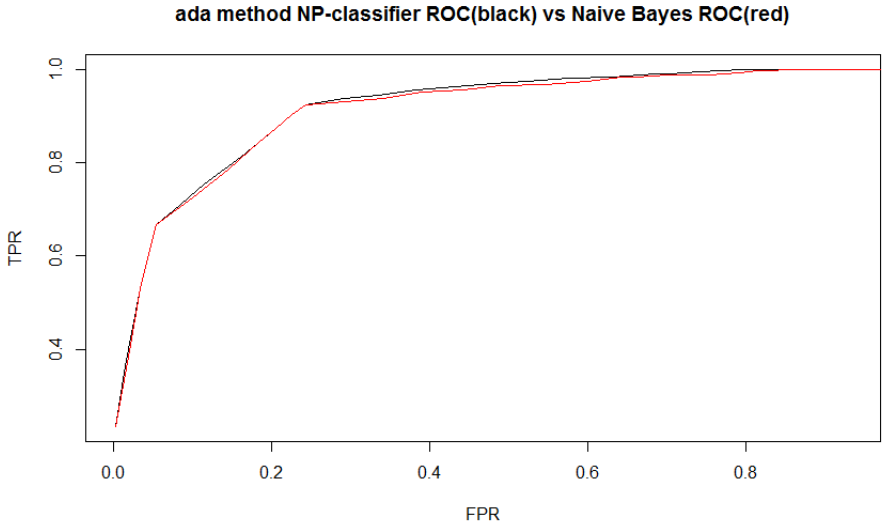


TABLE 4. Type I error and accuracy of NPC with continuous underlying distributions

Size of training sample		Type I error	Accuracy
labeled 0	labeled 1		
1000 2d normal	1000 2d normal	0.038	0.8085
100 2d normal	10000 2d normal	0	0.00990099
10000 2d normal	100 2d normal	0.0417*	0.9557426
1000 2d normal mixture	1000 2d normal	0.038	0.7935
1000 2d cauchy	1000 2d normal	0.022	0.6365

*This simulation is very slow.

seldom out-performs the naive Bayes classifier and the accuracy is not satisfying as shown in Figure 3. A primary guess of the failure of NPC in this situation is that $P_0\{\hat{r(x)} \leq t\}$ is near discontinuous almost surely. The repeated observations from the underlying discreteness may cause density ratio estimator to behave badly. Regarding this problem, the author responded me by kindly suggesting using more suitable classifier when the underlying distribution seems to be discrete.

> "Regarding the discrete distribution, when using npc, you can use other classifiers which work well for discrete distributions, for example, you can use logistic regression or even random forest. They don't involve the KDE. "[7]

[7]Email correspondence with Dr.Feng.

TABLE 5. Type I error and accuracy of NPC with discrete underlying distributions

Size of training sample		Type I error	Accuracy
labeled 0	labeled 1		
1000 2d multinomial	1000 2d multinomial	0.016	0.525
100 2d multinomial	10000 2d multinomial	0.04	0.1234653
10000 2d multinomial	100 2d multinomial	0.0382 *	0.9535644

*This simulation is very slow.

References

[1] Tong, Xin, Yang Feng, and Jingyi Jessica Li. "Neyman-Pearson (NP) classification algorithms and NP receiver operating characteristic (NP-ROC) curves." arXiv preprint arXiv:1608.03109 (2016).

[2] Zhao, Anqi, et al. "Neyman-Pearson Classification under High-Dimensional Settings." arXiv preprint arXiv:1508.03106 (2015).

[3] Fan, Jianqing, and Yingying Fan. "High dimensional classification using features annealed independence rules." Annals of statistics 36.6 (2008): 2605.

[4] Han, Min, Dirong Chen, and Zhaoxu Sun. "Analysis to Neyman-Pearson classification with convex loss function." Analysis in Theory and Applications 24.1 (2008): 18-28.

[5] Pfanzagl, Johann. "A characterization of the one parameter exponential family by existence of uniformly most powerful tests." Sankhyā: The Indian Journal of Statistics, Series A (1968): 147-156.

[6] Alexander, Anatoli Juditsky, and Arkadi Nemirovski. "Hypothesis testing by convex optimization." Electronic journal of statistics 9.2 (2015): 1645-1712.

[7] Lehmann, Erich L., and Joseph P. Romano. Testing statistical hypotheses. Springer Science & Business Media, 2006.

[8] Berger, James O. Statistical decision theory and Bayesian analysis. Springer Science & Business Media, 2013.

[9] Rockafellar, Ralph Tyrell. Convex analysis. Princeton university press, 2015.